

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»
Ройтлінгенський університет техніки та економіки (Німеччина)
Еслінгенський університет прикладних наук (Німеччина)
Технічний університет Фрайберзька гірничо академія (Німеччина)
Університет Кобленц-Ландау (Німеччина)
Краківська гірничо-металургійна академія (Польща)
Вроцлавський технічний університет (Польща)
Дніпропетровський національний університет імені Олеся Гончара
ДКХ «Дніпровський машинобудівний завод»
ДАТ «КБ Дніпровське»

ПРОБЛЕМИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ОСВІТІ, НАУЦІ ТА ПРОМИСЛОВОСТІ

XVIII МІЖНАРОДНА КОНФЕРЕНЦІЯ

24 листопада 2023 року
м. Дніпро

Збірник наукових праць
№ 8

Дніпро
НТУ «ДП»
2023

УДК 622 (06)
П78

Редакційна колегія:

А.А. Азюковський, Г.Г. Півняк, О.Б. Іванов, І.М. Удовик, М.О. Алексєєв, В.І. Корнієнко, В.В. Гнатушенко, В.В. Ткачов, В.В. Слесарєв, Л.І. Мещеряков, Б.І. Мороз, А.В. Бубліков, Т.А. Желдак, Г. Грюллер, Н. Нойбергер, А. Дерен, Я. Сконечний, Яцек Стефанські, Павел Чарнул, О.І. Сироткіна.

П78 **Проблеми** використання інформаційних технологій в освіті, науці та промисловості: XVIII міжнар. конф. (24 листопада 2023 р., м. Дніпро): зб. наук. пр. / ред. кол.: А.А. Азюковський та ін.; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2023. – № 8. – 219 с.

ISBN 978-966-350-760-6

Подано результати теоретичних та експериментальних досліджень з різних аспектів використання інформаційних технологій в освіті, науці та керування промисловістю. У публікаціях розглянуто питання створення та вдосконалення програмних засобів обробки та передачі інформації, математичного моделювання, дистанційної освіти, інформаційної безпеки та телекомунікації.

Для наукових, інженерно-технічних співробітників і студентів, які спеціалізуються в галузі обчислювальної техніки та інформаційних технологій.

УДК 622 (06)

ISBN 978-966-350-760-6

© НТУ «Дніпровська політехніка», 2023

РОЗДІЛ 1

МІЖНАРОДНЕ СПІВРОБІТНИЦТВО У СФЕРІ ОСВІТИ, НАУКИ І ВИРОБНИЦТВА

УДК 004.4

Б.І. Мороз¹, О.І. Сироткина², С.І. Кострицька¹, М.В. Ларикова¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²School of Computer Science University of Windsor, Windsor, Ontario, Kanada

SELENIUM VS. PLAYWRIGHT: EXPLORING END-TO-END TESTING IN MULTI-THREADED ENVIRONMENT

Анотація. Описано ефективність процесу наскрізного тестування веб-застосунків сучасними засобами автоматизації, такими як Playwright та Selenium, у багатопоточному середовищі. Дослідження проводиться за допомогою запуску тестів за різних обставин.

Ключові слова: *automation testing, multithread environment, parallel testing, web-application, Playwright, Selenium, end-to-end testing, JavaScript, TypeScript.*

Introduction. In recent years, automation testing has gained immense popularity within the software development process. Major corporations were among the early adopters of automated testing, recognizing its potential as a vital tool for ensuring the quality of software applications. Its primary function is to mitigate the risk of defects infiltrating production applications and impacting end-users.

The pioneer among open-source script-based automation tools is Selenium [1]. It was created to automate testing for various applications, allowing testers to write scripts in languages like Java, C#, Python, Perl, JavaScript, and Ruby. The chart provided below (Fig. 1) highlights the growing interest in Selenium over the past decade. Notably, when Selenium made its debut, its potential role in the testing process was not immediately evident to software companies. However, with the passage of time, Selenium's pivotal role in the quality control process became increasingly clear.

Over the past year, a new player has made significant inroads in the world of open-source, script-based automation libraries – Playwright (Fig. 2) [3]. It shares similarities with Selenium but also brings its own unique characteristics to the table. Notably, Playwright does not match Selenium's extensive feature set. For example, while Selenium can perform tests on actual devices, Playwright is limited to emulating them. Furthermore, Playwright may not offer support for certain legacy browsers, which can affect its ability to assess application stability.

The testing process comprises various stages and test types. Most of these tests are executed at a lower level and are typically the responsibility of developers. These are commonly known as unit and integration tests. However, the simulation of user interactions requires a higher-level approach, often seen in end-to-end and exploratory

testing. End-to-end testing involves automated scripts that replicate a user's entire journey through an application, either for a single session or a specific end goal. In contrast, exploratory testing is a manual form of end-to-end testing.

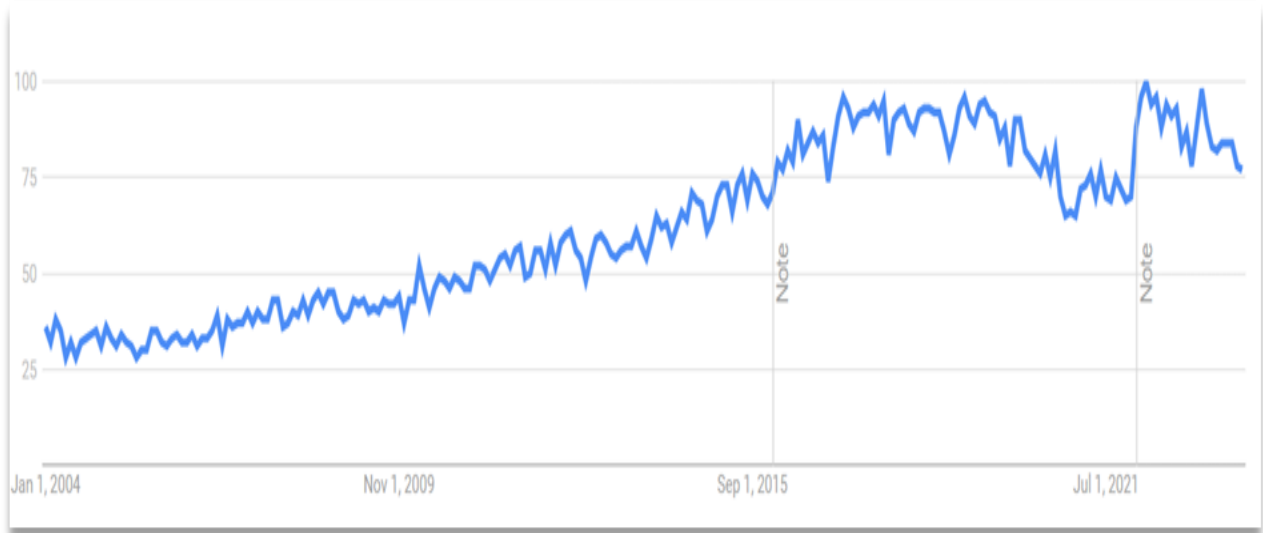


Figure 1. Google Trend statistics of 'Selenium' search request [2]

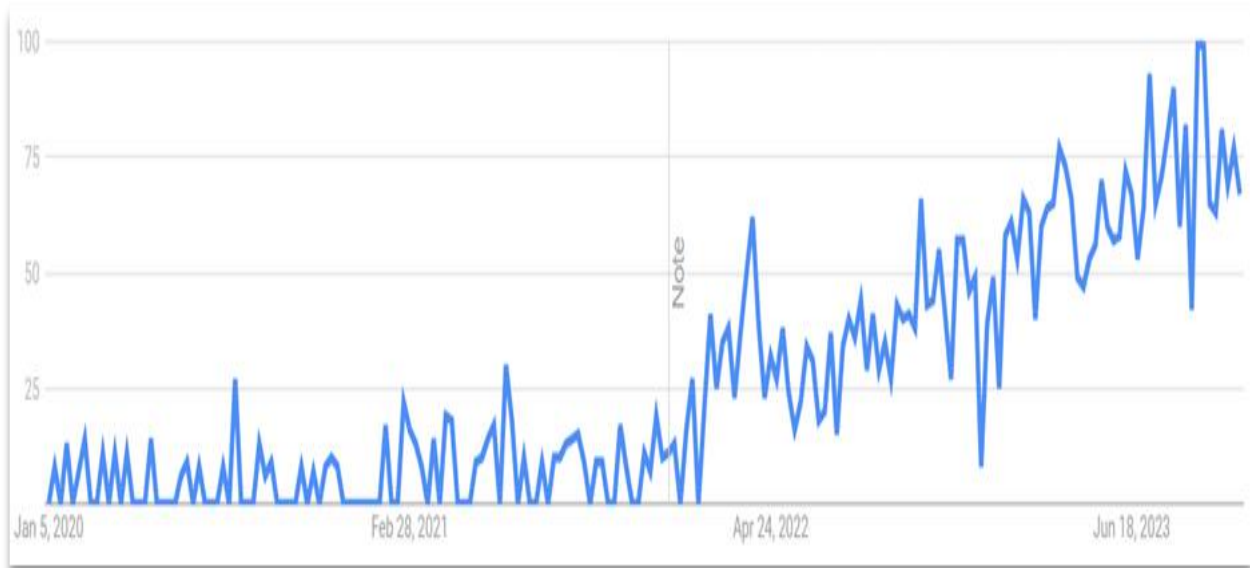


Figure 2. Google Trend statistics of 'Playwright automation' search request [4]

Formulation of the problem. Which solution proves to be more efficient and faster in parallel end-to-end testing, Playwright, or Selenium? To address this question effectively, we must embark on a comprehensive exploration of the challenges inherent in multithread testing. This exploration entails a series of pivotal steps:

1. Thoroughly examining the multithread testing strategy.
2. Weighing the pros and cons of parallel testing.
3. Uncovering techniques for executing tests in parallel with Selenium.

4. Discovering methods for parallel test execution with Playwright.
5. Developing identical test scripts for both tools.
6. Carrying out parallel tests and assessing the results.

The main content of the work. In order to standardize both tools and achieve script uniformity, JavaScript will serve as the common language for this project. Scripts will be written in VSCode to enhance the development experience, utilizing extensions that facilitate test execution. As our testing subject, we have chosen 'Swag Labs,' an open-source testing application known for its various defects, including a critical performance glitch, which adds a layer of complexity to our testing. This choice offers a broad spectrum of challenges, including:

- Varied browser environments.
- Differing thread counts.
- The presence of a performance glitch.
- Diverse operating systems.

To ensure an optimal experience when running automation tests in parallel [5], adhering to best practices is essential. Firstly, each test should be individual and self-contained, allowing for simultaneous execution without dependencies on others. Equally important is the avoidance of static objects. If there is uncertainty about whether such objects need to be shared between tests or may be unpredictably modified by them, it is advisable to steer clear of static objects. Lastly, but by no means least, it is crucial to reset all generated data during the test process.

When analyzing the multithread strategy, there are pros and cons that influence the grouping and separation of tests, as outlined in Table 1. If a project has the necessary resources, time, and a willingness to embrace challenges, there is no reason to avoid parallelizing the testing process.

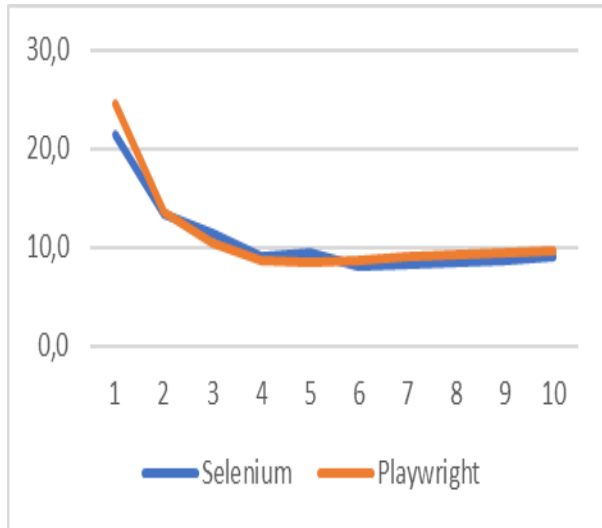
Table 1.

Pros and cons of multithread testing

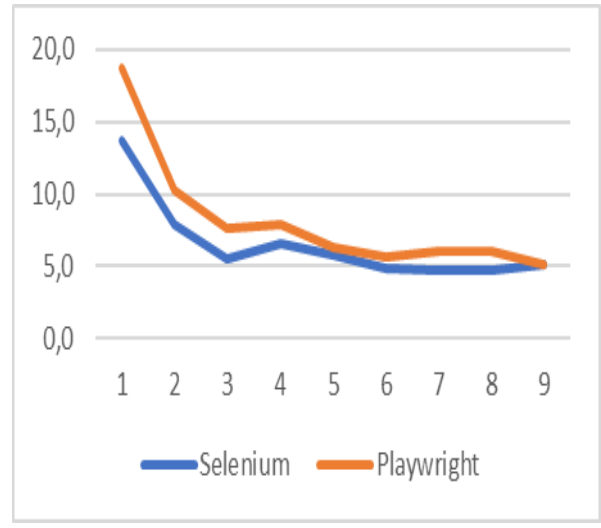
Advantages	Disadvantages
Less time on execution	Hard to configurate and rises complexity of architecture
Better coverage	Not all features allow parallel running
Improved testing process because it gives ability and time test more	Reduced ability to debug and troubleshoot

After collecting all the data and running tests under various conditions, several conclusions can be drawn. Firstly, the time dynamic of each run exhibits a hyperbolic trend – an increase in threads leads to faster system response until a saturation point,

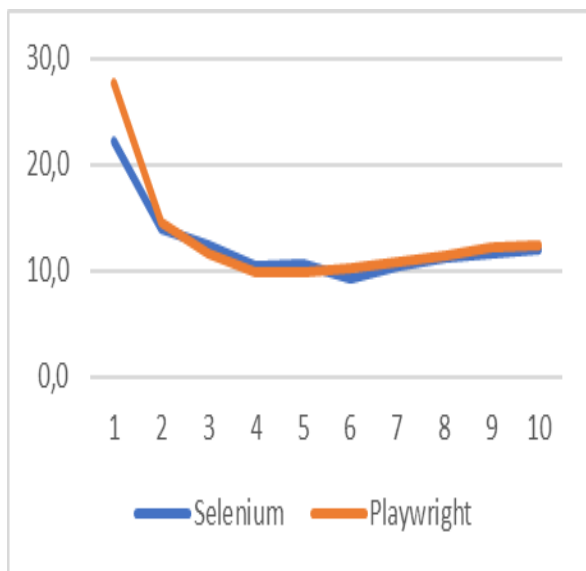
after which the performance levels off. Notably, the results reveal that Playwright exhibits slightly slower performance than Selenium on Windows OS when using the Chrome driver (Graphs 1-4). While this difference may seem minor and insignificant with only 10 tests, it becomes more apparent with larger test suites. For projects with over 1000 tests, the time difference would scale up significantly.



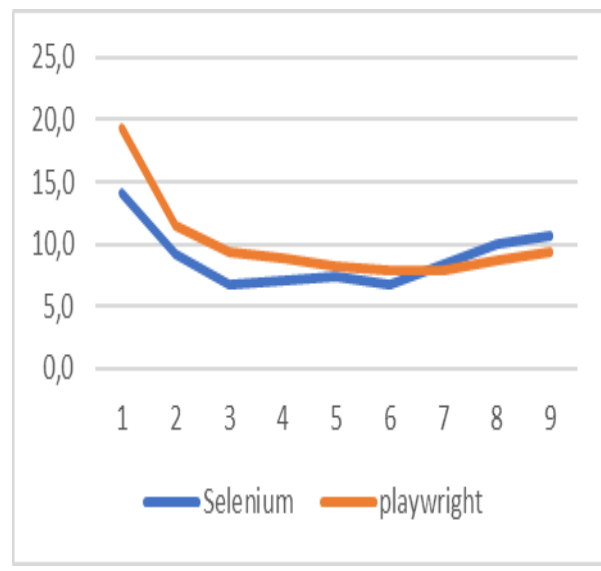
Graph 1. Results of headless test run using Chrome with glitch defect. Playwright minimum: 8.5s; Selenium minimum: 8.19s.



Graph 2. Results of headless test run using Chrome without glitch defect. Playwright minimum: 5.14s; Selenium minimum: 4.75s.

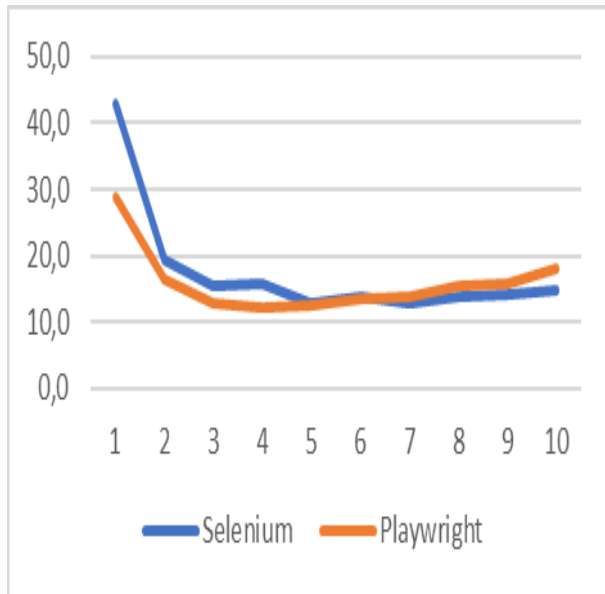


Graph 3. Results of headed test run using Chrome with glitch defect. Playwright minimum: 9.16s; Selenium minimum: 9.31s.

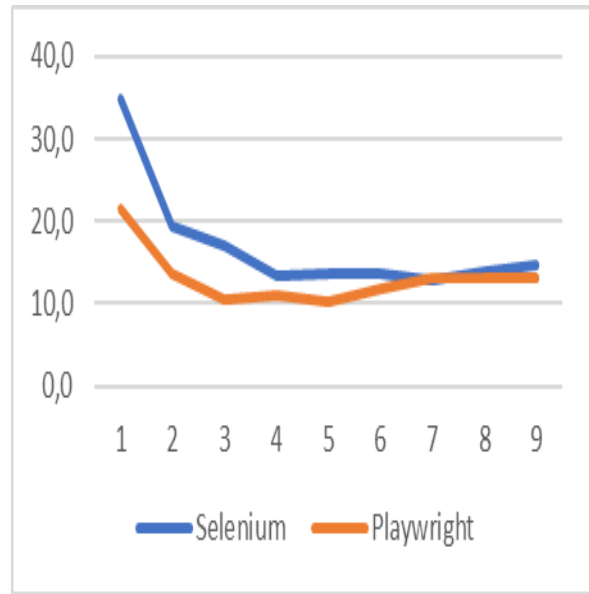


Graph 4. Results of headed test run using Chrome without glitch defect. Playwright minimum: 7.82s; Selenium minimum: 6.72s.

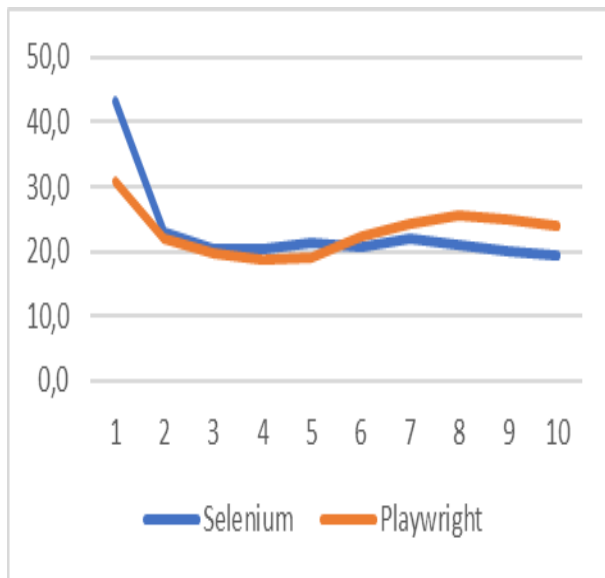
However, Selenium shows less favorable results when running on Windows OS using the Firefox driver (Graphs 5-8). Although the impact is not substantial with this number of tests, it becomes more noticeable with a higher volume.



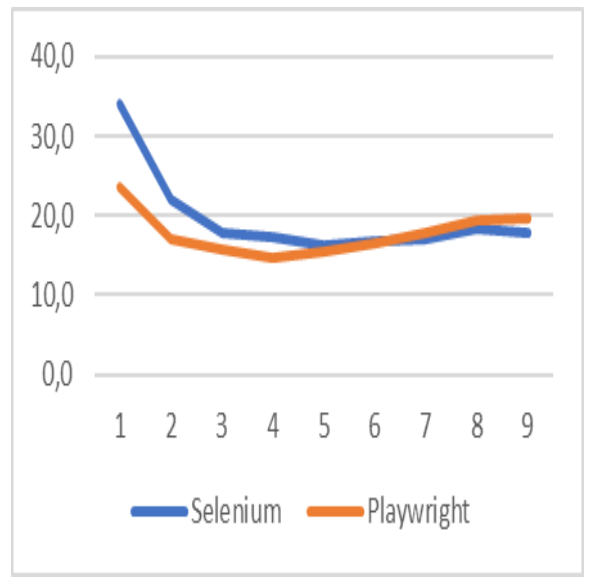
Graph 5. Results of headless test run using Firefox with glitch defect. Playwright minimum: 12.24s; Selenium minimum: 12.8s.



Graph 6. Results of headless test run using Firefox without glitch defect. Playwright minimum: 10.24s; Selenium minimum: 12.9s.



Graph 7. Results of headed test run using Firefox with glitch defect. Playwright minimum: 18.66s; Selenium minimum: 19.23s.



Graph 8. Results of headed test run using Firefox without glitch defect. Playwright minimum: 14.7s; Selenium minimum: 16.37s.

In the case of Mac OS, there is already a clear time difference between Playwright and Selenium. Playwright completes all tests in just 3.6 seconds, while Selenium takes at least 15.60 seconds. Another challenge faced by Selenium on Mac OS pertains to the absence of a newer stable version of the Chrome driver. This deficiency results in performance lags, making the Chrome driver respond sluggishly to testing tool actions. This issue is particularly prominent during manual interactions in Chrome. Consequently, for a smoother experience, Selenium on Mac OS consistently demands a more recent stable version.

The scientific novelty of this study lies in its investigation of parallel execution of end-to-end tests using Playwright and Selenium across a range of conditions. These conditions include factors such as web application response speed deviations, utilization across different browsers, session reuse, varying browsers and operating systems, and different thread counts.

Conclusions. Selenium, as an established tool with a large and well-established community, offers a plethora of features and capabilities in automation testing. In contrast, Playwright is a relatively recent addition to the scene and comes with its own set of limitations.

However, what sets Playwright apart is its superior performance in parallelization and faster execution of actions during test runs. This has contributed to its increasing popularity and the expansion of its community. As a result, parallelization, although a complex process, has become more accessible and efficient for testers. Nevertheless, it is important to note that the parallelization process is intricate, with numerous factors affecting the stability, speed, and potential conflicts of tests.

ПЕРЕЛІК ПОСИЛАНЬ

1. Selenium (software) – Wikipedia [Електронний ресурс]. – 29.09.2023. – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Selenium_\(software\)](https://en.wikipedia.org/wiki/Selenium_(software)).
2. Selenium - Explore - Google Trends [Електронний ресурс]. – 30.10.2023. – Режим доступу до ресурсу: <https://trends.google.com/trends/explore?date=all&q=%2Fm%2F0c828v&hl=en>.
3. Playwright (software) – Wikipedia [Електронний ресурс]. – 27.10.2023. – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Playwright_\(software\)](https://en.wikipedia.org/wiki/Playwright_(software)).
4. Playwright automation - Explore - Google Trends [Електронний ресурс]. – 30.10.2023. – Режим доступу до ресурсу: <https://trends.google.com/trends/explore?date=2020-01-01%202023-10-30&q=Playwright%20automation&hl=en>.
5. Parallel Testing: A Complete Guide [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://testsigma.com/parallel-test-runs>.
6. Playwright vs. Selenium in 2023: Which Is Better? - ZenRows [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.zenrows.com/blog/playwright-vs-selenium#which-is-better>.

CREATING USING THE MATHCAD SYSTEM OF LABORATORY EXPERIMENTATION ON THE SUBJECT "NUMERICAL METHODS IN COMPUTER SCIENCE"

Annotation. Methodical recommendation for laboratory experimentation on the subject of "Numerical methods in computer science", based on the MathCAD system, will be briefly described. The experimentation for each laboratory research corresponds to open-source code, transparently related to mathematical models on the topic of the work. The developed experimentation consists of 6 research variations. Input data for each research has 26 options.

Keywords: *numerical methods, numerical differentiation, numerical integration, solving algebraic equations, solving differential equations.*

Introduction. Currently, in the training of IT specialists, one of the main educational components of the professional constituent is the subject "Numerical methods in computer science". In the study programs of some specialties it is called as "Computational Methods" or "Computer Mathematics". Known approaches to the organization of laboratory experimentation, in this subject, such as [1 – 3] used as instrument MATLAB, Python or VBA for Excel.

At the same time, while introducing students to the tools of numerical methods, it is desirable for software implementations of these tools to have open-source transparently related to their mathematical models. Solving such problems, even with the use of specialized languages that are adapted to mathematical calculations, such as MATLAB, Python or VBA for Excel, not to mention the popular C++, C# and Java, is laborious and doesn't provide transparency of the generated code. The best software in this case, in our opinion, is the system of computer algebra MathCAD [4], or its further development MathCAD Prime [5]. The main difference, between these systems and similar ones, is the availability of graphic templates for entering program sentences. The templates reproduce the symbols used in mathematics, making the program code compact and transparent. MathCAD uses its own scripting language. To organize the simplest cycles, ranked variables are used, more complex algorithms are implemented within program blocks.

Purpose of the research is to create, using MathCAD system methodical recommendation and software for laboratory experimentation on the subject of "Numerical methods in computer science", in which each laboratory research corresponds to open-source code, transparently related to mathematical models on the topic of the work.

The main content of the work. The developed laboratory experimentation consists of 7 research topics. Input data for each research has 26 options. A brief overview, of these research variations, is given below.

Research № 1. *Name: Numerical differentiation.*

Input data:

- polynomial and transcendental functions;
- two formulas for numerical differentiation of 1st and 2nd or 3rd order.

Brief content of research:

- numerically calculate the values of derivatives for all given formulas and functions at a point $x=1$ with a differentiation steps $h = \{0.01, 0.05\}$;
- numerically calculated values of derivatives for all given functions at a point $x=1$ correspond to the value obtained using a graphical differentiation template;
- analytically calculate derivatives for all given functions using a graphical differentiation template and find their values at a point $x=1$.

Typical conclusions:

- the analytically calculated derivatives for all given functions at a point $x=1$ take on values that coincide with the corresponding values that are calculated using a graphical differentiation template;
- the values of the derivatives at a point $x=1$ calculated using numerical differentiation formulas are close to the corresponding values calculated by other methods;
- with increasing differentiation step, the errors in the approximate values of the derivatives, which are calculated using numerical differentiation formulas, increase.

Research № 2. *Name: Numerical integration.*

Input data:

- definite integral;
- four methods of numerical integration (from a variety: quadrature formulas of left, middle, right rectangles, trapezoids, parabolas and the Monte Carlo method), which are determined by the option number.

Brief content of research:

- numerically calculate definite integral using the chosen methods. For Monte Carlo method use $N = \{10^3, 10^5\}$ random numbers, for other methods – $n = \{10, 50\}$ integration steps. Perform calculations using the summation template and using a solver block;
- calculate the finite differences 1st and 2nd order of the integrand function on the integration segment for a given number of steps $n=10$;
- calculate theoretical value upper bound estimation error of integral for rectangles formula;
- calculate the value of a definite integral using the appropriate graphical template;

- analytically calculate the antiderivative of the integrand function. Calculate the value of the definite integral using the Newton-Leibniz formula.

Typical conclusions:

- the analytically calculated definite integral take on value that coincide with the value that are calculated using a graphical integration template;
- the values of the definite integral calculated using all of numerical methods are close to the corresponding values calculated by analytically and using a graphical integration template;
- with increase in the number of integration steps, the errors of the approximate values of the integral, which are calculated using all of numerical integration formulas, decrease.

Research № 3. *Name:* Solving algebraic equations with real roots.

Input data:

- two equations of the form $f_1(x)=0$ and $f_2(x)=x$ where $f_1(x)$ is the polynomial function and segments of their roots localization;
- two numerical methods of solving algebraic equations with real roots (from a variety: dichotomies, chords and tangents), which are determined by the option number.

Brief content of research:

- plot the dependence of $y = f_1(x)$ on the localization segment of the root of the equation $f_1(x)=0$ and graphically define the root;
- numerically define the root of the equation $f_1(x)=0$ by choosed methods;
- solve the equation $f_1(x)=0$ by root and polyroots functions;
- plot the dependence of the iteration numbers on upper boundary value of the module of the difference in approximations of the solution vector at successive iteration steps $\Delta = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for choosed methods;
- plot the dependence of $y = f_2(x) - x$ on the localization segment of the root of the equation $f_2(x)=x$ and graphically define the root;
- numerically define the root of the equation $f_2(x)=x$ by method of iteration;
- solve the equation $f_2(x)=x$ by root function;
- plot the dependence of the iteration numbers on upper boundary value of the module of the difference in approximations of the solution vector at successive iteration steps $\Delta = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for method of iteration.

Typical conclusions:

- the numerically calculated root of the equation $f_1(x)=0$ by choosed methods take on values that coincide with the corresponding values that are obtained graphically and used root and polyroots functions;

- the numerically calculated root of the equation $f_2(x) = x$ by iteration method take on values that coincide with the corresponding value that are obtained graphically and used root function;

- the dependences of the iteration numbers on upper boundary value of the module of the difference in approximations of the solution at successive iteration steps has monotonically decreasing character for all numerical methods.

Research № 4. *Name:* Solving algebraic equations with complex roots

Input data is equation of the form $f(x) = 0$ where $f(x)$ is the polinomyal function and the semiring of localization of its complex root.

Brief content of research:

- numerically define the root of the equation by method of Monte Carlo for numbers of iteration $N = \{10^3, 10^4, 10^5, 10^6\}$;

- solve the equation by polyroots function;

- plot in general graphycal field the dependences on the number of itteration of the module of difference between approximation value of the root (which obtained by method of Monte Carlo) and conditionally exact value of the root (which obtained by polyroots function) and theoretical upper bound of the root localization error

$$\Delta(N) = \sqrt{\frac{\pi \cdot (R_1^2 - R_2^2) \cdot \ln(1-p)}{8 \cdot N}}$$

where R_1, R_2 is the outer and inner radii of the root localization semiring, $p = 0.9$ is the he probability of the approximate value of the root falling within the Δ -neighborhood of the true one.

Typical conclusions:

- numerically calculated value of root correspond to the value obtained using by polyroots function;

- module of difference between approximation value of the root and the conditionally accurate root value is always less than the corresponding value of the upper bound of the root localization error;

- the dependences on the number of itteration of the module of difference between approximation value of the root and conditionally exact value of the root and theoretical upper bound of the root localization error has monotonically decreasing character.

Research № 5. *Name:* Solving systems of linear algebraic equations.

Input data is a compatible definite system of 3 linear algebraic equations with 3 unknowns and a dominant main diagonal.

Brief content of research:

- solve the system by simple itterations method;

- plotting the dependence of the iteration numbers on upper boundary value of the norm of the difference in approximations of the solution vector at successive iteration steps $\Delta = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ for simple itterations method;

- solve the system by matrix method;

- solve the system by lsolve function;

- solve the system by decisive block;
- solve the system by Cramer method.

Typical conclusions:

- the solutions to the system of linear algebraic equations, obtained by all the methods used, coincide;
- the dependence of the iteration numbers on upper boundary value of the norm of the difference in approximations of the solution vector at successive iteration steps for simple iterations method has monotonically decreasing character.

Research № 6. *Name:* Solving of ordinary differential equations.

Input data:

- 1st order ordinary differential equation in Cauchy form with initial condition;
- numerical method of ordinary differential equations integration (from a variety: corrected Euler method, modified Euler method, Runge-Kutt method,), which are determined by the option number.

Brief content of research:

- numerically integrate the differential equation using the chosen method for $n = \{30, 100\}$ integration steps;
- integrate the differential equation by Odesolve function;
- integrate the differential equation by rkfixed function;
- plot in general graphical field all the obtained partial solutions of the differential equation.

Typical conclusions:

- partial solutions of the differential equation, which were obtained by all the methods used, correspond to each other;
- with an increase in the number of integration steps, the deviation of the solution obtained by the numerical method from solutions using functions Odesolve and rkfixed decreases.

Research № 7. *Name:* Solving partial differential equations.

Input data is a shape and parameters of one-dimensional spatial distribution.

Brief content of research:

- for a given one-dimensional spatial distribution – function of the initial condition (initial distribution of the rod temperature), numerically solve the task of rod cooling down, using difference approximation of the parabolic type partial differential equation. To animate the graph of the solution function $U_{i,j}$. Let select: duration of the process $T = 1$ s, rod length $L = 1$ m, equation proportionality coefficient $d = 1$ m/s, the number of integration steps at time $n = 2 \cdot 10^4$, at spatial coordinate $m = 100$;

- using analytical solutions to the task of rod cooling down, for previous conditions to plot a graph of function-solution $u(t,x)$ for the moments of the time $t = \left\{ \frac{L}{100 \cdot d}, \frac{L}{50 \cdot d}, \frac{L}{20 \cdot d}, \frac{L}{10 \cdot d}, \frac{L}{5 \cdot d}, \frac{L}{2 \cdot d}, \frac{L}{d} \right\}$. Select the number of members of the partial sum in the row that indicates the decision $k = 100$.

- for a given one-dimensional spatial distribution – function of boundary conditions (distribution of electric potential of vertical boundaries) numerically solve the Dirichlet task for the Laplace equation on a two-dimensional square domain, using difference approximation of the elliptic type partial differential equation. Plot the graph of the isolines of the solution function $U_{i,j}$. Let select: the length of the side of the region $p = q = 1$ m, the value of the potential on the horizontal boundaries $U_{0,j} = U_{m,j} = 0$, the number of internal nodes of the grid $m - 1 = 3$.

- using analytical solutions to the Dirichlet task for the Laplace equation on a two-dimensional square domain for previous conditions to plot the graph of the isolines of the solution function $u(y, x)$. Select the number of members of the partial sum in the row that indicates the decision $k = 10$ and the number of grid nodes in which function-solution is determined $m = 100$.

Typical conclusions:

- numerical solutions for equations of both types correspond to analytical ones;
- For an elliptic type equation, the isolines of the numerical solution are piecewise linear, while those of the analytical solution are smooth. This is explained by the small number of equations of the difference scheme, which was used in the numerical solution.

Conclusions. This paper has briefly described methodical recommendations for laboratory experimentation on the subject of "Numerical methods in computer science" based on the MathCAD system. The experimentation for each laboratory research corresponds to open-source code, transparently related to mathematical models on the topic of the work. The developed experimentation consists of 7 research variations. Input data for each research has 26 options. Methodical recommendations and appropriate software are posted on the distant education platform of Dnipro University of Technology, Department of Computer System's Software [6].

References

1. Дейнега Л.Ю. Методичні вказівки до лабораторних робіт з дисципліни «Чисельні методи» для студентів напряму підготовки 122 «Комп'ютерні науки». [Текст] / Г.М. Шило, Н. О. Миронова, Л.Ю. Дейнега. – Запоріжжя: ЗНТУ, 2017. – 36 с.
2. Мірошкіна І.В. Методичні рекомендації до лабораторних робіт з дисципліни «Алгоритми та методи обчислень» для здобувачів освітнього ступеня бакалавра спеціальності 123 «Комп'ютерна інженерія». [Текст] / І.В. Мірошкіна – Черкаси: ЧДТУ, 2018. – 106 с.
3. Потьомкіна Н.І. Методичні вказівки до програмування та розв'язання задач чисельними методами на персональних ЕОМ. [Текст] / Н.І. Потьомкіна, М.М. Товстоног – Дніпропетровськ: ДГАУ, 1994. – 28 с.
4. User's Guide Mathcad 14.0. Parametric Technology Corporation. [Електрон. ресурс]. Режим доступу: <https://www.slideshare.net/vstran1/math-cad-14-user-manual> (дата звернення: 11.12.2023).
5. Brent Maxfield. Essential PTC Mathcad Prime 3.0. A Guide for New and Current Users. [Текст] / Elsevier Inc., 2014, 563 с.
6. Кожевников А.В. Лабораторний практикум та індивідуальні завдання з дисципліни «Чисельні методи в інформатиці». [Електрон. ресурс]. Режим доступу: <https://do.nmu.org.ua/course/view.php?id=1427> (дата звернення: 11.12.2023).

С.Д. Приходченко¹, С.Г. Семенов², К.С. Родна¹, О.К. Лисицький¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²University Education Commission, Krakow, Poland

ВИВЧЕННЯ МЕТОДІВ ГЕЙМІФІКАЦІЇ ТА РОЗРОБКА ІННОВАЦІЙНИХ ГЕЙМІФІКАЦІЙНИХ ПІДХОДІВ

Анотація. Дослідження присвячені вивченню гейміфікації як нового підходу до навчання та мотивації студентів. Основною метою дослідження є впровадження гейміфікованого додатку у вигляді телеграм-боту в процес навчання студентів. Зокрема, робота спрямована на випробування інноваційних методів гейміфікації в конкретній освітній сфері. У подальших дослідженнях планується розробка і вдосконалення гейміфікованих підходів для оптимального впровадження в освітній процес з метою підвищення якості навчання та залучення студентів до активної участі.

Ключові слова: освітній процес, гейміфікація, телеграм-бот, модернізація процесу, інновації в освіті, студентська мотивація, педагогічні технології, активне навчання, інтерактивний підхід, взаємодія викладача та студента.

Вступ. В сучасному світі навчальні заклади активно працюють над впровадженням цифрових технологій у навчальний процес, щоб покращити якість освіти та збільшити інтерес студентів, які належать до сучасного покоління. Однією з інноваційних і ефективних стратегій є використання гейміфікації у навчанні.

Історія гейміфікації сягає свого коріння ХХ століття, коли ігрові механіки були вперше використані в маркетингу та продажах. Термін "гейміфікація" з'явився в 1980-х роках і відтоді здобув широке застосування у різних сферах діяльності, включаючи освіту.

Сучасний підхід до навчання відзначається широким впровадженням гейміфікованих методик, особливо в онлайн-навчанні. Ігри та ігрові елементи стають не лише частиною освітнього процесу, але і його ключовим стимулом. Застосування гейміфікації у навчанні дозволяє зберегти інтерес студентів, адже вони вже звикли до ігор та інтерактивного підходу.

Розуміючи, що ігровий підхід приваблює як дітей, так і дорослих, варто підкреслити потужний потенціал гейміфікації для модернізації навчального процесу. Використання телеграм-бота для проведення тестів і оцінювання навичок студентів стає не лише зручним, але й позбавленим стресу. Такий інтерактивний підхід до навчання представляє собою інноваційний крок у сфері освіти, спрямований на підвищення якості навчання. Завдяки широкому поширенню телеграм-ботів, їх інтерактивності та доступності для всіх, гейміфікований навчальний додаток в цьому форматі стає не тільки сучасним, але й вкрай ефективним засобом залучення студентів.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані такі завдання: визначити основні аспекти гейміфікації як нового підходу до навчання та мотивації студентів;

- розглянути можливості впровадження гейміфікованого додатку у вигляді телеграм-боту в процес навчання студентів;
- провести аналіз існуючих методів гейміфікації в освіті для з'ясування їхніх переваг та недоліків;
- дослідити популярність використання месенджерів, зокрема телеграму, серед підлітків у контексті освітнього процесу;
- випробувати розроблені гейміфіковані підходи у практичному середовищі та зібрати дані про їхню ефективність;
- зрозуміти та оцінити вплив розроблених гейміфікованих підходів на мотивацію студентів та якість їхнього навчання.

Основний зміст роботи. Під час роботи були використані наступні інструменти:

- мова програмування Python;
- відкрита бібліотека для роботи з ботами pyTelegramBotAPI;
- хмарне середовище та база даних MongoDB;
- екосистема застосунку для вікторин Kahoot;
- екосистема компанії WorldIT для проведення дослідів та отримання результатів;
- сервіси з можливістю відкритого збору статистики.

Гейміфікація, як стратегічний метод навчання та мотивації студентів, наростає у популярності завдяки своїм унікальним можливостям стимулювання навчання та активізації уваги. Використання сучасних технологій дозволяє впроваджувати гейміфіковані елементи у навчальний процес, роблячи його захопливим та цікавим для студентів.

Цей підхід має величезні переваги. По-перше, гейміфікація забезпечує залучення студентів та створює мотивацію для активного навчання. Ігрові елементи та виклики, вбудовані в освітній процес, стимулюють студентів до досягнення нових цілей та розвитку.

По-друге, гейміфікація сприяє ефективнішому засвоєнню матеріалу. Вона дозволяє студентам взаємодіяти з навчальним контентом у цікавий спосіб, що сприяє глибшому розумінню та запам'ятовуванню інформації.

Гейміфікація в освіті не лише робить навчання цікавішим, але й покращує їхнє розуміння та засвоєння навчального матеріалу.

Деякі статистичні дані свідчать про значущий позитивний вплив гейміфікації на навчання:

- Згідно з дослідженням American Journal of Play, 80% студентів вважають, що гейміфікація збільшила їхню мотивацію до виконання навчальних завдань.

- За даними опитування Edutopia, 63% вчителів повідомляють про покращення результатів навчання учнів, які брали участь в гейміфікованих програмах.

Створення навчання цікавим та захоплюючим завжди викликало труднощі для вчителів по всьому світу. Один із найефективніших методів досягнення цієї мети полягає у використанні ігор для учнів. Сучасні технології дозволяють ефективно інтегрувати гейміфіковані елементи у навчальний процес, але виклик полягає у тому, щоб учні були повністю залучені до них. В такому контексті, платформа Kahoot є яскравим прикладом успішного поєднання гри та освіти.

Kahoot - це ігрова онлайн-платформа для вчителів та учнів, створена для того, щоб зробити навчання легким та цікавим. Забезпечуючи легкий доступ через будь-який пристрій з доступом до інтернету, вона дозволяє вчителям створювати вікторини для гравців, що беруть участь в них. Формат гри часто включає серію запитань, на які учні відповідають, обираючи з запропонованих варіантів.

Крім того, Kahoot здатний забезпечити вчителям дані з гри для оцінки успішності учнів. Ця платформа створена таким чином, щоб робити навчання легким і цікавим, і водночас досить складною для створення захопливого освітнього досвіду. Крім освітнього сектору, подібна гейміфікація також отримала визнання в інших галузях, зокрема у бізнесі та рекрутингу, що підкреслює її універсальність та ефективність.

Новаторський підхід до навчання представлений у формі розробленого телеграм-боту, який надає можливість проведення вікторин прямо у месенджері. Інтегруючи функціонал, схожий на популярну платформу Kahoot, застосунок дозволяє вчителям, студентам і батькам отримувати доступ до навчального процесу безпосередньо через застосунок, який усі використовують щоденно.

Вчителі можуть легко розпочинати вікторини та ефективно взаємодіяти зі студентами, просто використовуючи месенджер. Це надає зручність та легкість управління навчальним процесом, а також дозволяє вчителям стежити за успіхами та результатами кожного учня. Студенти, в свою чергу, мають можливість активно брати участь у вікторинах, обираючи правильні відповіді зі списку варіантів, і відчувати емоційну насагу до самовдосконалення.

Однією з ключових переваг цього підходу є швидкість та доступність для всіх учасників навчального процесу. Вчителі можуть легко організувати та вести вікторини, студенти з комфортом беруть участь, а батьки отримують оперативні повідомлення щодо успішності своїх дітей. Це інноваційний, сучасний спосіб здійснення навчання, що підкреслює актуальність та ефективність використання сучасних технологій в освітньому процесі.

У майбутньому телеграм-бот може розширити свій функціонал, надаючи користувачам ще більше можливостей для зручного навчання. Можливим є розвиток у напрямку «школи в смартфоні», де є усі необхідні інструменти для навчання, спілкування викладачів і студентів, а також взаємодії з батьками.

Зручне меню у вигляді динамічних клавіатур, простий доступ до функцій та відсутність необхідності використовувати велику кількість сторонніх сервісів роблять навчальний процес максимально комфортним для всіх учасників.

Наукова новизна телеграм-боту з вікторинами полягає в інтеграції гейміфікованого підходу до навчання безпосередньо в застосунку-месенджері. Цей бот не лише спрощує процес проведення та участі в вікторинах для викладачів, студентів та батьків, але і забезпечує повний зв'язок між учасниками навчального процесу. Усі функції, що пов'язані з проведенням тестів, відстеженням успіхів та спілкуванням, доступні в єдиному середовищі відомого та популярного месенджера, що дозволяє створити ефективну систему навчання.

Висновки. Усе вищезгадане підкреслює важливість гейміфікації в освіті та навчальних процесах. Інтеграція цього підходу в застосунок-месенджер відкриває нові перспективи для ефективного навчання та створює інноваційне середовище для взаємодії між викладачами, студентами і батьками. Комбінування гейміфікаційних елементів у вікторинах та простота використання телеграм-боту створюють унікальний інструмент для покращення якості навчання та залучення уваги учасників навчального процесу. Такий підхід відкриває перспективи для подальшого розвитку, можливості якого можуть вмістити увесь процес навчання в цікавому та зручному інструменті.

ПЕРЕЛІК ПОСИЛАНЬ

1. Karl M. Kapp "The Gamification of Learning and Instruction: Game-Based Methods and Strategies for Training and Education" / Pfeiffer – San Francisco, 2012 – 336 с.
2. Jane McGonigal "Reality is Broken: Why Games Make Us Better and How They Can Change the World" / Penguin Books – New York, 2011 – 400 с.
3. Andrzej Marczewski "Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design" / CreateSpace Independent Publishing Platform – 2015 – 264 с.
4. Yu-kai Chou "Actionable Gamification: Beyond Points, Badges, and Leaderboards" / Octalysis Media – 2015 – 412 с.
5. Gabe Zichermann, Christopher Cunningham "Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps" / O'Reilly Media – Sebastopol, 2011 – 210 с.

А. Л. Ширін¹, М. Потемпа², І.Г. Гуліна¹, А. В. Петрушенко¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Firmy „GeoBit”, Chrzanów, Polska

МЕТОДИ УПРАВЛІННЯ СТАНОМ ФРЕЙМВОРКУ FLUTTER

Анотація. У роботі висвітлено порівняння методів управління станом у фреймворку Flutter та визначено їх переваги і недоліків. Отримані результати даної роботи можна застосувати у подальшому в розробці мобільних застосунків з використанням методів управління станом у фреймворку Flutter.

Ключові слова: методи управління станом, мобільний застосунок, фреймворк, BLoC, Flutter, GetX, MobX, Redux, setState.

Вступ. Сьогодення у сфері інформаційних технологій є різноманітним та насиченим. Розробка програмного забезпечення (ПЗ) вже далеко зайшла за межі можливого і неможливого. Найпопулярнішим видом ПЗ є мобільні застосунки, адже смартфон саме той пристрій, який завжди під рукою. У світі мобільної розробки важливою складовою є ефективне управління станом застосунку. Вдало організоване управління станом є ключовим фактором, який впливає на швидкодію та загальну продуктивність ПЗ.

Методів управління станом може бути багато навіть у межах одного фреймворку, тому так важливо розуміти, який з них швидше працює, надає навантаження на оперативну пам'ять або займає найбільше пам'яті вихідного коду. Дуже мало робіт направлено на дане дослідження. Це є вагомим недоліком, адже перевірка вище вказаних метрик є достатньо важливими параметрами, які можуть впливати на бізнес та кінцевого користувача програмним продуктом.

Одними із найпопулярніших засобів розробки для конструювання логічного та вдалого управління станом є Flutter та Dart. Дана пара досі набирає свою популярність у всьому світі. Вони вже мають велику аудиторію та кількість офіційної та сторонньої документації, але не мають достатньої інформації про швидкодію методів управління станом на мобільних застосунках.

Таким чином актуальність обраної теми обґрунтована розвитком наукової діяльності, дослідженням методів управління станом на основі фреймворку Flutter. Отримані результати даної роботи дозволяють у подальшому використовувати їх для проведення додаткових досліджень, а також у визначенні методу управління станом, необхідного для розробки ПЗ.

Постановка задачі. Для досягнення поставленої мети необхідно вирішити наступні завдання:

- викласти основні аспекти існуючих методів управління станом фреймворку Flutter;
- визначити переваги та недоліки кожного з методів управління станом;
- надати схеми роботи методів управління станом;

– визначити найвигідніший варіант методу управління станом для розробки мобільних застосунків.

Основний зміст роботи. Під час виконання даної роботи були використані наступні теоретико-методичні матеріали:

– офіційні теоретико-методичні рекомендації щодо розробки мобільних застосунків з використанням фреймворку Flutter;

– теоретико-методичні рекомендації щодо розробки мобільних застосунків з використанням методів управління станом фреймворку Flutter;

– фреймворк Flutter, мова програмування Dart, методи управління станом BLoC, Redux, MobX, GetX, setState.

Управління станом є важливою частиною архітектури будь-якої програми, орієнтованої на користувача. Вони описують обробку змін даних у користувацькому інтерфейсі, викликаних введенням даних користувачем або фоновим процесом, що звертається до внутрішньої служби. Управління станом забезпечує синхронізацію даних та інтерфейсу застосунку і запобігає невідповідності змін у даних. У Flutter доступно багато рішень для управління станом, в основному завдяки підтримці великої спільноти фреймворку [1].

Найпопулярнішими методами управління станом у фреймворку Flutter є такі методи як setState, Business Logic Components (BLoC), GetX, Redux, MobX [1], які будуть розглянуті нижче.

Методу управління станом setState використовується для перемальовування віджета та його нащадків, коли стан віджету змінюється. Даний метод приймає функцію зворотного виклику, яка оновлює стан віджету та викликає перемальовування [2].

Оновлення стану у setState відбувається всередині функції зворотного виклику, що передається setState, а потім відбувається оновлення стану віджету за допомогою оператора присвоєння (рис. 1).

```
dart Copy code  
  
setState(() {  
  // Update the widget's state  
  _counter = _counter + 1;  
});
```

Рис. 1. Фрагмент коду оновлення стану у setState

Перевагами setState є:

– простота використання: надає простий спосіб оновлення стану віджету та оновлення інтерфейсу;

– швидкість розробки: дозволяє зосередитися на функціональності віджету, а не на складних методах управління станом.

Недоліками setState є:

- неефективність у великих застосунках;
- обмежена можливість асинхронності;
- складність управління великою кількістю станів;
- потреба акуратності при оновленні стану через можливість приведення до помилок та небажаний поведінці застосунку.

Загалом, setState є корисним інструментом для простих і невеликих віджетів у Flutter, але у більших і складніших мобільних застосунках часто потрібен більш просунутий метод управління станом, як BLoC або MobX, щоб забезпечити ефективнішу та надійнішу роботу програми.

BLoC – найпопулярніше рішення управління станом у Flutter, яке допомагає розділити шар презентації і шар бізнес-логіки, полегшуючи підтримку стану і оновлення інтерфейсу, не торкаючись інших частин коду (рис. 2). Він надає власний пакет тестування під назвою bloc_test. Реалізує комплексне рішення для тестування будь-якого застосунку Flutter, який використовує пакет BLoC. Тестування за допомогою цього пакета є прямим і добре задокументованим [3].

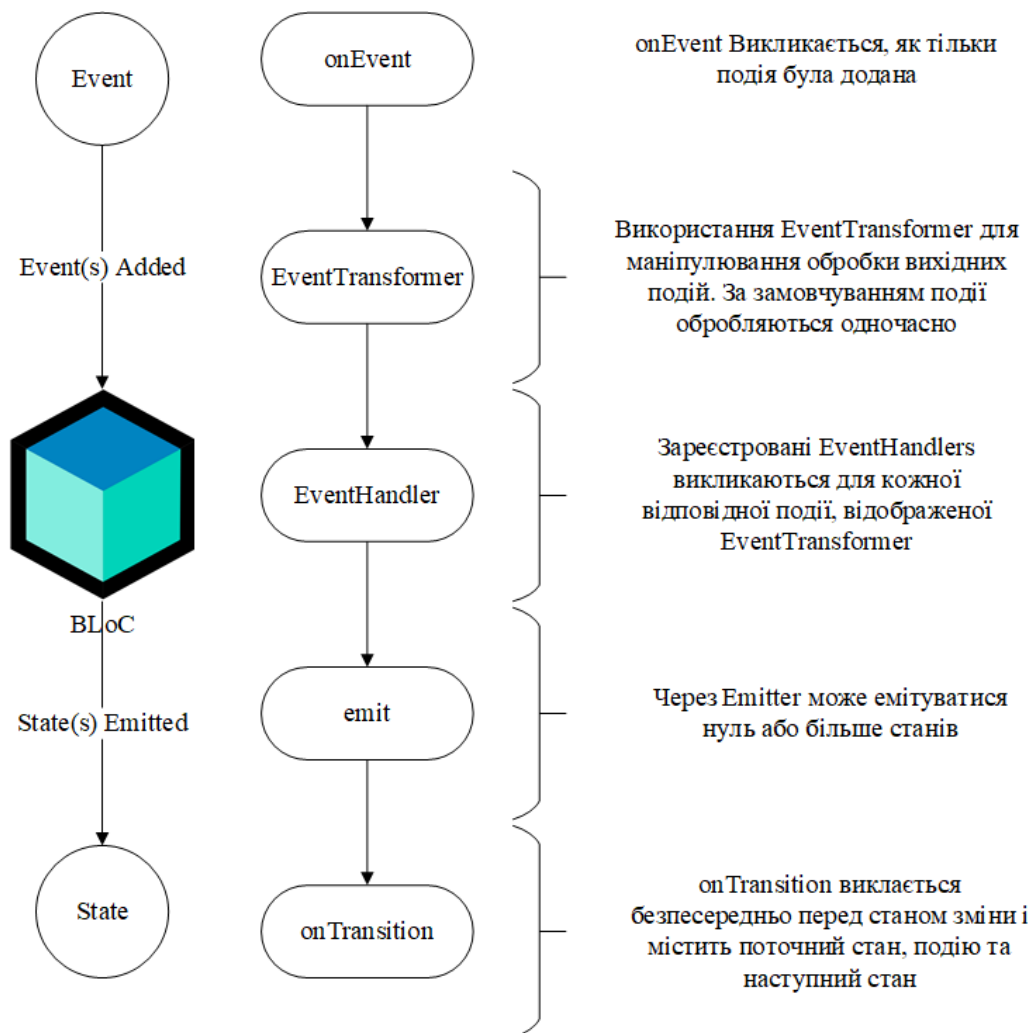


Рис. 2. Графічне представлення роботи BLoC

Перевагами BLoC є:

- відокремлення презентаційного шару від бізнес-шару;
- спрощення тестування застосунків;
- краща продуктивність для великих обсягів даних;
- базування на реактивному програмуванні;
- забезпечення безпечного виконання операцій на різних потоках.

Недоліками BLoC є:

- призначений для невеликих або середніх за розміром проєктах;
- надмірна кількість коду.

BLoC не може бути вторинним пакетом, який використовується для управління станом, оскільки він покладається на шаблонний код. Єдиним варіантом сумісності буде використання BLoC як основного рішення, а GetX або MobX для вторинного. Документація BLoC є повною, з багатьма прикладами для розробників. Він є популярним серед розробників Flutter. Шаблон BLoC може зробити код більш організованим і зручним для підтримки, але вимагає ретельного планування і виконання [4].

MobX – це всеохоплююче рішення для управління станами застосунків (рис. 3). Воно використовує принципи реактивного програмування для управління станами програмних продуктів. Він забезпечує спостережуваний стан, який сповіщає, коли відбуваються зміни. Він вимагає більше кроків установлення через те, що він інтенсивно використовує генерацію коду. Розробнику не потрібно писати багато шаблонного коду, використовуючи MobX, порівняно з BLoC. Тобто весь шаблонний код, який не буде написаним на початку, буде згенерований функцією кодогенератора MobX, тому технічно він все одно вимагає багато коду для запуску. MobX досить добре задокументований, і, незважаючи на всі його складнощі (кодоген і все інше), він простий у налаштуванні [5].

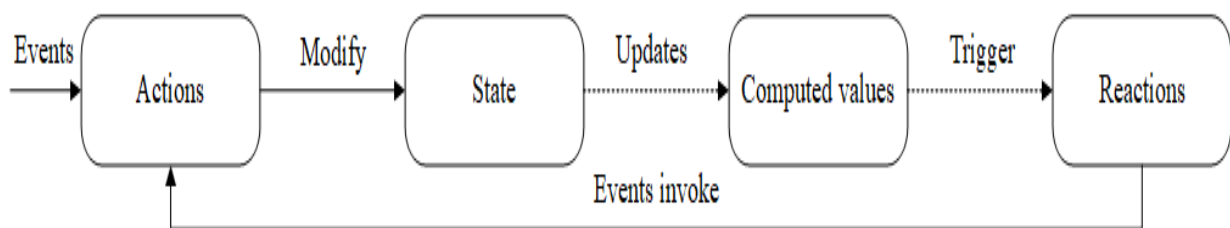


Рис. 3. Графічне представлення роботи MobX

Перевагами MobX є:

- заснований на реактивному програмуванні, що дозволяє легко відстежувати зміни стану та автоматично оновлювати користувальницький інтерфейс;

- потреба меншої кількості коду в порівнянні з іншими станами.

Недоліками MobX є:

- обмежена підтримка для різних платформ;
- недостатньо широка адоптація.

Однак тестувати MobX не так просто, як використовувати його, оскільки розробники не задокументували кроки для тестування Flutter застосунків за допомогою даного фреймворку. Використання MobX для управління станами у Flutter дозволяє більш ефективно і керовано створювати реактивні застосунки з мінімальним шаблонним кодом.

GetX – це інтуїтивно зрозуміле і пряме рішення для управління станом у Flutter. GetX – це пакет, який забезпечує управління стану та ін'єкцію залежностей (рис. 4) тощо. Серед усіх рішень для управління станами, GetX є найпростішим. Розробникам не потрібно розуміти нові концепції або парадигми, щоб працювати з GetX. Тести з використанням GetX добре задокументовані розробником пакета, особливо для управління станом. Його можна використовувати як основне або додаткове рішення, яке буде працювати так само добре. Однак, поєднання GetX з більш складними рішеннями для управління станом, такими як BLoC, може відігравати більш другорядну роль, враховуючи, що BLoC вимагає більше шаблонів для роботи [6].

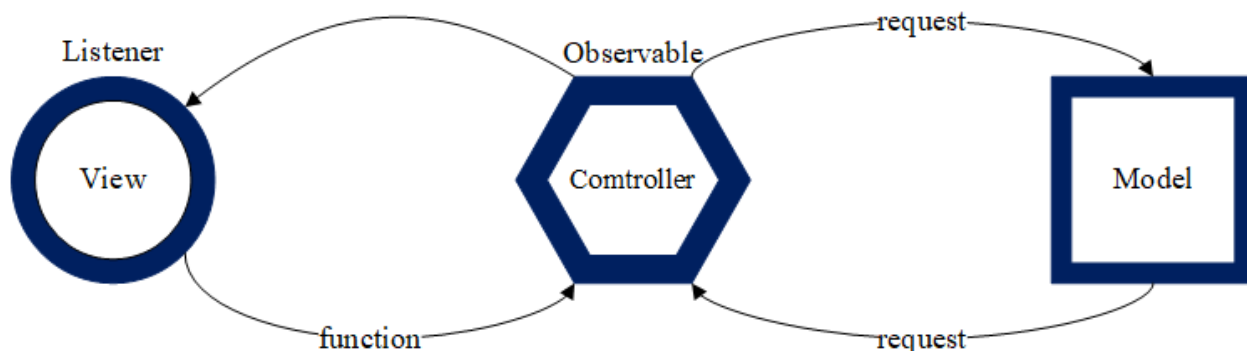


Рис. 4. Графічне представлення роботи GetX

Перевагами GetX є:

- простота та компактність коду;
- висока продуктивність;
- підтримка багатьма платформами;
- підтримка інтуїтивного Application Programming Interface (API).

Недоліками GetX є:

- невелика спільнота;
- нестабільність API;
- не широке розповсюдження як в інших методів управління станом.

Якщо розробник шукає просте в освоєнні рішення для управління станами, яке буде ефективним, варто звернути увагу на GetX. У порівнянні з більш простими рішеннями, такими як setState, GetX вимагає менше шаблонного коду

для налаштування та управління станом застосунку, що робить його чудовим варіантом для початківців та досвідчених розробників.

Redux – це популярна бібліотека управління станами для Flutter застосунків, яка працює за схемою односпрямованого потоку даних (рис. 5). Перевагою Redux, як і в GetX, є можливість використовувати сховище всередині StatelessWidget, усуваючи будь-яку додаткову складність, необхідну для обробки станів. Це допомагає спростити кодову базу і полегшити її підтримку, особливо в міру того, як застосунок зростає в розмірі і складності [7].

Перевагами Redux є:

- забезпечення суворого та передбачуваного потоку управління станом;
- гарна масштабованість для крупних і складних проєктів.

Недоліками Redux є:

- використання централізованого сховища, в якому зберігається весь стан програмного продукту;
- легка інтеграція з інструментами для збереження та відновлення історії стану застосунку;
- не надання вбудованого механізму для управління асинхронними операціями.

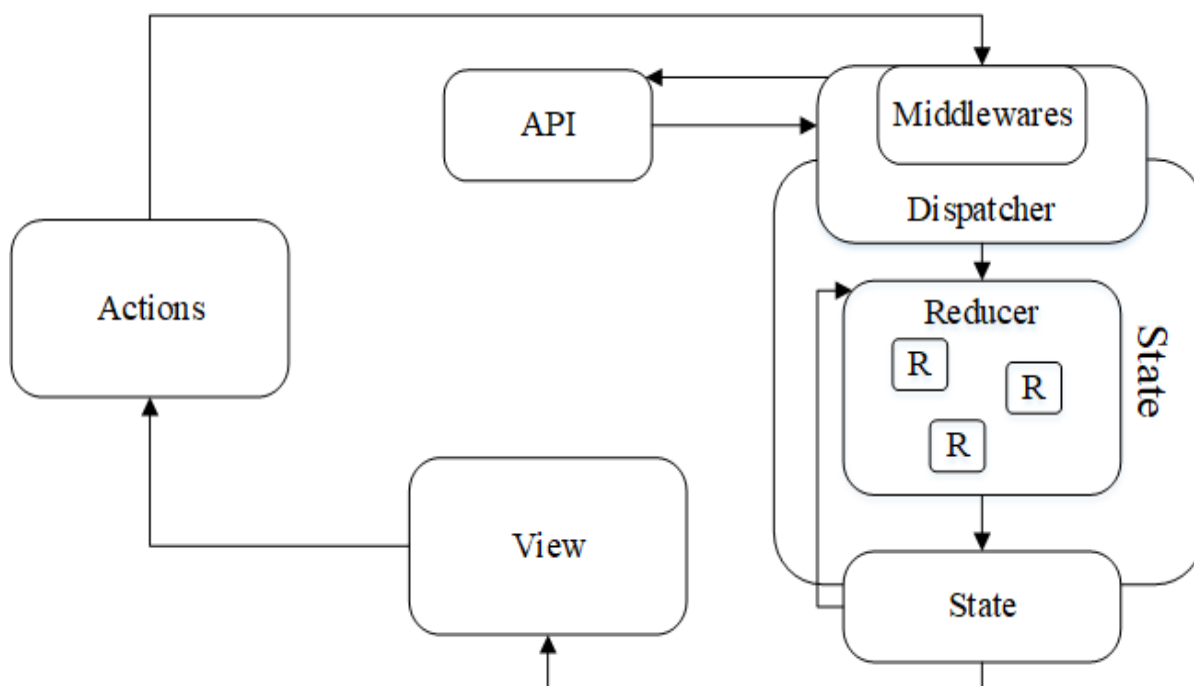


Рис. 5. Графічне представлення роботи Redux

Однак пошук робочої документації про те, як працюватимуть тести, вимагає зусиль. Супровідники flutter_redux надають лише репозиторій на GitHub з деякими прикладами тестів, які допоможуть пройти цей процес. Redux є посереднім пакетом для простоти. Його легко зрозуміти, але він вимагає трохи шаблонного коду для налаштування редукторів і станів. Використання Redux з

Flutter може допомогти спростити кодову базу і полегшити її підтримку в міру зростання розміру і складності програмного продукту. Крім того, Redux чудово підходить для обміну станами між екранами, оскільки він відокремлює логіку стану від інтерфейсу користувача, і він менш складний, ніж BLoC [7-8].

Наукова новизна наданих матеріалів обґрунтовується у спрощенні вибору методу управління станом фреймворку Flutter, надаючи їх найважливіші аспекти, переваги та недоліки кожного.

Висновки. Підсумовуючи, у порівнянні із іншими методами BLoC є найкращим варіантом управління станом у застосунках Flutter. Він має більшу спільноту і може краще підходити для складних проєктів з декількома екранами та потоками даних. Зрештою, рішення для управління станом, яке буде використовуватися у програмному продукті Flutter, буде залежати від потреб самого проєкту.

ПЕРЕЛІК ПОСИЛАНЬ

1. The top 5 flutter state management solutions: a deep dive. Kodeco. URL: <https://www.kodeco.com/39848254-the-top-5-flutter-state-management-solutions-a-deep-dive> (дата звернення: 16.10.2023).
2. Jablečník M. Most popular Flutter state management libraries in 2023. Dev. URL: <https://dev.to/mjablecnik/most-popular-flutter-state-management-libraries-in-2023-amc> (дата звернення: 16.10.2023).
3. Performance analysis of bloc and provider state management library on flutter / R. R. Prayoga et al. *Jurnal mantik*. 2021. Vol. 5, no. 3. P. 1591-1597.
4. Nwogu J., Ackerson D. Understanding state management in flutter. *Semaphoreci*. URL: <https://semaphoreci.com/blog/state-management-flutter> (дата звернення: 16.10.2023).
5. Boukhary S., Colmenares E. A clean approach to flutter development through the flutter clean architecture package. Las Vegas, 5-7 December 2019. 2019. P. 1115-1120.
6. Ameen S. Y., Mohammed D. Y. Developing cross-platform library using flutter. *European journal of engineering and technology research*. 2022. Vol. 7, no. 2. P. 18-21.
7. Taxonomy of cross-platform mobile applications development approaches / W. S. El-Kassas et al. *Ain shams engineering journal*. 2017. Vol. 8, no. 2. P. 163-190.
8. Люлік А. Ю. Засоби розробки мобільних застосунків. *Кібербезпека в сучасному світі*: матеріали III Всеукр. науково-практ. конф., м. Одеса, 19 листоп. 2021 р. С. 127–133.

Л.В. Кабак¹, Б. І. Мороз¹, Я. Сконечный², Ю. А. Сіда¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Технічний університет «Краківська гірничо-металургійна академія ім. Станіслава Сташиця» (AGH), м. Краків, Польща

THE MODERN POPULARITY OF NON-RELATIONAL DATABASES

Annotation: The theses give the reasons why a non-relational database model is becoming popular nowadays. It describes the main advantages of a non-relational model and the trends that have influenced their growth in demand among developers.

Keywords: non-relational databases, Database Management System, database model, NoSql.

Introduction. Nowadays almost every application or website uses databases to store some information, for example, data about their users. Therefore, it is an important moment for developers to make the right choice of database type at the stage of product development. For many years, a relational model has been the leader of the database types, but every year a non-relational model is gaining more and more popularity.

A non-relational database (often called NoSql databases) is a database that does not use the tabular schema of rows and columns found in most relational database systems. Instead, such databases use a storage model that is optimized for the specific requirements of the type of data being stored. For example, data may be stored as simple key/value pairs, as JSON documents, or as a graph consisting of edges and vertices [1].

Analysis of recent research. There is the historical trend of the DBMS(Database Management System) popularity by categories in figure 1. In the ranking of each month, the best three systems per category are chosen and the average of their ranking scores is calculated [2].

So we can see that the popularity of relational databases is table while the favor of certain non-relational types grows.

Let's look at the main reasons why the non-relational model is growing in popularity:

Possibility of distribution. Internet leaders like Amazon, Google, and Facebook created NoSQL (not-only-SQL) as a tool for massively scalable database management systems that could write and read data anywhere in the world while delivering performance and availability to billions of users [3].

So NoSQL database systems are designed to use multiple locations involving different data centers/regions for global scalability. Just today, large Internet resources require such a skill from database systems because of the growing number of users, so the non-relational database model becomes an excellent choice for this.

Performance. Non-relational databases are initially built to have high performance. Such databases often perform faster because a query doesn't have to view

several tables in order to deliver an answer, as relational datasets often do [5]. In addition, we can quickly enter data due to the lack of need to structure the data in the table as in relational databases. Also, the probability of an error during the data entry operation is extremely low, because there is no strong data typing. NoSQL databases can support rapidly developing applications requiring a dynamic database able to change quickly and accommodate large amounts of complex, unstructured data [6]. Therefore, this database model is best used in cases where it's needed to deliver a high speed of work on large amounts of data. Because non-relational databases do not need the same amount of preparatory work that relational databases usually need.

Complete trend, starting with January 2013

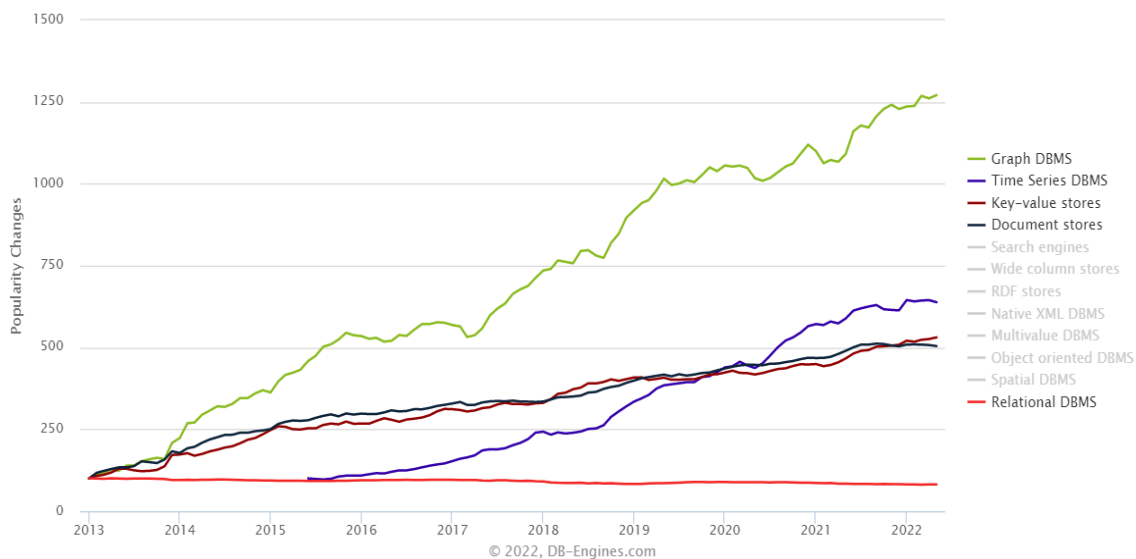


Figure 1. The trend of the categories' popularity

Scaling. The avalanche of data on the World Wide Web has exacerbated the problem of vertical scalability - the computing power of iron can grow indefinitely, and the price of a few simple servers is less than one high-performance. In this situation, horizontal scaling will be a good solution, when several independent machines are connected together and each of them processes only part of the requests. This architecture makes it possible to quickly increase cluster capacity by adding a new server. Relational databases are usually deployed over a single server or node, so they can only scale vertically - by adding more CPU or RAM to the existing infrastructure. So Relational Database Management System supports only [vertical scaling](#), whereas non-relational DBMS supports [horizontal scaling](#). NoSQL databases are distributed by design, so can easily scale vertically - by adding a new machine or server to the system [4]. Therefore, supporting horizontal scaling makes non-relational databases more ambitious for major projects.

Flexibility. Non-relational databases are highly flexible as they can store and combine any type of data. This data can be structured or unstructured. It is a ponderable privilege on relational type which cannot store unorganized data. So NoSQL databases

can store data in native formats, which means developers don't have to adapt the data to the store. Storing data "as is" means not having a front-end ETL system to shoe-horn semi-structured data into row and column formats, and fewer applications to develop or buy to get a new database launched [5]. Besides, non-relational databases allow you to dynamically update the schema. So a developer can change the schema of the database and add new data of any type in the process.

Services. DBMS(Database Management System) is a considerable factor in working with databases because with them we have easy access to the database and all the necessary tools to work with it. In recent years, non-relational DBMS are actively increasing their number, and existing ones are improving the quality of their services.

Therefore, the emergence of convenient and affordable services for working with non-relational databases has also become the reason for the popularity of this model. Because they can provide not only a standard set of services but also a wide range of integration services such as global replication, analytics, data visualization, geolocation, memory performance, and more. So comfortable working with such services and a wide range of tools makes the developer's work process more efficient and less time consuming

Conclusion: Non-relational databases are gaining momentum due to some factors listed above. Indeed, they meet the requirements of modern software development, so most likely we will see more growth in the use of such databases and in the progress of NoSql DBMS to work with them.

References

1. Microsoft Docs: Non-relational data and NoSql. URL: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data>
2. DB-Engines: DBMS popularity broken down by database model. URL: https://db-engines.com/en/ranking_categories
3. DataStax: What is NoSql. URL: <https://www.datastax.com/what-is/nosql>
4. Keboola: Relational vs non-relational database: Which one should you use? URL: <https://www.keboola.com/blog/relational-vs-non-relational-database-when-to-use-one-instead-of-the-other>
5. MongoDB: Advantages of NoSQL database. URL: <https://www.mongodb.com/nosql-explained/advantages>
6. MongoDB: What is a Non-Relational Database. URL: <https://www.mongodb.com/databases/non-relational>

INFORMATION SUPPORT FOR THE PROCESS OF DIAGNOSING THE CONDITION OF METHANE ANALYZERS

Анотація. Робота присвячена дослідженню та розробці інформаційного забезпечення для проведення процесу діагностики стану аналізаторів метану. Запропонований підхід включає використання методів збору, аналізу та інтерпретації даних. Інформаційне забезпечення призначене для автоматизації процесу діагностики, покращення точності вимірювань та раннього виявлення потенційних несправностей в аналізаторах метану.

Keywords: *information support, methods, algorithms, diagnostics, microcontroller, automatic gas protection, methane analyzers*

Introduction. In the modern world, where the efficient use of resources and care for the environment are becoming increasingly important, the issues of control and diagnostics of various processes are of particular importance. Coal deposits in Ukraine have complicated mining and geological conditions, such as insignificant thickness of coal seams, high concentration of gases, propensity to spontaneous combustion, which leads to increased risk of hazardous situations. Therefore, they should be equipped with effective safety monitoring and management systems. In these systems, special attention should be paid to the prevention of explosions of methane-air mixtures, which are one of the main manifestations of risks[1]. One of the critical aspects in this context is the reliability and accuracy of methane analyzers, which are an integral part of the systems for controlling explosive atmospheres in mine workings.

Nowadays, the thermocatalytic method is mainly used in gas analyzers to determine the methane content in the atmosphere of mine workings. However, difficult conditions of use of gas analyzers, such as accidental shocks, water ingress, influence of extreme parameters and other factors, as well as unauthorized interference in their operation, can affect the efficiency and stability of these devices[2]. This can lead to failures in the functioning of explosion protection means, which is one of the causes of emergencies in mine workings. The use of information support for controlling the state of methane analyzers will significantly improve the reliability and stability of gas analyzers, which reduces the probability of failures and accidents associated with inadequate control of gases in mine workings.

Modern information support technologies provide unique opportunities for more effective diagnostics of methane analyzers condition. The use of analytical software solutions can quickly identify potential problems, thereby improving the overall reliability and stability of measurements[3]. In this article we will consider how information support can improve the efficiency of the process of diagnosing the condition of these analyzers, which contributes to more accurate control of production processes and reduction of risks associated with the development of mineral deposits.

Main Content. One of the reasons for unstable operation of methane analyzers is the zero shift of the measuring bridge due to external influences. The use of microcontrollers in monitoring and control systems allows to carry out diagnostics of zero of thermocatalytic methane sensors in the presence of necessary information support. When using microcontrollers there appears a possibility to control thermocatalytic methane sensor, what allows to automate the process of diagnostics and correction of zero readings of thermocatalytic methane analyzers. This is possible because the dependence of the voltage on the thermocatalytic sensor thermocell on the current value is unambiguously described by the following theoretical function [1]

$$U_e = I_e R_{e2} + \beta b_e R_{0e} I_e^3. \quad (1)$$

where U_e - voltage on the thermocouple, V; I_e - current through the thermocouple, A; R_{e2} - resistance of the thermocouple at gas temperature, Ohm; β - temperature coefficient of resistance, $1/^\circ\text{C}$; b_e - thermoresistive coefficient of the thermocouple, $^\circ\text{C}/\text{A}^2$; R_{0e} - resistance of the thermocouple at 0°C , Ohm

The thermoresistive coefficient describes the relationship between the electrical, thermal and geometric characteristics of the thermocouple and remains a constant value. The resistance of the platinum resistor of the thermocouple at a certain gas temperature is calculated using a known expression.

$$R_{e2} = R_{0e}(1 + \beta t_2), \quad (2)$$

where t_2 – gas temperature, $^\circ\text{C}$.

Taking into account expressions (1) and (2), the voltage on the series connected working p and comparative n elements, respectively, is defined as follows:

$$U_p = I_e R_{0p}(1 + \beta t_2) + \beta b_p R_{0p} I_e^3, \quad (3)$$

$$U_n = I_e R_{0n}(1 + \beta t_2) + \beta b_n R_{0n} I_e^3. \quad (4)$$

In production conditions, absolute identity of voltampere characteristics of working and comparison elements is not achievable. In this connection, when using traditional schemes of connecting the thermocatalytic sensor to the measuring bridge, provided that the current through the thermocouple corresponds to the operating mode of the thermogroup, the zero of the measuring bridge is set by changing the resistance of ballast resistors in the comparative branch of the bridge. Reduction of current through the elements to the level at which the reaction of thermocatalytic oxidation of methane on the working element stops, due to the mismatch of voltampere characteristics of the working and comparative elements, causes a shift of zero of the measuring bridge, which complicates the process of automatic diagnostics of this parameter.

At the current through the thermocouples, denoted as I_e and corresponding to the operating mode of the sensor, due to the divergence of the voltampere characteristics of these elements, there is a voltage difference between them, denoted as ΔU . According to equations (3) and (4), in this case this difference is as follows:

$$\Delta U = U_p - U_n = I_e(R_{0p} - R_{0n})(1 + \beta t_z) + \beta I_e^3(b_p R_{0p} - b_n R_{0n}). \quad (5)$$

When the current is reduced to the level I'_e , at which the reaction of methane oxidation on the working element does not occur, the voltage difference on these elements is equal to

$$\Delta U' = U'_p - U'_n = I'_e(R_{0p} - R_{0n})(1 + \beta t_z) + \beta I_e'^3(b_p R_{0p} - b_n R_{0n}). \quad (6)$$

From the analysis of equations (5) and (6) it follows that the change of ΔU can occur due to fluctuations of gas temperature, as well as due to instability of resistance of thermocouples or their thermoresistive coefficients. Let us represent expressions (5) and (6) in the form of

$$\Delta U = I_e a + I_e^3 b, \quad (7)$$

$$\Delta U' = I'_e a + I_e'^3 b. \quad (8)$$

where $a = (R_{0p} - R_{0n})(1 + \beta t_z)$; $b = \beta(b_p R_{0p} - b_n R_{0n})$.

Having information about the values of element resistances obtained in the analyzers debugging mode, it is possible to calculate the coefficient a . In the analyzers zero readings check mode, based on certain current values and taking into account the gas temperature, it is possible to calculate the value of coefficient a by software and, based on equation (8), accurately determine the actual value of coefficient b . This, in turn, makes it possible to check the stability of the analyzer zero readings automatically during the analyzers operation, taking into account equation (7), and to correct these readings when they change within predetermined limits.

Conclusion. Information support of the process of methane analyzers condition diagnostics is a key tool in ensuring the efficiency and safety of industrial processes. The investigated method is based on reducing the supply voltage of the thermocouple to the level at which there is no methane oxidation reaction on the working thermocouple, provides an opportunity for automated diagnostics of the state of stationary thermocatalytic methane analyzers. This provides rapid identification of potential problems, reducing the likelihood of measurement failures and ensuring stability in industrial processes. It was also found that the use of modern microprocessors allows not only to automatically check the stability of zero readings of analyzers, but also to correct these readings in case of their change.

REFERENCES

1. Alekseev M.O., Holinko O.V. Automatic Diagnostics of Stationary Gas Analyzers Thermocatalytic / Collection of Research Papers of the National Mining University, 2018 – № 53 – p. 223-229
2. Milan Sága, Michal Bartoš, Vladimír Bulej, Ján Stanček, Dariusz Wiecek. Development of an Automated Diagnostic and Inspection System Based on Artificial Intelligence Designed to Eliminate Risks in Transport and Industrial Companies / Transportation Research Procedia 55 (2021) 805–813.
3. K. M. Shynkaruk, S. A. Chehovski, N. M. Pindus .Experimental studies of the thermocatalytic method for measuring the heat of combustion of natural gas /: Methods and Devices of Quality Control. No. 1(46) (2021)

УДК 004.65

С.Д. Приходченко¹, А. Дерен², А.А Мартиненко¹, М.А. Батальский¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Вроцлавський технологічний університет, Польща

JDBC DATABASE INTERACTION METHOD

Abstract. The article provides an overview of the Java Database Connectivity (JDBC) API. It considers the main components of JDBC architecture and its role in enabling Java programs to interact with databases. The practical use of JDBC is examined through common tasks like establishing connections, executing queries, processing results, and handling errors.

Keywords: *JDBC, Java Database Connectivity, database, SQL, Java.*

Introduction. JDBC (Java Database Connectivity) is a standard Java API for connecting to databases and executing SQL queries. It allows Java programs to access databases in a unified manner. JDBC establishes a connection between the Java application and the database, sends SQL statements and receives results.

Main Content. In a typical JDBC application, the steps involve loading the appropriate JDBC driver, establishing a connection to the database, creating statements for SQL queries, executing queries, processing the results, and handling exceptions appropriately. The architecture provides a standardized way for Java applications to interact with various databases seamlessly.

- JDBC API - set of interfaces used by the application
- JDBC Driver - implements the interfaces and interacts with a specific database

In the JDBC (Java Database Connectivity) architecture, the JDBC Manager is not explicitly referred to as a distinct component. However, what I believe you might be referring to is the role played by the DriverManager class in JDBC. Let me elaborate on this aspect.

The DriverManager class manages the drivers. It establishes connections and handles communication.

Common JDBC tasks include:

- Loading the JDBC driver using `Class.forName()`
- Opening connection to the database via `DriverManager.getConnection()` by providing url, username and password
- Creating `Statement`, `PreparedStatement` and `CallableStatement` objects to execute SQL queries
- Using `executeQuery()` for `SELECT` queries which return a `ResultSet`
- Using `executeUpdate()` for `INSERT`, `UPDATE`, `DELETE` queries
- Processing query results and metadata from `ResultSet`
- Handling `SQLException` errors and warnings
- Committing or rolling back transactions
- Closing `ResultSet`, `Statements` and `Connection` objects via `close()` method

JDBC drivers serve as essential components in the Java Database Connectivity (JDBC) architecture, facilitating communication between Java applications and various database systems. These drivers play a crucial role in translating the standardized JDBC calls into database-specific commands, enabling seamless interaction.

The selection of a specific JDBC driver type depends on factors such as performance requirements, database system compatibility, and the desired level of independence from external dependencies. Each driver type has its strengths and considerations, allowing developers to choose the most suitable option for their specific application and database environment.

JDBC drivers:

- JDBC-ODBC bridge (Type 1) - ODBC calls translated into JDBC
- Native API (Type 2) - Access database via native libraries
- Network protocol (Type 3) - Use network protocol to interact with middleware
- Database protocol (Type 4) - Directly interact with the database

Advanced features in JDBC enhance the capabilities of Java applications when interacting with databases. These features go beyond basic database operations and offer additional functionalities for improved performance, security, and flexibility.

These advanced features empower Java developers to implement sophisticated database interactions, optimize performance, and enhance the overall robustness and security of their applications. Each feature addresses specific requirements, providing flexibility and scalability in database-driven Java applications.

Advanced features:

- Retrieving database metadata via `DatabaseMetaData`
- `PreparedStatement` avoids SQL injection, improves performance
- `CallableStatement` to call stored procedures
- Savepoints and transaction control
- `RowSet` to use data offline
- `DataSource` objects for connection pooling

The **Connection** interface in JDBC serves as a fundamental component for establishing and managing connections between a Java application and a database. This interface encapsulates the essential operations involved in the interaction with a database.

The **Connection** interface plays a central role in JDBC, providing a unified interface for establishing, managing, and interacting with a database. Its versatility enables developers to perform a range of operations crucial for database-driven Java applications, ensuring efficient and secure communication between the application and the database system.

Connection Interface:

- Represents a connection to the database.
- Established using the `DriverManager.getConnection(url, username, password)` method.
- Handles operations like establishing, managing, and closing the connection.

The **Statement** interface in JDBC is a key component that facilitates the execution of SQL queries and updates against a database. It provides methods for creating, executing, and managing SQL statements within a Java application.

The **Statement** interface is a fundamental element in JDBC, providing a means for developers to execute SQL queries and updates in a straightforward manner. Its versatility, supporting both general-purpose and precompiled statements, makes it a central component in the development of database-driven Java applications.

Statement Interface:

- Used to execute SQL queries and updates.
- Two main types: `Statement` for general-purpose queries and `PreparedStatement` for precompiled queries, providing better performance and security.

The **ResultSet** interface in JDBC is a crucial component that represents the result set of a database query. It provides methods for traversing and manipulating the data retrieved from the database.

The **ResultSet** interface plays a central role in JDBC, serving as a container for query results and offering a variety of methods for navigating, extracting, and manipulating data. Its versatility and functionality make it an indispensable part of database interactions in Java applications.

ResultSet Interface:

- Represents the result set of a database query.
- Allows traversal and manipulation of the result set.
- Essential methods include `next()`, `getInt()`, `getString()`, etc.

The basic steps for JDBC database interaction provide a systematic approach for Java applications to connect to a database, execute queries, and process results.

These basic steps below provide a foundational framework for Java applications to interact with databases using JDBC. Developers can customize these steps based on

specific requirements and expand upon them to incorporate additional features or functionalities as needed.

Conclusion. JDBC is a cornerstone for Java developers when it comes to working with databases. Its versatility and standardized approach make it a powerful tool for creating dynamic and data-driven applications. As you progress in your journey as Java developers, mastering JDBC will undoubtedly enhance your ability to build robust and efficient database interactions in your applications.

REFERENCES

1. "Java Database Best Practices". George Reese. May, 2003.
2. "Java Database Programming with JDBC". Pratik Patel, Karl Moss. 1996.
3. "JDBC: Practical Guide for Java Programmers". Gregory D. Speegle. 2002.
4. "Database Programming with JDBC & Java, Second Edition". George Reese. August, 2000.
5. "Java Persistence with Hibernate". Christian Bauer and Gavin King. October, 2006.

РОЗДІЛ 2

ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ

УДК 004.415.3:681.6

М.О. Алексєєв¹, О.М. Алексєєв¹, Є.С. Голобояр¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

АРХІТЕКТУРНЕ РІШЕННЯ В МОБІЛЬНИХ ДОДАТКАХ ПІД ПЛАТФОРМУ IOS ЧЕРЕЗ ІНТЕГРАЦІЮ SWIFTUI

Анотація. Описано процес розробки програмних продуктів за допомогою фреймворку SwiftUI. Підвищення швидкості побудови мобільних додатків досягається за рахунок будівництва найбільш ефективних архітектурних рішень в мобільних додатках під платформу iOS через інтеграцію фреймворку SwiftUI.

Ключові слова: *iOS, SwiftUI, Xcode, MVC, MVVM, VIPER, FLUX, UIKit, architecture, mobile apps.*

Вступ. Основними гравцями на ринку смартфонів продовжують залишатися компанії Apple та Google з відповідними операційними системами iOS і Android. Кожна система має свої переваги та недоліки, iOS позиціонується як закрита система з максимальним захистом даних користувачів, у свою чергу Android базується на відкритій кодовій базі та пропонує набагато більше кастомізацій пристроїв, але з більшими ризиками до вразливостей.

Розробка мобільних додатків пройшла довгий шлях, зазнавши багато змін. Якщо брати платформу iOS, то перші програми писалися мовою програмування Objective-C, також розробники мали вручну обробляти керування пам'яттю. У 2014 році з'явився Swift, який швидко витіснив Objective-C, здобувши високу популярність серед розробників завдяки багатьом перевагам, таким як: суворі статична типізація, велика кількість засобів функціональної парадигми програмування, відкритий код, підтримка опціональних типів та дженериків, надзвичайно швидкий розвиток. Наступним значним кроком було представлення компанією Apple фреймворку SwiftUI, якому і присвячена дана робота. Основним аспектом даного фреймворка є повний перехід на декларативну модель програмування, коли інтерфейс користувача є функцією від стану, відповідно він завжди буде відображати правильний стан при однакових вхідних параметрах. Якщо раніше розробники довгий час використовували імперативний підхід, вручну оновлюючи інтерфейс при оновленні даних, то тепер інтерфейс завжди прив'язаний до даних і саму програму не можна будувати іншим чином. Фактично сам фреймворк заохочує розробників писати більш надійний код.

Для аналізу ефективності фреймворку SwiftUI було вибрано розробку прототипу мобільного додатка Photogram для обміну інформацією у вигляді фото

між користувачами. Додаток було побудовано на базі архітектури MVVM (Model View ViewModel), у той же час було проаналізовано інші типові архітектури для створення додатків для платформи iOS: MVC, VIPER, Flux.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- порівняння мов програмування Objective-C та Swift;
- порівняння фреймворку UIKit і SwiftUI;
- огляд архітектур під час написання додатків для системи iOS;
- порівняння ефективності SwiftUI та UIKit для побудови елементів інтерфейсу користувача;
- порівняння механізму оновлення інтерфейсу користувача зі зміною стану у SwiftUI та UIKit;
- зробити висновки щодо доцільності ефективності фреймворку SwiftUI.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти.

- порівняльний аналіз типових архітектур побудови мобільних додатків;
- функціональне програмування;
- об'єктно-орієнтоване програмування.

Було розглянуто побудову мобільного додатка PhotoGram з використанням SwiftUI та патерну проектування MVVM. Були висвітлені архітектурні засади побудови додатка, порівняння ефективності SwiftUI та UIKit для побудови елементів інтерфейсу користувача, а також порівняно механізми оновлення інтерфейсу користувача із зміною стану у SwiftUI та UIKit.

Архітектурні засади побудови додатка PhotoGram засновані на патерні проектування MVVM (Model-View-ViewModel). Ця архітектура розділяє додаток на три основні компоненти: Model, View та ViewModel.

Компоненти Model відповідають за роботу з даними додатка. Вони можуть включати структури даних, що пов'язані з коментарями, сповіщеннями, постами та користувачами. Для збереження та синхронізації даних з сервером може використовуватись Firebase або інші відповідні інструменти.

View компоненти в SwiftUI містять у собі блоки елементів інтерфейсу користувача та суцільні екрани. SwiftUI надає зручний спосіб побудови інтерфейсу за допомогою декларативного підходу. Екрани в SwiftUI будуються за допомогою елементу View, який може включати різноманітні візуальні елементи, такі як кнопки, тексти, зображення та інші.

Компоненти ViewModel відповідають за логіку та управління додатком. Вони використовують комбінацію Swift Combine для отримання, обробки та оновлення даних зі збережених репозиторіїв. ViewModel служать посередниками між Model та View, надаючи необхідні дані та забезпечуючи реагування на дії користувача.

Порівняння ефективності SwiftUI та UIKit показало, що SwiftUI має декілька переваг у порівнянні з UIKit. SwiftUI надає більш простий та зручний спосіб побудови інтерфейсу, оскільки використовує декларативний підхід, де програміст описує бажаний стан інтерфейсу, а не вказує кроки для його побудови. В SwiftUI також доступна багата бібліотека вбудованих елементів інтерфейсу, що спрощує розробку. Однак на даний момент SwiftUI може мати деякі обмеження порівняно з UIKit, особливо щодо розширеної функціональності, яку пропонує UIKit.

Порівняння механізму оновлення інтерфейсу користувача показало, що SwiftUI та UIKit мають різні підходи до оновлення інтерфейсу. У SwiftUI оновлення інтерфейсу користувача відбувається шляхом зміни стану (state) відповідних елементів View. Під час зміни стану SwiftUI автоматично оновлює відповідні частини інтерфейсу. У UIKit оновлення інтерфейсу вимагає прямого програмного коду для зміни властивостей елементів інтерфейсу.

Загалом, побудова мобільного додатка PhotoGram з використанням SwiftUI та MVVM виявляється ефективним підходом. SwiftUI спрощує розробку інтерфейсу користувача за рахунок декларативного підходу та наявності вбудованих елементів інтерфейсу. MVVM надає чітку структуру та розділення відповідальностей між компонентами додатка. Однак, перед вибором SwiftUI для побудови додатка необхідно врахувати його обмеження порівняно з UIKit та оцінити, наскільки важлива розширена функціональність для конкретного проєкту.

Було проведено детальний аналіз створення мобільного додатка PhotoGram за допомогою SwiftUI та архітектурного патерну MVVM. Приділяючи особливу увагу розробці користувацького інтерфейсу, використовувався декларативний інструмент розробки SwiftUI, який значно полегшує створення інтуїтивно зрозумілого та зручного користувацького інтерфейсу.

Впровадження Firebase як надійної хмарної платформи дозволило створити масштабовану та безпечну базу даних, що забезпечує стабільність додатка та полегшує впровадження майбутніх оновлень.

Основною частиною розділу стало дослідження можливостей для подальшого покращення додатка Photogram. Було визначено ключові області для додавання нових функцій, включаючи обмін повідомленнями, можливість ділитися фото та відео, а також інтеграцію з іншими соціальними мережами. Для реалізації цих цілей було запропоновано використання ряду технологій, включаючи Firestore та Firebase Storage.

У контексті ринку праці та нових проєктів ми дійшли висновку, що SwiftUI та UIKit активно використовуються розробниками. SwiftUI поступово набирає обертів через свою декларативність та зручність використання, незважаючи на значну роль UIKit у виконанні більш складних та специфічних завдань.

Отже, отримані результати підтверджують ефективність використання комбінації SwiftUI, MVVM та Firebase для створення сучасних мобільних

додатків. Вони також вказують на значні можливості для подальшого розвитку та покращення цих додатків.

Наукова новизна розробки складається в дослідженні ефективності фреймворку SwiftUI та виборі оптимальної архітектури для побудови мобільних додатків.

Висновки. Основна перевага SwiftUI полягає в його простому та зрозумілому синтаксисі. Розробники можуть використовувати декларативний підхід до побудови інтерфейсу, описуючи, як повинен виглядати інтерфейс, а не як його побудувати крок за кроком. Це дозволяє скоротити кількість написаного коду та полегшує розуміння структури додатка. SwiftUI надає миттєве оновлення інтерфейсу. Це означає, що будь-які зміни в коді негайно відображаються на екрані, що спрощує процес тестування та налагодження. Розробники можуть вносити зміни та бачити результат миттєво, що дозволяє ефективно використовувати час та зусилля.

У результаті порівняння між фреймворками SwiftUI та UIKit для написання окремих компонентів було виявлено, що SwiftUI є більш ефективним підходом. Завдяки декларативному синтаксису SwiftUI можна зменшити кількість написаного коду для розміщення об'єктів у користувацькому інтерфейсі. Це дозволяє прискорити процес розробки та полегшити його розуміння.

ПЕРЕЛІК ПОСИЛАНЬ

1. Sahar, A., Clayton, C. «iOS 13 Programming for Beginners: Get started with building iOS apps with Swift 5 and Xcode 11» / A. Sahar, C. Clayton. – Birmingham: Packt Publishing, 2020. – 822p.
2. Head First. Патерни проектування / Ерік Фрімен, Елізабет Робсон, Кеті Сьерра і Берт Бейтс; пер. з англ. Г. Якубовська – Харків : ВД «Фабула», 2020. – 672 с.
3. Odersky, M. «Programming in Scala: A Comprehensive Step-by-Step Guide» / M. Odersky, L. Spoon, B. Venners. – 3rd ed. – California: Artima Inc, 2016. – 859 p.
4. McCarthy, J. «Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I» / John McCarthy. – Massachusetts Institute of Technology, 1960. – 184 p.
5. Berendregt, H. «The polymorphic lambda calculus» / Henk Berendregt. – In: Information and Computation, Volume 110, Issue 2, 1993. – pp. 257-289.

В.М. Куваєв¹, Л.І. Мещеряков¹, А.Т. Харь¹, А.І. Політов²

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Дніпровський Національний університет імені Олеся Гончара, Дніпро, Україна

ОРГАНІЗАЦІЯ ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ ПІДТРИМКИ ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ КЕРУВАННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ

Анотація. Описані особливості підходів до реалізації людино-машинного інтерфейсу підтримки технічного забезпечення систем керування технологічним процесом, що дають можливість скоротити час пошуку порушень які виникають при експлуатації таких систем та скоротити час настроювання систем при заміні датчиків що встановлюються в технологічній лінії.

Ключові слова: системи керування технологічним процесом, людино-машинний інтерфейс, технічне забезпечення.

Вступ. На поточний час найбільш широке застосування при розробці людино-машинних інтерфейсів систем керування технологічним процесом знайшли програмні засоби SCADA [1,2,3]. Таки системи надають широкий спектр інструментарію для програмування людино-машинного інтерфейсу, але вони орієнтовані за своїм функціоналом на операторів-технологів, що безпосередньо керують технологічним процесом, залишаючи поза уваги питання зручностей експлуатації таких систем, зокрема їх технічного забезпечення. У той же час достатньо типовими порушеннями у таких системах є пошкодження з'єднань їх складовими частинами, вихід з ладу датчиків, що встановлені безпосередньо у технологічній лінії. Да і самі датчики потребують періодичного обслуговування з наступним настроюванням у місці їх встановлення у технологічній лінії. Все це призводить до непродуктивних витрат часу і додаткових простоях технологічної лінії

Постановка задачі. Виходячи з викладеній у вступі проблемі виникає задача у необхідності організації людино-машинного інтерфейсу підтримки технічного забезпечення систем керування технологічним процесом для скорочення часу пошуку порушень які виникають в процесі експлуатації таких систем та скорочення часу настроювання таких систем при заміні датчиків що встановлюються в технологічній лінії

Основний зміст роботи. Використання комп'ютерних технологій у системах, що призначені для керування технологічними процесами, дозволяють інтегрувати у них цілу низку функцій, що раніше були притаманні різним системам автоматизації, та отримати синергетичний ефект від такого об'єднання. У той же час суттєво зросла складність таких систем і, відповідно, вимоги до організації у них людино-машинного інтерфейсу. Перш за все, це торкнулося операторів технологічних процесів, що є достатньо логічним, оскільки від якості настроювання і контролю технологічного процесу залежить його продуктивність, якість продукції

і своєчасність виявлення порушень як режимів роботи обладнання, так і його стану. Тем не менш, як питання скорочення часу локалізації і усунення таких порушень, так і питання настроювання параметрів елементів апаратно-програмного забезпечення таких систем, наприклад, після їх заміни у ремонті, залишилися поза увагою розробників програмного інструментарію HMI SCADA-систем, які і користувачів цього продукту.

Проблема створення зручного людино-машинного інтерфейсу для налагодження і супроводження технічного забезпечення складної комп'ютерної системи, критичної до режиму реального часу, вирішувалася при впровадженні і супроводженні комп'ютерної системи керування швидкісним режимом прокатки сортової лінії дрібносортно-дротового стана ДСДС 250/150 ПАТ «АрселорМіттал Кривий Ріг».

З функціональною структурою і основними програмними рішеннями цієї системи можна ознайомитися в [4]. Людино-машинний інтерфейс для супроводження технічного забезпечення комп'ютерної системи керування швидкісним режимом прокатки створювався як окрема підсистема, доступ до якої здійснювався з HMI оператора технолога.

Загальний вигляд людино-машинного інтерфейсу для супроводження технічного забезпечення комп'ютерної системи керування швидкісним режимом прокатки сортової лінії клітей стана ДСДС 250/150 ПАТ «АрселорМіттал Кривий Ріг» наведено на рис. 1, а чистових груп клітей дрібносортного стана ДСС 250-2 того ж підприємства – на рис. 2. Структури меню людино-машинних інтерфейсів для супроводження технічного забезпечення цих систем ідентичні — вони виконані у вигляді дворівневого блокноту у якому права сторінка є ведучою і вміщає перелік модулів з прив'язкою до обчислювальних вузлів комп'ютерної системи, в яких вони встановлені, та перелік параметрів, що доступні до настроювання. Ліва сторінка є підлеглою вибраному пункту меню на правій сторінці, відображає інформацію і вміщає елементи що потрібні для встановлення відповідних параметрів системи або перевірки її зовнішніх ланцюгів підключення датчиків, органів керування, виконуючих механізмів тощо.

Так, рис.1 ілюструє елементи меню, що використовуються для налаштування параметрів встановлення датчиків прогину прокату, який використовується для контролю відхилення прокату від осі прокатки і стабілізації цього відхилення в положення, що задано оператором.

Для коректної роботи системи у кожному міжклітьовому проміжку, де прокатка ведеться з автоматичною стабілізацією прогину прокату, треба знати положення осі прокатки, тобто положення прокату що відповідає «нульовому» прогину, і положення дна передавального жолоба у перетині контролю положення прокату датчиком прогину – максимально можливому прогину прокату у міжклітьовому проміжку. Датчики встановлюються безпосередньо поблизу руху прокату, працюють у важких умовах і захищені спеціальними кожухами. Вони періодично знімаються для профілактики при проведенні штатних ремонтних робіт на прокатному стані.

Наявність сервісу по встановленню «нульового прогину» у людино-машинному інтерфейсу технічного супроводження системи спрощує процедуру налаштування положення датчика прогину при його встановленні на штатне місце. Так відкривши пункт меню «Встановлення нульового прогину» і перейшовши на сторінку «Налаштування» можна безпосередньо контролювати код широтно-імпульсному модульованому (ШІМ) сигналу, який надходить від датчику прогину прокату і який пропорціональний величині прогину прокату. Встановивши на дно передавального жолоба імітатор гарячого прокату, корегують кут нахилу датчика у вертикальній площині таким чином, щоб код ШІМ, що надходить з датчика, був близьким до максимально можливого значення. Після завершення регулювання кута нахилу датчика, він фіксується, а поточний код ШІМ заносять у стовбець таблиці, який відповідає датчику, що налаштовується, у рядок «Дно прогину». Потім встановлюють імітатор у положення осі прокатки і код ШІМ датчика прогину заносять у той ж стовбець, рядок «Корекція осі прокатки». Таким чином, наявність цього сервісу значно спрощує і скорочує час налаштування параметрів встановлення датчиків прогину.

За допомогою інтерфейсу технічного супроводження системи автоматизації можуть бути безпосередньо проконтрольовані вхідні сигнали, що надходять у систему. Так, наприклад, сигнали ШІМ з датчиків прогину прокату можна також проконтролювати через меню технічного супроводження системи до входів модулів вводу відповідних сигналів. На рис.2 наведені відеограми входів матриці №3 модуля UNIO95-5, що запрограмована на введення ШІМ-модульованих сигналів. Доступ до цієї інформації здійснюється вибором на сторінці «Найменування» пункту «Пристрій-модуль-матриця» відповідного елемента з наступним переходом на підлеглу сторінку.

Інша, матриця №4 модуля UNIO95-5, запрограмована на вивід дискретних сигналів. Оскільки операція ручного виводу сигналів може бути небезпечною, то вона є доступною (активною) тільки в режимі «теплого» резерву роботи пристрою при перевірці вихідних ланцюгів, а назви параметрів надані червоним кольором.

Запропонований комплекс рішень по людино-машинному інтерфейсу технічного супроводження складних інформаційно-керуючих систем автоматизації дозволив скоротити час і спростити обслуговування системи автоматизації як під час вводу в експлуатацію, так і в процесі її подальшого супроводження.

Наукова новизна полягає у обґрунтуванні рішень по організації людино-машинного інтерфейсу підтримки технічного забезпечення систем керування технологічним процесом які скорочують час пошуку порушень в технічному забезпеченні і настроюванні датчиків технологічної автоматизації після ремонту що запобігає непродуктивних втрат часу на обслуговування таких систем.

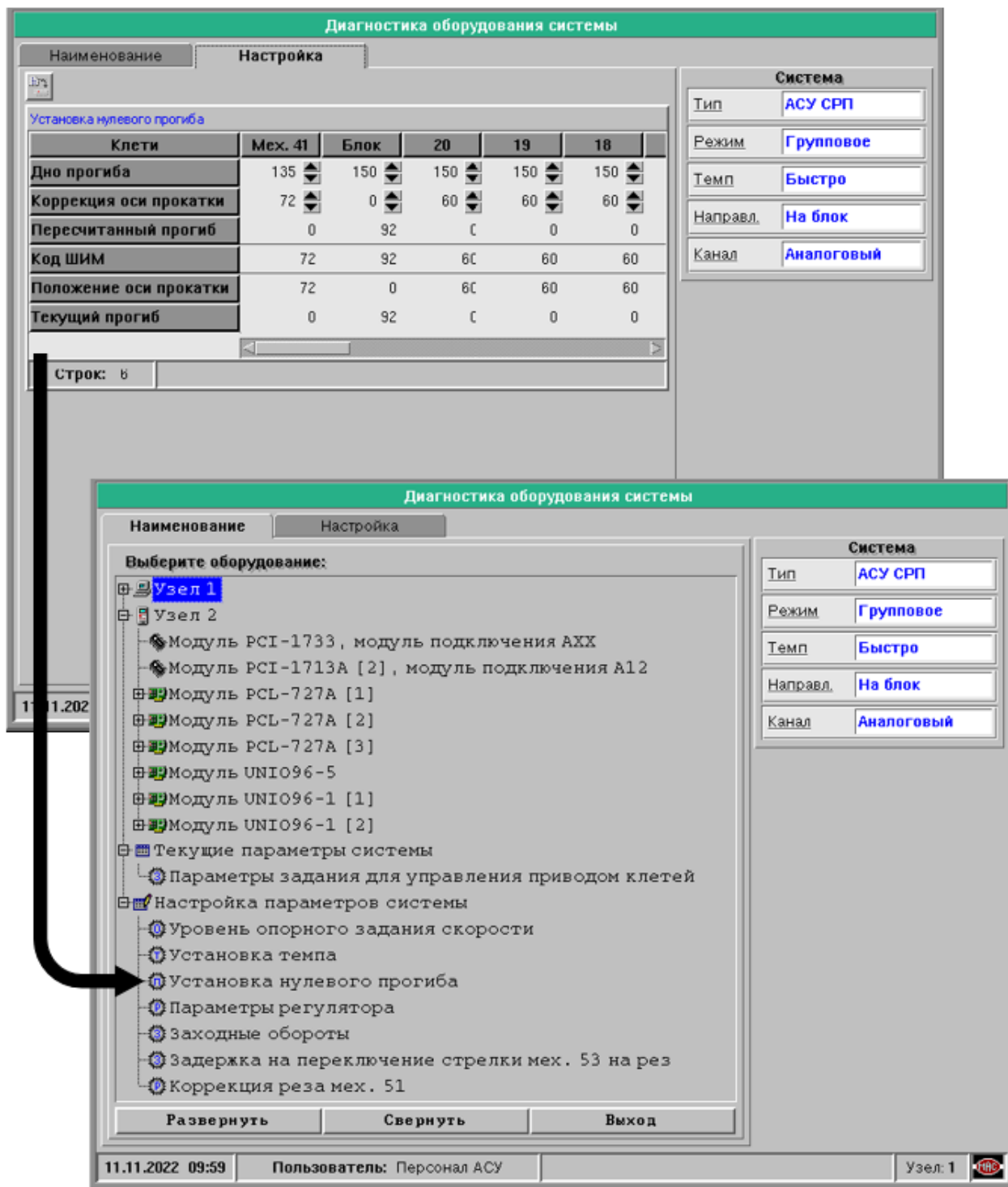


Рис. 1. Людино-машинный интерфейс для супроводження технічного забезпечення комп'ютерної системи керування швидкісним режимом прокатки сортової лінії дрібносоротно-дротового стана ДСДС 250/150 ПАТ «АрселорМіттал Кривий Ріг». Налаштування параметрів встановлення датчиків прогину прокату

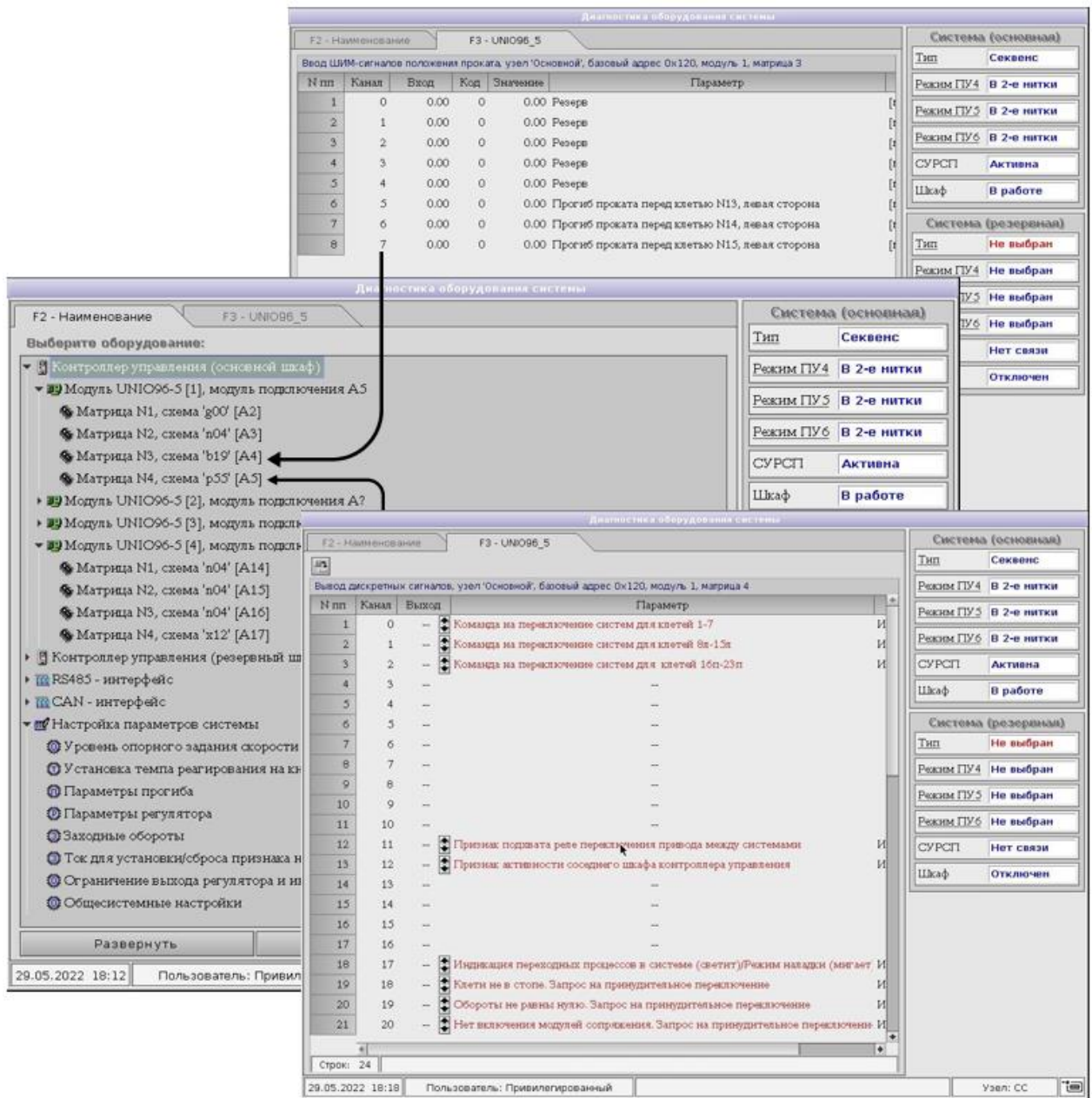


Рис. 2. Відеограми входів матриці №3 модуля UNIO95-5, що запрограмована на введення ШІМ-модульованих сигналів

Висновки. При розробці складних систем керування технологічним процесом слід передбачати крім традиційного людино-машинного інтерфейсу оператора-технолога ще й людино-машинний інтерфейс підтримки технічного забезпечення цієї системи, що скорочує витрати часу на її налагодження і супроводження. Такий інтерфейс повинен забезпечувати швидкий і зручний доступ до вхідних/вихідних сигналів з можливостями їх безпосереднього зчитування/видачі через модулі введення/виведення з/на зовнішні пристрої, а також налаштування параметрів елементів апаратно-програмного комплексу систем після їх заміни, після ремонтів, при внесення змін у технологічний процес тощо.

Організація людино-машинного інтерфейсу технічного забезпечення у вигляді дворівневого блокнота, в якому на першій сторінці відображається меню, за

яким відбувається вибір модуля введення/виведення для перевірки зовнішніх ланцюгів підключення системи чи параметрів, що підлягають налаштуванню, а на другій, підлеглий, відображається у табличному вигляді інструментарій для такого налаштування, забезпечує прозору та зручну реалізацію такого інтерфейсу.

Людино-машинний інтерфейс технічного забезпечення систем керування технологічним процесом скорочує час на пошук порушень у зовнішніх ланцюгах підключення системи і налаштування її параметрів, що в кінцевому рахунку запобігає непродуктивним простоям технологічного процесу і спрощення вимог до технічного персоналу, що її обслуговує.

ПЕРЕЛІК ПОСИЛАНЬ

1. Пулена О.М. Розроблення людино-машинних інтерфейсів та систем збирання даних з використанням програмних засобів SCADA/HMI: Навч. посіб. Київ : Видавництво Ліра-К, 2020. — 594 с.
2. Bailey D., Wright E. Practical SCADA for Industry. - Newnes, 2005. — 304 p.
3. Автоматизована система диспетчерського керування WindEx. URL: https://activolt.com.ua/wp-content/uploads/2021/04/windex-.2_2021.pdf (дата звернення 24.04.2023).
4. Kuvaiev, V., Ishchuk, P., Politov, A., Buriak, V. (2021). Proqramni rishennia po zabezpechenniu nadiinoho funktsionuvannia skladnykh informatsiino-keruiuchykh system krytychnykh do rezhymu realnoho chasu [Software solutions to ensure the reliable operation of complex information and control systems critical to real-time]. Information Technology: Computer Science, Software Engineering and Cyber Security, 1, 16–24. [in Ukrainian]. doi: <https://doi.org/10.32782/IT/2021-1-3>

УДК 004.932:528.854

В.Ю. Каштан¹, В.В. Гнатушенко¹, О.В. Коробко¹, О.В. Гай¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ МОНІТОРИНГУ ВУЛКАНІЧНОЇ АКТИВНОСТІ В ЛА-ПАЛЬМІ

Анотація. Робота присвячена актуальній задачі розробки інформаційної технології моніторингу вулканічної активності на основі обробки та аналізу даних дистанційного зондування та геоінформаційних методів. Розроблено інформаційну технологію за допомогою якої було виявлено та здійснено аналіз впливу вулканічної активності в Ла-Пальмі на морське середовище та навколишню екосистему.

Ключові слова: інформаційна технологія, моніторинг вулканічної активності, Sentinel-2.

Вступ. Виверження вулканів – явище, що швидко розвивається, ускладнюючи моніторинг та прогноз [1]. Але за останні чотири десятиліття завдяки поширенню нових та інноваційних інструментів і методів вчені досягли прогресу в прогнозуванні вивержень вулканів. Нові дослідження показали, що

поєднання наземної інформації, супутникових даних і технологій штучного інтелекту [2] сприяє послідовному та довгостроковому моніторингу вулканів, дозволяючи дослідникам мати повний огляд вулканічних систем. Розуміння поведінки вулканів може допомогти відстежувати попереджувальні ознаки заворушень, прогнозувати виверження вулканів і можливі пов'язані з ними зсуви та цунамі, а отже, своєчасно активувати плани на випадок надзвичайних ситуацій.

Супутникові дані надають інформацію в реальному часі та є безпечнішим способом збору інформації, так як не потрібно наближатися до кратерів і вентиляційних отворів для збору даних. Такі методи дистанційного моніторингу можуть виявляти ледь помітні рухи гір, деформацію ґрунту, викиди газів, гідротермальну активність і підвищення температури. Через потік даних, зібраних наземними та космічними приладами, дослідники також використовують методи машинного навчання, форму штучного інтелекту, за допомогою якої комп'ютерні алгоритми, такі як нейронні мережі, можна навчити виявляти закономірності в даних.

Метою роботи є розробка інформаційної технології моніторингу вулканічної активності в Ла-Пальмі.

Постановка задачі. Для досягнення мети необхідно вирішити наступні завдання:

- виконати аналіз особливостей моніторингу вулканічної активності на основі даних дистанційного зондування та дослідити різні підходи до виявлення змін на космічних знімках;

- розробити методику аналізу даних отриманих з космічних знімків для оперативного моніторингу вулканічної активності;

- розробити комплексну методику дешифрування та виявлення просторово-часових змін, за космічними знімками.

Основний зміст роботи. В роботі запропоновано інформаційну технологію моніторингу вулканічної активності використовує різночасові знімки супутника Sentinel-2, аналізує теплові індекси та проводить аналіз параметрів води у водоймах. Спочатку завантажуються різночасові знімки Sentinel-2. Далі було виконано атмосферну корекцію, яка видаляє вплив атмосфери на супутникові знімки. Один з популярних методів - це метод DOS (Dark Object Subtraction), який видаляє вплив атмосферного розсіювання. Формула для DOS виглядає так [1]:

$$R_{atmospheric\ corrected} = R_{measured} - R_{dark\ object} \quad (1)$$

Третій крок, який включає розрахунок теплових індексів (*NDVI*, *VAI*, *VEI* та *VHI*), що допоможе виявити теплові аномалії на поверхні землі. Гарячі об'єкти (наприклад, лавові потоки) особливо випромінюють у спектральній області SWIR. Більше того, поверхні з вищою температурою випромінюють сильніше на коротших, а не довших довжинах хвиль SWIR. Отже, сенсори з каналами в інфрачервоному діапазоні можуть бути використані для ідентифікації та

картування вулканічних теплових аномалій. З іншого боку, вдень відбите сонячне випромінювання становить велику проблему. Тому, пропонуємо в роботі використати алгоритм NHI, який використовує два нормалізованих індекси для виявлення та картографування вулканічних гарячих точок. Для аналізу вулканічної активності проведено аналіз параметрів води у водоймах на основі побудови картх каламутності поверхневих шарів морської води, вміст хлору у воді та вміст сірчистого ангідриду. Розрахунок відображення водної поверхні (Remote Sensing Reflectance - Rrs) [3]:

$$Rrs = (Lw / Es), \quad (2)$$

де Lw - відображення водної поверхні (пропорційно рівню світла, яке покидає воду); Es - сумарна інсоляція на водній поверхні.

Завершальним етапом є візуалізація отриманих даних в географічній інформаційній системі QGIS.

Обговорення результатів. Аналіз даних показав, що значення VHI (рис.1) від +0.1 до +1 вказують на те, що рослинний покрив перебуває в нормальному, здоровому стані, без стресу. Значення VHI від 0 до +0.1 можуть вказувати на початковий стрес рослинного покриву, але стан рослин ще не є критичним. Значення VHI від -0.1 до 0 можуть свідчити про помітний стрес рослинного покриву та можливість виникнення проблем. Значення VHI від -0.1 до -1 вказують на серйозний стрес рослинного покриву і можливу шкоду для рослин.

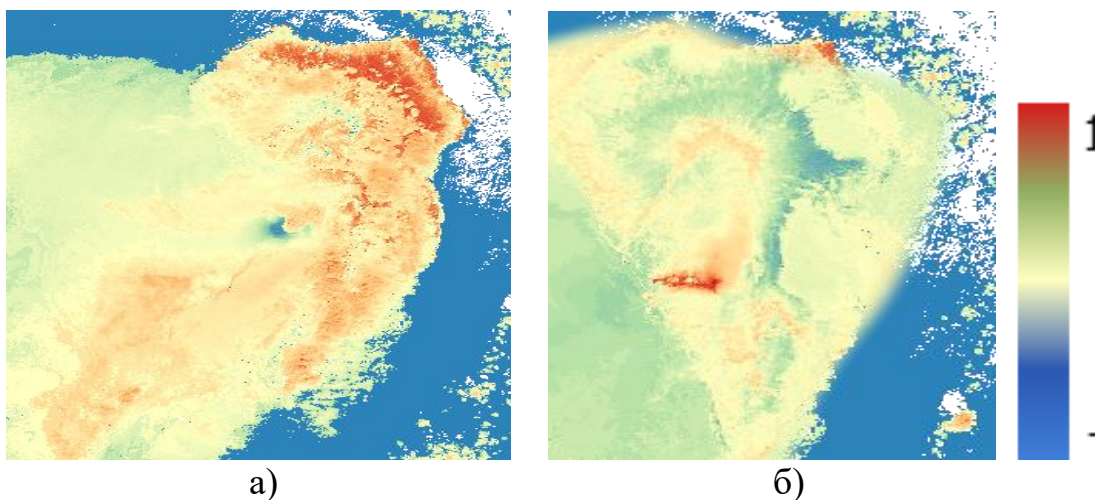


Рис. 1. Результати індексу VHI: а) 14.11.2021; б) 14.12.2021

Наукова новизна полягає у застосуванні інтегрованого методу моніторингу вулканічної активності, який поєднує в собі сучасні підходи обробки супутникових даних та аналізу теплових індексів з поглибленим вивченням впливу вулканічної активності на водні екосистеми та атмосферу.

Висновки. В роботі розроблено і описано інформаційну технологію моніторингу вулканічної активності в Ла-Пальмі з використанням оптичних супутникових даних Sentinel-2. Результати свідчать про видимі зміни в

каламутності поверхневої морської води, спричинені виверженням вулкана. Підвищений рівень каламутності спостерігався після утворення лавової дельти та поширення лавового шлейфу. Вулканічні шлейфи, містять в собі матеріали зі змінним хімічним складом, які можуть впливати на хімічний склад океану та водоростеві спільноти.

Такий аналіз важливий для зрозуміння впливу вулканічної активності на морське середовище та навколишню екосистему. Ці дані можуть бути корисні для дослідження впливу вулканічних вивержень на океанічній екосистемі і хімічний склад води.

ПЕРЕЛІК ПОСИЛАНЬ

1. Bedoya-Velásquez, A.E.; Hoyos-Restrepo, M.; Barreto, A.; García, R.D.; Romero-Campos, P.M.; García, O.; Ramos, R.; Roininen, R.; Toledano, C.; Sicard, M.; et al. Estimation of the Mass Concentration of Volcanic Ash Using Ceilometers: Study of Fresh and Transported Plumes from La Palma Volcano. *Remote Sens.* 2022, 14, 5680.
2. Crawford B. et al. Mapping pollution exposure and chemistry during an extreme air quality event (the 2018 Kīlauea eruption) using a low-cost sensor network // *Proceedings of the National Academy of Sciences.* – 2021. – Т. 118. – №. 27. – С. e2025540118.
3. Córdoba-Jabonero C. et al. Fresh volcanic aerosols injected in the atmosphere during the volcano eruptive activity at the cumbre vieja area (La Palma, Canary Islands): temporal evolution and vertical impact // *Atmospheric Environment.* – 2023. – Т. 300. – С. 119667.

УДК 330

Л.В. Кабак¹, Д.М. Мороз¹, В.О. Буток¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

СУЧАСНІ ТЕХНОЛОГІЇ УПРАВЛІННЯ ОСОБИСТИМИ ФІНАНСАМИ ЗА ДОПОМОГОЮ АНАЛІТИЧНИХ ІНСТРУМЕНТІВ

Анотація. Застосування SWOT-аналізу в контексті управління особистими фінансами дає людям можливість об'єктивно оцінити своє фінансове становище, визначити напрямки подальшого поліпшення, розробити стратегії для досягнення фінансової стабільності та досягнення особистих фінансових цілей.

Ключові слова: SWOT-аналізу фінансової стабільності управління особистими фінансами.

Вступ. Питання підвищення ефективності управління особистими фінансами зосереджують увагу багатьох науковців та фінансових консультантів. Від ефективності управління особистими фінансами залежить наша здатність досягати фінансових цілей, забезпечувати фінансову стабільність та визначати напрямки розвитку. Для вирішення проблем, які виникають у процесі фінансового планування, важливо не лише оптимізувати бюджетні процеси та застосовувати інноваційні підходи, але й удосконалювати підходи до управління

особистими фінансами та розширювати спектр управлінських методів, включаючи використання аналітичних інструментів.

Аналіз останніх досліджень і публікацій. Світовий досвід надає нам численні ефективні технології та інструменти, які успішно використовуються в багатьох країнах, але які досі не мають широкого поширення в Україні[1]. До причин, які гальмують впровадження нових технологій управління особистими фінансами, можна віднести таке:

1. Недостатня обізнаність особистих фінансових керівників щодо передових методів управління фінансами, включаючи використання аналітичних інструментів. Цей недолік можна виправити через самонавчання та участь у фінансових тренінгах та курсах.

2. Багато з управлінських інновацій вимагають певних витрат, включаючи програмне забезпечення та фінансові інструменти. Проте відсутність таких витрат може призвести до втрати можливостей для оптимізації та збільшення ефективності управління фінансами.

3. У складних економічних умовах багато людей фокусуються на поточних фінансових завданнях, таких як пошук джерел фінансування, оптимізація відносин з бізнес-партнерами та державними органами. Але використання сучасних методів управління особистими фінансами потребує стратегічного підходу[2].

Виклад основного матеріалу. Незважаючи на існуючі виклики, поширення успішного досвіду управління особистими фінансами в Україні є корисним і необхідним. Використання дієвих методів управління фінансами вважається інтенсивним способом досягнення фінансової стабільності та досягнення особистих фінансових цілей, включаючи використання аналітичних інструментів. Серед таких методів особливе місце посідають:

- Особистий фінансовий контролінг інтерпретується, як система інформаційного забезпечення координації всіх аспектів управління особистими фінансами, включаючи використання аналітичних інструментів. Реалізація функцій управління особистими фінансами здійснюється за допомогою стратегічного процесу, такого як створення, виконання та управління бюджетами для досягнення фінансових цілей. Бюджет визначає доходи і витрати на певний період і допомагає управляти ресурсами з урахуванням фінансових можливостей.

- Критерії: в області особистих фінансів використовуються для порівняння фінансових показників людини з показниками інших людей, груп або середніми показниками по країні. Цей метод дозволяє оцінити, наскільки ефективно вони фінансуються, порівнюючи їх з рівнем групи.

- Фінансовий аналіз: включає збір, обробку та інтерпретацію фінансової інформації для оцінки фінансового стану та прибутковості. Цей метод включає аналіз доходів, витрат, заощаджень та інвестицій для прийняття обґрунтованих фінансових рішень.

- Інвестування: означає вкладення грошей у різні фінансові інструменти для отримання прибутку. Інвестиційний аналіз включає вибір та оцінку інвестиційних можливостей, розуміння ризиків та прибутковості та визначення власного інвестиційного горизонту.

Ще одним інструментом стратегічного управління є матриця якісного стратегічного аналізу, або матриця SWOT-аналізу[3], спочатку розроблений для бізнес-середовища, аналізує і планує особисту фінансову ситуацію. аналіз базується на плюсах і мінусах у контексті управління особистими фінансами. виявлення можливостей і загроз.

Таблиця 1

SWOT-аналіз фінансового стратегування.

Сильні сторони:	Слабкі сторони:
Гарна фінансова дисципліна	Високі кредитні борги
Здатність до збереження та інвестування грошей	Низький рівень доходу
Спроможність ефективно бюджетувати	Обмежений доступ до фінансових можливостей
Вищий рівень доходу.	Нецілеспрямовані витрати
Можливості:	Загрози:
Зростання ринку праці та можливості підвищення доходів	Економічна нестабільність
Зниження процентних ставок на кредити	Збільшення інфляції
Розширення можливостей інвестування	Зростання ставок за кредитами
Посилення регуляції фінансового сектору	Втрата робочого місця або скорочення доходу

- Сильні сторони включають фінансові переваги, такі як стабільні джерела доходу, ефективний контроль витрат та розумні інвестиційні рішення. Визначивши ці сильні сторони, ви можете зміцнити їх переваги та підвищити фінансову стабільність.

- Слабкі сторони включають сфери, де фінансове становище може бути вразливим, такі як високі витрати порівняно з доходом та недостатня фінансова грамотність. Виявляючи слабкі місця, ви можете розробити стратегії їх покращення.

- Можливості включають перспективи фінансового зростання та розвитку, такі як нові інвестиційні можливості, збільшення доходів та розширення фінансових портфелів.

- Загрози включають фактори, які можуть негативно вплинути на фінансове становище, такі як економічні кризи, зміни на фінансових ринках та непередбачені обставини.

Висновок: Застосування SWOT-аналізу в контексті управління особистими фінансами дає людям можливість об'єктивно оцінити своє фінансове становище, визначити напрямки подальшого поліпшення, розробити стратегії для досягнення фінансової стабільності та досягнення особистих фінансових цілей. З віковою глобалізацією та швидкими змінами у світовій економіці ефективне управління особистими фінансами стало стратегічною необхідністю. Фінансова стабільність та оптимізація доходів можуть бути досягнуті шляхом аналізу фінансових даних, оцінки ризиків та можливостей, а також за допомогою новітніх стратегій та методів. Аналіз і адаптація глобального досвіду є важливим елементом успішного управління, а використання передових аналітичних інструментів відкриває нові можливості для стратегічного розвитку особистих фінансів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Куцяк В. О. Особисті фінанси в умовах розвитку ринкових відносин в Україні : дис. ... канд. екон. наук [08.00.08 «Гроші, фінанси і кредит»] / В. О. Куцяк. – ТНЕУ, 2016. – 220 с.
2. Довгань Л.Є. Стратегічне управління : навч. посібн. / Л.Є. Довгань, Ю.В. Каракай, Л.П. Артеменко. – К. : Центр навч. літ-ри, 2009. – 440 с
3. Коновалова О.В. SWOT-аналіз як основний інструмент стратегічного управління, його переваги і недоліки / О.В. Коновалова, Т. Андрущакевич. [Електронний ресурс]. – Доступний з http://www.rusnauka.com/3_SND_2010/Economics/58123.doc.htm

УДК 004.413.2

В.В. Спірінцев¹, Д.Р. Басараб¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

КЕРУВАННЯ ФУНКЦІЯМИ ВІДЕОДЗВІНКА ЗА ДОПОМОГОЮ РОЗПІЗНАВАННЯ ЖЕСТІВ В РЕЖИМІ РЕАЛЬНОГО ЧАСУ

Анотація. Описано процес поліпшення користувальницького досвіду при здійсненні відеодзвінка в програмі для організації відеоконференцій Zoom, за рахунок розпізнавання певного набору жестів (вмикання та вимикання камери або мікрофону, перемикання слайдів, початок і завершення запису відеоконференцій) в режимі реального часу для організації віддаленого керування функціями відеодзвінка.

Ключові слова: розпізнавання жестів, відеодзвінок, віддалене керування, нейронні мережі, машинне навчання, Zoom, Python, Tensorflow, Cvzone, OpenCv.

Вступ. Відеодзвінки є найпопулярнішим способом комунікації на сьогоднішній день. Через карантинні обмеження та воєнні дії процес навчання учнів та студентів цілковито перейшов в онлайн-формат відеоконференцій, де одним із лідерів є платформа Zoom. Одночасно з тим стрімко розвиваються й технології розпізнавання об'єктів на фото та відео за допомогою нейронних мереж. Поєднавши ці дві перспективні галузі можна створити систему, де управління функціями відеодзвінка можна буде здійснювати за допомогою жестів рук. Така технологія вже була анонсована самою платформою Zoom, що свідчить про її актуальність для користувачів, проте анонсовані функції від розробників обмежуються лише “підняття руки” та додавання реакції “палець догори”. Ознайомившись з основним функціоналом програми Zoom, можна зробити висновок, що крім згаданих функцій було б добре мати можливість гортання слайдів презентацій, виставлення початку та кінця запису відеодзвінка, запуск демонстрації екрану та інші корисні функції за допомогою жестів рук в режимі реального часу.

Постановка задачі. Для досягнення поставленої мети в роботі необхідно вирішити наступні завдання:

- здійснити огляд існуючих методів розпізнавання жестів рук за допомогою нейронних мереж;
- визначити перелік основних функцій програми Zoom, які будуть інтегровані із системою розпізнавання жестів в режимі реального часу;
- розробити алгоритм взаємодії системи розпізнавання жестів на базі нейронних мереж та застосунку Zoom та зробити висновок щодо доцільності створення даної системи.

Основний зміст роботи. Під час роботи над проектом були використані наступні методи та інструменти:

- методи розпізнавання жестів рук людини;
- Google Teachable Machine, Python з модулями CVzone та OpenCv, Zoom.

Розпізнавання жестів – це технологія, яка дозволяє комп'ютерам розуміти рухи рук людини та реагувати на них. Вона використовується в широкому діапазоні додатків, включаючи відеоігри, інтерфейси користувача та в режимі доступності для людей з обмеженими можливостями. Розпізнавання жестів

зазвичай працює за допомогою камери, яка відстежує рухи рук, дані камери обробляються алгоритмом розпізнавання жестів, що визначає, який саме жест було виконано людиною.

Існує кілька різних методів розпізнавання жестів рук [1-4]. Одним з найпоширеніших методів є визначення контурів. Цей метод використовує камеру для створення контуру руки. Далі по контуру визначають положення пальців, кисті та інших частин руки. Іншим поширеним методом є розпізнавання скелета руки. Цей метод використовує зображення камери для створення скелетної структури, що складається з точок, які представляють положення пальців, а ці точки в свою чергу з'єднані між собою лініями. Розпізнавання жестів рук має ряд переваг. В першу чергу, це є більш природний спосіб взаємодії з комп'ютером, ніж використання клавіатури чи миші. Крім того, розпізнавання жестів може бути корисним для людей з обмеженими можливостями, які не можуть легко користуватися комп'ютером.

Для відстеження жестів рук в даній роботі було використано бібліотеку CVzone мови програмування Python [5]. Ця бібліотека має ряд переваг перед конкурентами, а саме:

- порівняно висока швидкість виявлення контурів руки;
- точність відстеження руки та пальці навіть у складних умовах, таких як слабе освітлення або коли рука частково закрита;
- простота використання та легка інтеграція в існуючі програми;
- CVzone пропонує різноманітні функції, які роблять його корисним для відстеження рук або обличчя;
- бібліотека з відкритим вихідним кодом, тобто, ви можете використовувати її безкоштовно та змінювати її код відповідно до ваших потреб.

Тренування нейронної мережі відбувається на базі Google Teachable Machine [6], веб-інструменту, який дозволяє легко створювати моделі машинного навчання без необхідності кодування. Процес можна поділити на наступні етапи:

- збір тренувальних даних;
- визначення класів для розпізнавання, шляхом призначення міток своїм даним тренувань;
- навчання моделі для визначати шаблонів у даних і використання цих шаблонів для класифікації нових даних;
- експортування моделі у TensorFlow.

Після того, як систему розпізнавання інтегровано в проект за допомогою TensorFlow можна перейти до написання умов виклику тієї чи іншої функції відеодзвінка в Zoom. В рамках роботи було розроблено перелік жестів, що відповідають за перемикавання слайдів презентації, увімкнення і вимкнення камери та мікрофону користувача, початок та кінець демонстрації екрану, локального та хмарного запису конференції, а також виведення на екран віджету з усіма учасниками конференції. В ході тестування системи було виявлено деякі аспекти, що можуть бути додані для поліпшення користувальницького досвіду

та оптимізації алгоритму програми. В першу чергу це додавання жестів, що означають початок та завершення віддаленого керування, з метою уникнення випадкових дій у разі активної жестикуляції користувача під час розмови. Також для покращення точності розпізнавання було вирішено не виділяти кожен жест окремо, а заповнювати масив на 20 елементів, котрий буде поповнюватись новим передбаченням кожну мілісекунду для того, аби вибрати елемент, що повторюється в масиві найчастіше. Цей елемент і буде передано для подальшого керування в якості розпізнаного жесту з покращенням точності передбачення. Список передбачень очищується одразу після видачі значення покращення точності.

Інтерфейс програми являє собою зображення із вбудованої вебкамери, що транслюється в режимі реального часу, із доданими графічними елементами, а саме:

- текстовий рядок, що показує чи увімкнений режим контролю для відеодзвінку;
- текстовий рядок, що показує останній жест, що було розпізнано;
- рамка навколо руки користувача (якщо рука у кадрі) із текстовим рядком, що показує передбачення нейромережі;
- список із 10 зображень, на яких наведено жести, що треба показати для виконання тієї чи іншої дії.

Початковий інтерфейс, що користувач побачить одразу після запуску програми наведено на рис. 1.

Як зазначалось раніше, для активації режиму віддаленого керування користувачу необхідно показати відповідний жест, а саме “палець догори”. Приклад інтерфейсу програми одразу після зчитування даного жесту наведено на рис. 2. Можна побачити, що у верхній частині вікна текст буде змінено із “Gesture control is OFF” на “Gesture control is ON”, а також текстове поле, що містить відомості про останній успішно зчитаний жест оновить значення на “start control”.

На рис. 3 наведено приклад жесту, що активує певну функцію у відеоконференції. Після зчитування саме такого жесту та за умови, що режим контролю увімкнено, мікрофон користувача буде вимкнено у програмі Zoom. Повторне зчитування такого жесту зробить обернену дію, тобто увімкне мікрофон.



Рис. 1. Початковий інтерфейс програми одразу після запуску

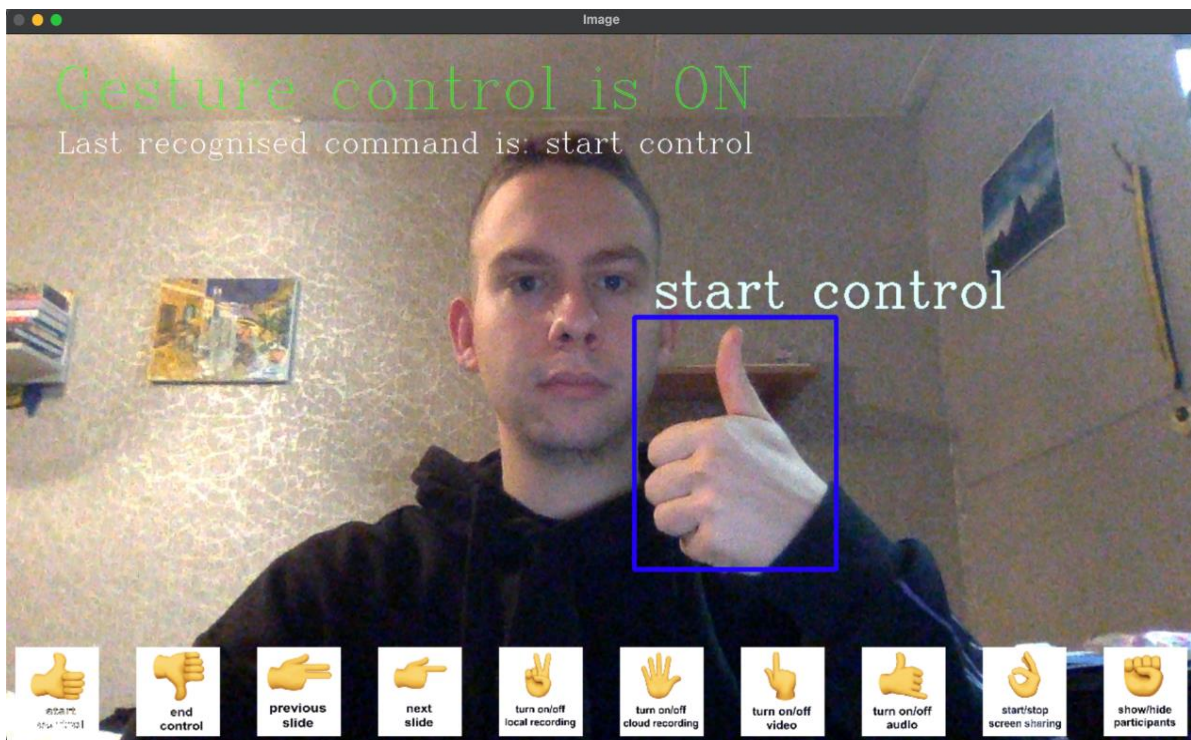


Рис. 2. Інтерфейс програми у разі показу користувачем жесту, що відповідає за дозвіл віддаленим керуванням



Рис. 3. Інтерфейс програми у разі зчитування жесту, що відповідає за увімкнення/вимкнення мікрофону

Наукова новизна розробки полягає у запропонованому в роботі підході по організації взаємодії системи розпізнавання жестів на базі нейронних мереж та застосунку для відеоконференцій Zoom для підвищення ефективності користувальницького досвіду учасників відеоконференцій та віддаленого керування функціями відеодзвінка в режимі реального часу.

Висновки. В результаті було спроектовано програмне забезпечення, котре зчитує сигнал із показаного жесту руки людини в режимі реального часу та передає його у вигляді комбінації клавіш до операційної системи комп'ютера, що в свою чергу надає змогу використовувати функції застосунку Zoom без використання клавіатури, тобто дистанційно.

ПЕРЕЛІК ПОСИЛАНЬ

1. Posture and Gesture Recognition for Human-Computer Interaction. [Electronic resource]. – Access mode: <https://www.intechopen.com/chapters/8712>– Title from the screen.
2. Pavlovic V. I., Sharma R., Huang T. S. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. IEEE Transactions on Patter Analysis and Machine Intelligence. 1997. Vol. 19, №7. P. 683-695
3. Viblis M. K., Kyriakopoulos K. J. Gesture Recognition: The Gesture Segmentation Problem. Journal of Intelligent and Robotic Systems. 2000. Vol. 28, №2. P. 367-386
4. Zimmerman T. G., Lanier J., Blanchard et al C. A hand gesture interface device: project report (final). VPL Research, Inc. Redwood CityCA, 1987. 4 p.
5. CVzone. [Electronic resource]. – Access mode: <https://github.com/cvzone/cvzone> – Title from the screen.
6. Teachable Machine. [Electronic resource]. – Access mode: <https://teachablemachine.withgoogle.com/> – Title from the screen.

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДОСЛІДЖЕННЯ ПРОСТОРОВО-ЧАСОВИХ ЗМІН БЕРЕГОВОЇ ЛІНІЇ ТУЗЛОВСЬКИХ ЛИМАНІВ

Анотація. В роботі запропоновано інформаційну технологію, що дозволяє вивчати зміни берегової лінії на основі різночасових оптичних космічних знімків. Використовуючи методи сегментації та математичної морфології, розроблена технологія дозволяє провести аналіз часового та просторового розвитку берегової лінії. Отримані результати свідчать про ефективність запропонованого методу у картографуванні змін берегової лінії в умовах численних природних впливів.

Ключові слова: *берегова лінія, інформаційна технологія, просторово-часові зміни, оптичні знімки.*

Вступ. Просторово-часові зміни берегової лінії вказують на зміни в розташуванні прибережної зони протягом певного періоду часу. Це може включати зсуви берега, зміни розміру пляжів, ерозію узбережжя та інші процеси, які впливають на форму та структуру берегової лінії.

Останніми роками через зміни природних факторів, таких як глобальне потепління та часті антропогенні дії на берегових лініях, зміни берегових ліній прискорилися [1]. Збір і аналіз довгострокових послідовних даних про берегові лінії мають вирішальне значення для розуміння впливу природної та антропогенної діяльності на зміни узбережжя та для надання рекомендацій щодо захисту та сприяння сталому розвитку прибережних регіонів.

Отримання місцезнаходження берегової лінії є першим кроком у вивченні просторово-часових змін. Розвиток технології дистанційного зондування продемонстрував, що берегові лінії можна отримати із зображень дистанційного зондування, а оптичні супутники, такі як Landsat, можна використовувати для картографування діапазонів повеней. Зображення космічного радіолокатора з синтетичною апертурою (SAR) не обмежується темрявою або погодними умовами, а мультиполяризаційний SAR також надає інформацію про часовий ряд для класифікації наземних об'єктів [2].

Постановка задачі. Розробити інформаційну технологію дослідження просторово-часових змін берегової лінії на різночасових оптичних космічних знімках.

Для досягнення цієї мети необхідно вирішити наступні завдання:

- виконати аналіз особливостей моніторингу берегової лінії на основі оптичних космічних знімків;
- розробити метод сегментації для ефективного виділення берегової лінії на різночасових оптичних космічних знімках;

- покращити точність визначення контурів берегової лінії та усунути артефакти, використовуючи математичну морфологію;
- розробити інформаційну техноогію аналізу просторово-часової динаміки берегової лінії на основі зібраних даних з різних часових точок;
- провести валідацію розробленої технології, порівнявши отримані результати з відомими даними про берегову лінію.

Основний зміст роботи. Запропонована в роботі інформаційна технологія аналізу просторово-часових змін берегової лінії є комплексним процесом, що включає кілька етапів обробки та аналізу супутникових даних та топографічних карт.

Етап перший завантаження знімків за 2016. На початку вибираються та завантажуються супутникові знімки за 2016 рік з оптичного супутника Sentinel-2.

Другим етапом є попередня обробка даних та сегментація даних. Попередня обробка даних включає в себе калібрування супутникових знімків, видалення атмосферного шуму і покращення контрастності. На цьому етапі метод кластеризації в поєднанні з Orfeo ToolBox (OTB) в QGIS використовується для сегментації супутникових зображень і групування пікселів за їхніми спектральними характеристиками. Цей метод допомагає виділити об'єкти та регіони із зображень, у тому числі і берегову лінію.

Третій етап є важливим так як відповідає за картографування берегової лінії, базуються на морфологічних операціях та функціях. На даному етапі застосовується алгоритм ерозії з метою зменшення розміру об'єктів на зображенні та ефективного видалення шуму [3]:

$$Erosion(I, B) = \min(I(x, y) \text{ for } (x, y) \text{ in } B), \quad (1)$$

де $I(x, y)$ - інтенсивність пікселя (зазвичай від 0 до 255) на позиції (x, y) на зображенні, B - ядро ерозії.

Наступним етапом є застосування морфологічного оператора дилатації з метою уточнення контуру берегової лінії. Цей підхід сприяє підвищенню видимості та чіткості контурів берегової лінії, що є важливим для подальшого точного аналізу та вимірювань [3].

Завершальним етапом є програмна реалізація та використання функції `cv2.findContours` для побудови точних контурів берегової лінії. `cv2.findContours` - це функція з бібліотеки OpenCV, яка дозволяє знаходити контури об'єктів на зображенні. Вона працює з піксельними значеннями, що були підвищені під час дилатації. Н цьому етапі отримуємо дві бінарні маски, одна для 2016 року, інша – 2023 року, де чорними пікселями відзначені області берегової лінії (рис.1).

Для аналізу цих змін використовується коефіцієнт Пірсона (r), який служить для визначення зв'язку між двома величинами [3, 4]:

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}, \quad (2)$$

де \bar{x} , \bar{y} є середніми значеннями двох змінних x і y відповідно.

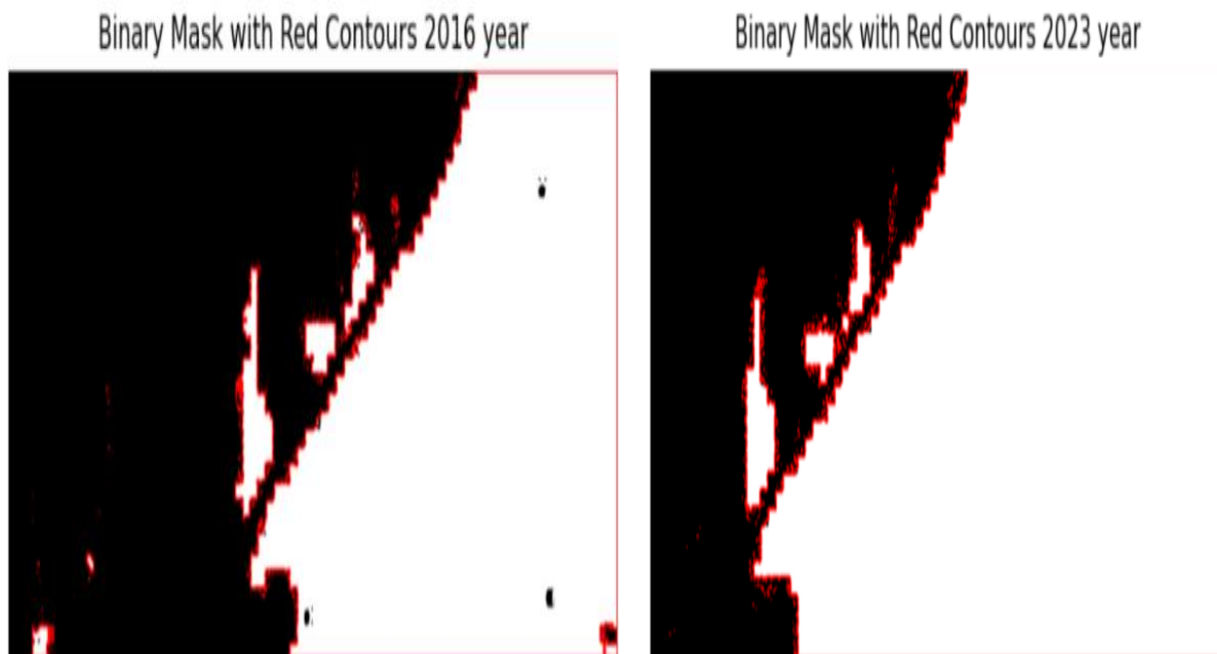


Рис. 1. Бінарна маска супутникового знімка за 2016 рік та 2023 рік

Експериментальні дослідження були проведені на території Національного природного парку "Тузлівські лимани", що розташований у Татарбунарському районі Одеської області. Цей парк включає в себе систему лиманів, таких як Шагани, Алібей та Бурнас, а також групу солоних лиманів лагунного типу. Від узбережжя Чорного моря ці лимани відокремлені піщаною косою завдовжки 29 кілометрів і шириною від 60 до 400 метрів. Заснований у 2010 році, Національний природний парк має за мету захист унікальної і вразливої природи Причорномор'я.

На рис. 2 представлено результат роботи розробленої інформаційної технології. Візуальний аналіз дозволив виявити зміни у береговій лінії, ідентифікувати території, де відбувається осушення лиманів. Це вказує на визначення областей, які протягом періоду з 2016 по 2023 роки, згідно з аналізом земельного покриття на супутникових знімках, втратили свій водний характер. Деякими з таких областей є, наприклад, Солоне озеро-лиман та озеро-лиман Хаджидер

Наукова новизна полягає у розробці технології виявлення сталих у часі просторово-часових змін берегової лінії, що ґрунтується на аналізі різночасових даних супутникових зйомок, розрахунку коефіцієнта кінцевої точки та коефіцієнта Пірсона. Це дозволило автоматично картографувати зміни на карті.



Легенда

— берегова лінія 2023 року

— берегова лінія 2016 року

Рис.2. Результати досліджень берегової лінії Тузловських лиманів

Висновки. В даній роботі було представлено інформаційну технологію дослідження просторово-часових змін берегової лінії Тузловських лиманів. Аналіз результатів показав, що на супутниковому знімку за 2023 рік берегова лінія розташована ближче до водних об'єктів порівняно з знімком за 2016 рік, що свідчить про втрату прибережних територій (обміління) протягом цього періоду..

ПЕРЕЛІК ПОСИЛАНЬ

1. Довгий С.О., Лялько В.І., Бабійчук С.М., Кучма Т.Л., Томченко О. В. Основи дистанційного зондування Землі: історія та практичне застосування: навч. посіб. / С. О. Довгий, В. І. Лялько, С. М. Бабійчук, Т. Л. Кучма, О. В. Томченко. – Київ: Інститут обдарованої дитини НАПН України, 2019. – 316 с.
2. Ai B., Zhang R., Zhang H., Ma C., Gu F. (2019). Dynamic process and artificial mechanism of coastline change in the Pearl River Estuary. *Regional Stud. Mar. Sci.* 30, 100715. doi: 10.1016/j.rsma.2019.100715.
3. Boussetta A., Niculescu S., Bengoufa S., Zagarni M. F. (2022). Spatio-temporal analysis of shoreline changes and erosion risk assessment along Jerba island (Tunisia) based on remote-sensing data and geospatial tools. *Regional Stud. Mar. Sci.* 55, 102564. doi: 10.1016/j.rsma.2022.102564
4. Wang X., Yan F., Su F. (2021). Changes in coastline and coastal reclamation in the three most developed areas of China, 1980–2018. *Ocean Coast. Manage.* 204, 105542. doi: 10.1016/j.ocecoaman.2021.105542.

ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГНОЗУВАННЯ ДЛЯ АНАЛІЗУ РОЗВИТКУ ВЕТЕРИНАРНИХ ПОСЛУГ

Анотація. Розглянуто застосування методів короткострокового прогнозування для аналізу розвитку ветеринарних послуг в Україні.

Ключові слова: системний аналіз, короткострокове прогнозування, ветеринарні послуги.

Вступ. Ветеринарна медицина – комплекс заходів направлений на профілактику захворювань тварин, лікування тварин, підвищення якості продукції тваринного походження, попередження та боротьби з хворобами, спільними для людей і тварин, одержання екологічно чистих продуктів харчування. У теперішній час розвиток приватного бізнесу у сфері ветеринарній медицині відіграє важливу роль у наданні послуг. Незважаючи на конкурентні умови, ринок ветеринарних послуг стабільно зростає, проте ця сфера залишається менш дослідженою порівняно з іншими галузями. Тому прогнозування та дослідження розвитку цієї сфери послуг є актуальним.

Метою даної роботи є дослідження застосування методів прогнозування для аналізу динаміки розвитку ветеринарних послуг на території України та отримання прогнозу на короткий період.

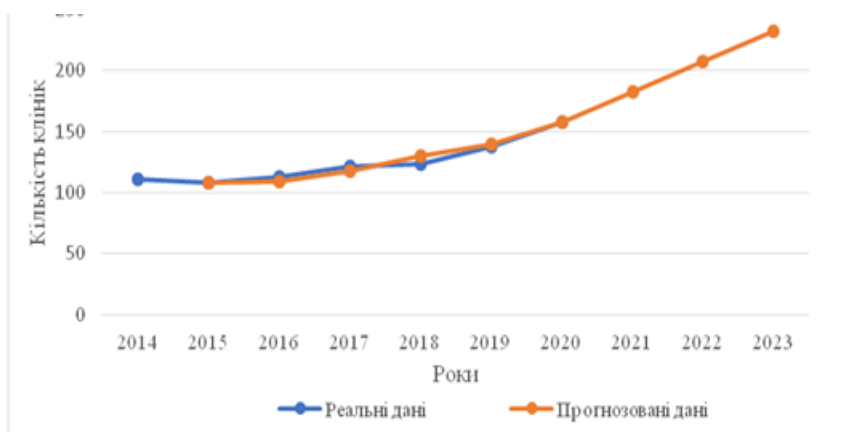
Основний зміст роботи. Вихідними даними є інформація про кількість ветеринарних закладів у різних областях України в період з 2014 по 2020 рік [1].

Для дослідження було обрано Дніпропетровську та Вінницьку області, оскільки вони мають різну динаміку розвитку. У дніпропетровській області в цей період спостерігалось стійке зростання кількості ветеринарних клінік, а у Вінницькій, починаючи із 2015 року – різкий спад. Було розглянуто три методи прогнозування, а саме: модель Хольта – Вінтерса, метод ковзного середнього та прогнозування за моделлю Брауна. Кожен із розглянутих методів має свої особливості і сферу застосування.

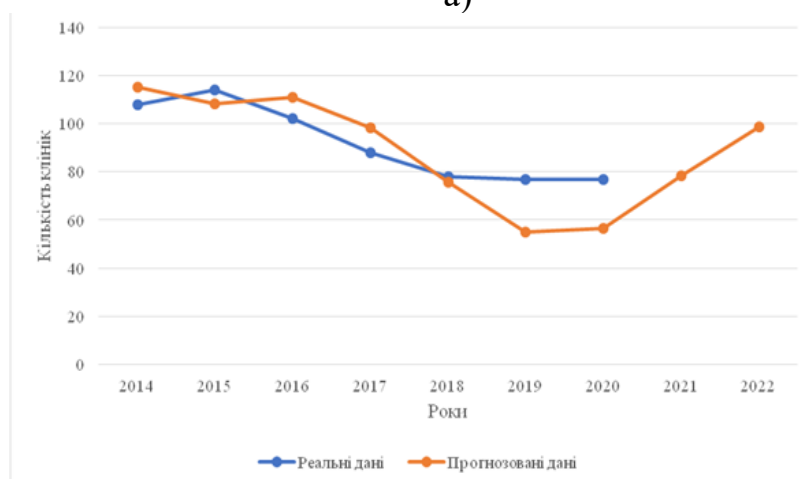
Модель Хольта–Вінтерса – це метод прогнозування часових рядів з використанням експоненційного згладжування [2]. Згладжування полягає у створенні зваженого середнього, вага якого обирається за принципом: чим старіша інформація для прогнозу, тим менший її вплив на поточний прогноз. Результати застосування показали, що він є найбільш прийнятним для даних по Дніпропетровській області, тут похибка прогнозу становила 1,6 %. Проте ця модель виявилася непринятною для даних Вінницької області – похибка прогнозу склала 16 %. (рис. 1).

Метод ковзного середнього є одним з широко відомих методів згладжування часових рядів. Згладжування за допомогою ковзного середнього

засноване на тому, що в середніх величинах взаємно погашаються випадкові відхилення. Це відбувається внаслідок заміни первинних рівнів часового ряду середньою арифметичною величиною всередині обраного інтервалу часу. Отримане значення відноситься до середини обраного інтервалу часу (періоду). В роботі було розглянуто інтервали у 2, 3, 4 роки. Цей метод був найкращим для прогнозування за даними Вінницької області (рис. 2). Найбільш прийнятним виявився період згладжування 2 роки.



а)



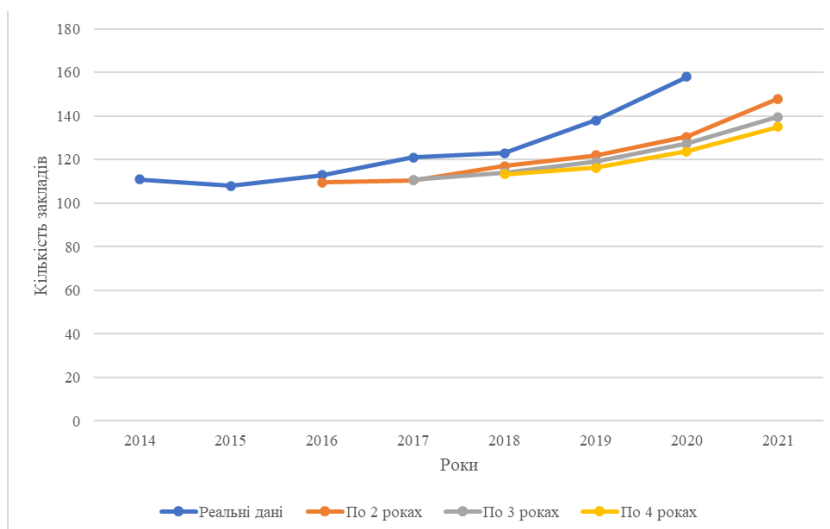
б)

Рис. 1. Вихідні дані та результати прогнозування за моделлю Хольта-Вінтерса: а) у Дніпропетровській області; б) у Вінницькій області

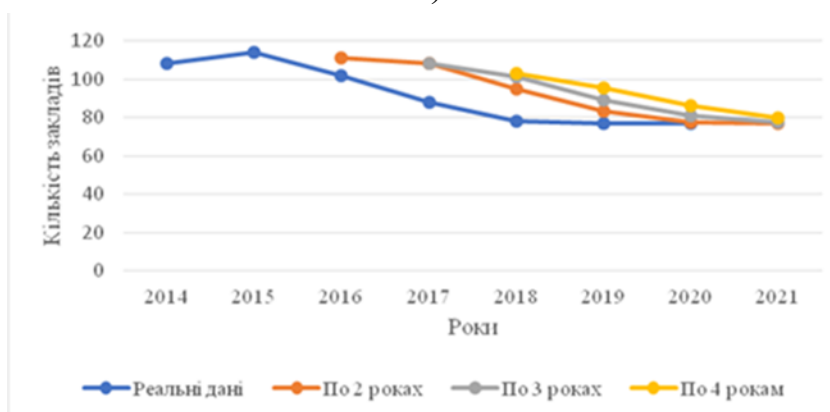
Прогнозування за моделлю Брауна дало велику помилку для обох областей.

Під час проведення дослідження прогнозування було виконано на 3 роки вперед. Зауважимо, що чим більша кількість років прогнозування, тим меншим його точність.

Наукова новизна розробки полягає у дослідженні і виборі оптимального методу прогнозування розвитку сфери ветеринарної медицини за регіонами України.



а)



б)

Рис. 2. Вихідні дані та прогноз за методом ковзного середнього:
а) у Дніпропетровській області; б) у Вінницькій області

Висновки. В роботі було досліджено застосування різних методів прогнозування для аналізу динаміку розвитку ветеринарної медицини за регіонами України. Аналіз показав, що при здійсненні прогнозування, доцільно використання кількох методів із метою обрання найбільш відповідного вихідному набору даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. Державна служба статистики України (ukrstat.gov.ua)
2. Бідюк П. І. Аналіз часових рядів: навчальний посібник // П. І. Бідюк, В. Д. Романенко, О. Л. Тимошук / М-во освіти і науки України, – Київ: Політехніка, – 2010. – 317с
3. Хом'як Т. В. Застосування методів згладжування для прогнозування обсягу виробництва/ Т. В. Хом'як, А. В. Малієнко, Г. В. Симоніць // Вісник Національного технічного університету «ХПІ». Збірник наукових праць, Серія: Системний аналіз, управління та інформаційні технології. – Харків: НТУ «ХПІ». – 2019. – № 1. – С. 8 – 12

ОСОБЛИВОСТІ СУЧАСНИХ БАЗ ДАНИХ ТА МОЖЛИВОСТІ ЇХ ВИКОРИСТАННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ РІЗНОГО ТИПУ

Анотація. в цій доповіді розглянуто різницю між двома основними типами баз даних, що використовуються в багатьох проектах. Також розглянуто переваги і недоліки цих типів.

Ключові слова: нереляційні бази даних, бази даних, реляційні бази даних, NoSQL, інформаційні системи.

Вступ. Дуже важко представити сучасний світ без баз даних, вони використовуються майже всюди: ІТ-проекти, державні установи, звичайні магазини і т.д. Однією з ключових характеристик бази даних є її тип, що може бути реляційним та нереляційним. В цій доповіді буде розглянуто відмінності між цими двома типами БД, засновуючись на основних характеристиках.

Актуальність проблеми. В реляційних базах даних дані зберігаються у вигляді таблиць, що складаються з стовпців та рядків. Зв'язки між таблицями розробляються заздалегідь. Будь-який рядок в таблиці можна відзначити унікальним ідентифікатором (ID), такий рядок буде називатися первинним ключем. Різні таблиці можуть бути зв'язані одна з одною за допомогою зовнішніх ключів. На відміну від реляційних баз даних, в нереляційних (або NoSQL) не використовується традиційна таблична схема рядків та стовпців. Модель зберігання даних визначається та оптимізується за потребами окремого проєкту. Наприклад, дані можуть зберігатися у форматі XML, JSON, як пара «ключ-значення» або навіть у вигляді графу. Нижче наведено графічні приклади структур БД (рис. 1-3).

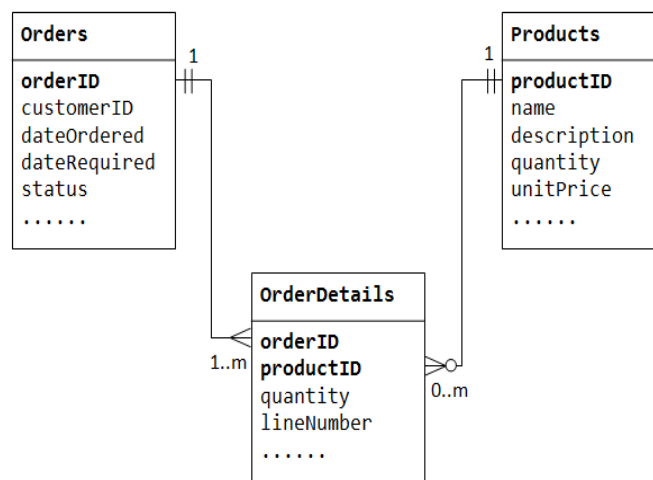


Рис. 1. Приклад структури реляційної БД [3]

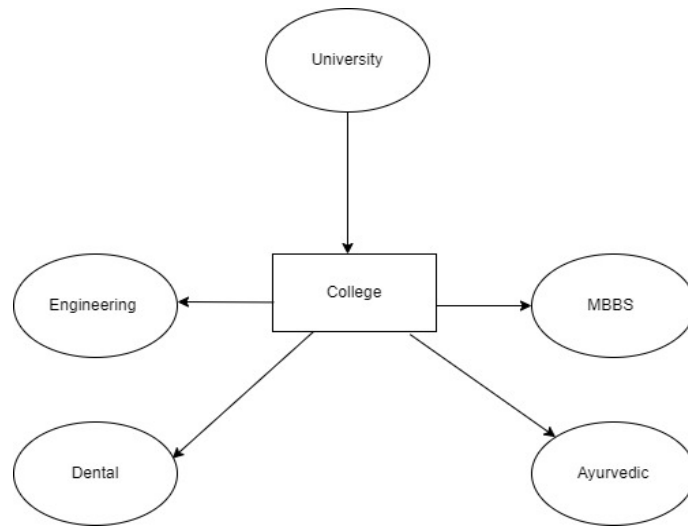


Рис. 2. Приклад структури нереляційної БД у вигляді графу [4]

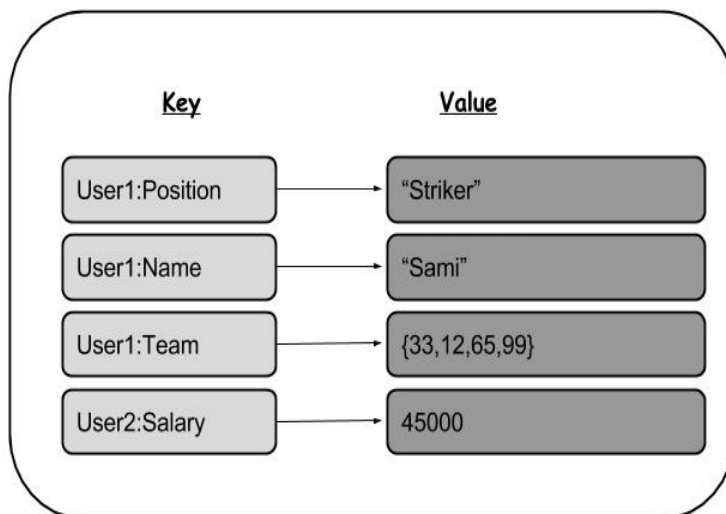


Рис. 3. Приклад структури нереляційної БД у вигляді ключ-значення [5]

Виклад основного матеріалу. Швидкодія цих різних типів БД майже однакова та може відрізнятись тільки в окремих випадках. Наприклад, під час роботи з великими об'ємами даних та обробкою у реальному часі краще показуються себе нереляційні бази даних [6], тому що вони, як правило, призначені для доступу до них за допомогою конкретних шаблонів.

В реляційних БД нормалізація є важливим аспектом під час розробки, для того щоб уникнути аномалії. Це відбувається шляхом розбиття великих таблиць на менші, добре структуровані таблиці. В NoSQL нормалізація зустрічається набагато рідше, оскільки вона вимагає строго визначеної структури таблиць.

Масштабування. Кожен проєкт, що розвивається, рано чи пізно стикається з проблемою масштабування через те, що даних в БД стає занадто багато і сервера не справляються із навантаженням, в наслідок чого все починає

«гальмувати». Реляційні бази даних масштабуються вертикально, тобто зростає навантаження на один сервер, в той час як в NoSQL навантаження розподіляється між різними серверами. Тобто, якщо в традиційних базах даних потрібен один потужний сервер, то в нереляційних БД можна використовувати декілька не дуже потужних серверів.

Переваги реляційних баз даних: Стабільна структура та вищий рівень нормалізації

- Гарантована наявність ACID властивостей (атомарність, консистентність, ізолюваність, довіреність)
- Цілісність та безпека даних
- Можливість використання складних SQL запитів
- Наявність зовнішніх ключів, що сприяє інтегрованості даних та уникненню дублювання інформації

Недоліки реляційних баз даних:

- Неefективність, якщо дані неструктуровані. Наприклад, якщо зберігається багато текстових документів чи зображень.
- Вертикальне масштабування
- Строга структурованість таблиць може призвести до ускладнення внесення змін до БД

Переваги нереляційних (NoSQL) баз даних:

- Ефективна робота з неструктурованими даними
- Горизонтальне масштабування
- Висока швидкодія при роботі з великими обсягами даних
- Варіативність структури БД

Недоліки NoSQL:

- Нереляційні бази даних можуть бути менш ефективними у виконанні складних аналітичних запитів
- Необхідно уважно проектувати БД. Через те, що схема гнучка, треба уважно продумувати структуру, щоб уникнути неоднозначностей
- Менше розширених можливостей для виконання запитів та аналізу даних

Бази даних які використовуються у проектах вибираються в залежності від типів інформаційних систем. Інформаційна система (ІС) – сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів. [8] Розглянемо основні існуючі типи інформаційних систем.

Типи інформаційних систем:

1. Системи планування ресурсів підприємства (англ. Enterprise Resource Planning або ERP). Такі системи охоплюють всі аспекти діяльності підприємства, включаючи логістику, управління виробництвом, фінанси та інше.

2. Системи управління взаємовідносинами з клієнтами (англ. Customer Relationship Management або CRM). Ці системи спрямовані на взаємодію з

клієнтами, включаючи в себе управління продажами, маркетинг, обслуговування клієнтів.

Таблиця 1

Використання баз даних в інформаційних системах

Тип інформаційної системи	Тип баз даних				
	Реляційна	Нереляційна	Мережева	Об'єктно-орієнтована	Розподілена
ERP	MySQL, PostgreSQL, Oracle, SQLite...	MongoDB	-	-	-
CRM	MySQL, Oracle, SQLite, IBM Db2, MS SQL Server...	MongoDB, Cassandra, Redis, Amazon DynamoDB	-	-	-
CMS	PostgreSQL, Oracle, SQLite, MS SQL Server...	MongoDB	-	-	-
SCM	Oracle, IBM Db2, PostgreSQL, SQLite, MySQL.	MongoDB, Cassandra, Amazon DynamoDB	-	-	-
Клієнтсько-серверна	Oracle, MySQL, SQLite, MS SQL Server...	MongoDB, Redis, Cassandra, Amazon DynamoDB	-	-	-
Соціально-мережева	MySQL, PostgreSQL, Microsoft SQL Server	Neo4j, ArangoDB, Couchbase, OrientDB, MongoDB	ArangoDB, Neo4j	Db4o, ObjectDB	Datomic, CouchDB
Онлайн-геймінг	MySQL, PostgreSQL, Microsoft SQL Server	Redis, MongoDB, Cassandra	Neo4j	MongoDB	CouchDB

3. Системи управління контентом (англ. Content Management System або CMS). Вони використовуються для створення, редагування та управління цифровим контентом, наприклад веб-сайтами.

4. Системи управління ланцюгом постачання (англ. Supply Chain Management або SCM). Спрямовані на планування, виконання та контроль логістичних процесів в ланцюгу постачання.

5. Клієнтсько-серверні системи. Ці системи забезпечують взаємодію між користувачем та сервером.

6. Соціально-мережеві системи, що надають засоби для створення, обміну та взаємодії. (Наприклад, Facebook, Instagram, X...)

7. Онлайн-геймінг системи. Це системи, що забезпечують гравцям можливість спілкуватися, взаємодіяти та грати в ігри в онлайн-середовищі.

У результаті проведеного аналізу було досліджено використання баз даних в інформаційних системах в залежності від типу БД і типу ІС. Інформація про використання різних баз даних в інформаційних системах наведена в таблиці 1.

Висновки: Комп'ютерні технології розвиваються дуже швидко і завжди з'являються альтернативи традиційним технологіям, в цьому випадку NoSQL став альтернативою реляційним БД. Реляційні БД, як і NoSQL, використовуються в багатьох інформаційних системах, тому обирати тип БД потрібно ґрунтуючись на конкретних потребах проєкту та типах даних, що використовуються в ньому, так як в одній інформаційній системі можуть використовуватися різні типи БД.

ПЕРЕЛІК ПОСИЛАНЬ

1. Microsoft Docs: Non-relational data and NoSQL. Посилання: <https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data>
2. Amazon AWS: Relational Database. Посилання: https://aws.amazon.com/relational-database/?nc1=h_ls
3. A Quick-Start Tutorial on Relational Database Design. Посилання: https://www3.ntu.edu.sg/home/ehchua/programming/sql/Relational_Database_Design.html
4. Graph Based Data Model in NoSQL. Посилання: <https://geeksforgeeks.org/graph-based-data-model-in-nosql/>
5. Key-value NoSQL Database | Download Scientific Diagram. Посилання: https://www.researchgate.net/figure/Key-value-NoSQL-Database_fig1_332188615
6. SQL vs NoSQL: A Performance Comparison. Посилання: <https://www.cs.rochester.edu/courses/261/fall2017/termpaper/submissions/06/Paper.pdf>
7. SQL vs NoSQL Database – A Complete Comparison. Посилання: <https://backendless.com/sql-vs-nosql-database-a-complete-comparison/>
8. Інформаційна система — Вікіпедія. Посилання: https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0
9. CMS, CRM and ERP – What Is It and Why? Посилання: <https://qarea.com/blog/cms-crm-and-erp-what-is-it-and-why>
10. Example of Supply Chain Management Information System & How it Works for your business? <https://dnasupplychain.com/example-of-supply-chain-management-information-system/>

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ОСОБИСТОГО ОСВІТНЬОГО БЛОГУ НА ПЛАТФОРМІ REACT NATIVE

Анотація. Робота присвячена розробці та дослідженню освітнього блогу на платформі React Native для операційних систем (ОС) Android та iOS. В межах проєкту реалізована взаємодія з API, використовується гнучкість компонентів React та пакети багаторазового коду за умови забезпечення зручного інтерфейсу та безпеки користувачів..

Ключові слова: *React Native, Android, iOS, Redux, нативні можливості, JavaScript, API платформи, інтерфейс користувача.*

Вступ. В сучасному освітньому середовищі зростає попит на інноваційні методи навчання та провідні технології обміну знаннями. У цьому контексті розробка мобільних додатків стає важливим засобом надання доступу до освітнього контенту. Відповідно до цього попиту, робота присвячена розробці та дослідженню особистого освітнього блогу, що базується на платформі React Native.

React Native – це фреймворк з відкритим вихідним кодом для створення додатків ОС Android та iOS з використанням бібліотеки React і нативних можливостей платформи додатків. React Native використовує мову JavaScript для доступу до API платформи, а також для опису зовнішнього вигляду та поведінки інтерфейсу користувача за допомогою компонентів React: пакетів багаторазового, вкладеного коду.

Метою роботи є створення та подальше дослідження особистого освітнього блогу, який дозволяє користувачам здобувати та обмінюватися знаннями. Використання React Native дозволяє забезпечити швидку та ефективну розробку для платформ Android та IOS, зберігаючи високу якість та продуктивність роботи.

Основний зміст роботи. Особливостями розробки на базі фреймворку React є використання гнучких компонентів React для побудови інтерфейсу користувача, а також пакетів багаторазового коду для оптимізації та спрощення розробки. Для організації взаємодії додатку з API використана мова JavaScript, що забезпечує ефективну взаємодію з API платформ Android та iOS, а також реалізовані механізми автентифікації та авторизації для забезпечення безпеки користувачів.

При розробці використано патерн Redux, який представляє собою управлінський контейнер стану для JavaScript-додатків, що особливо часто використовується в інтерфейсах користувача та додатках на основі React. Він базується на парадигмі однозначного збереження стану додатку в одному місці, сховищі (“store”), і управління цим станом через так звані “дії” та “редюсери”. Надамо короткі характеристики складових патерну Redux.

Сховище (Store) - це центральний елемент Redux, який утримує стан всього додатку в одному об'єкті. Сховище зберігає стан додатку, приймає дії (actions) та викликає редюсери (reducers) для зміни стану.

Дії (Actions) – це об'єкти, які вказують на зміну стану та передають дані для оновлення сховища. Дії оперують інформацією, яка буде використовуватися для зміни стану, а також містять тип дії та дані для оновлення.

Редюсери (Reducers) - чисті функції, які приймають поточний стан та дію, і повертають новий стан. Редюсери обробляють конкретні типи дій та вносять зміни в стан. Вони повинні бути чистими та детермінованими.

Підписники (Subscribers) – це функції, які відстежують зміни стану та викликаються при кожній зміні. Підписники дозволяють іншим частинам додатку реагувати на зміни в сховищі.

Посередники (Middleware) – це функції, які розширюють функціональність Redux, дозволяючи обробляти асинхронні дії, логування, трансформацію дій тощо.

Патерн Redux дозволяє ефективно управляти станом додатку та полегшує відлагодження та розширення. Він зручно використовується для проєктів будь-якої складності, забезпечуючи стабільність та чистоту коду.

Ключові етапи розробки особистого освітнього блогу на базі React Native:

1. Ініціалізація проєкту React Native: `npx react-native init EducationBlog`. Встановлено необхідні залежності: `npm install redux react-redux`.

2. Створення Сховища (Store), а саме папки `redux` і файлів `actions.js`, `reducers.js` та `store.js`. Визначено дії та редюсери для блогу.

3. Підключення Redux до проєкту. В кореновому компоненті (`App.js`) імпортовано `Provider` з `react-redux`. Компоненти додатку обернуто в `Provider`, передавши йому створений Store.

4. Створення Компонентів для відображення блогу, статей, коментарів тощо.

5. Створення Дій та Редюсерів. У `actions.js` визначено дії, такі як `FETCH_POSTS`, `ADD_COMMENT`. У `reducers.js` створено редюсери для обробки цих дій та оновлення стану.

6. Зв'язок Компонентів із Redux. Використано `connect` з `react-redux` для підключення компонентів до Redux-сховища.

7. Взаємодія зі Сховищем через Дії.

8. Вдосконалення шляхом розширення дій та редюсерів, відлагодження.

9. Тестування.

Оцінка ефективності додатку здійснена у напрямках дослідження ефективності та зручності використання освітнього блогу, додатково реалізовано збір та аналіз фідбеку користувачів.

Освітній блог можна використовувати в сферах сприяння навчанню та обміну знаннями:

- в онлайн-навчанні для надання освітніх матеріалів, відеоуроків, тестів та іншого контенту для студентів будь-якого рівня;

- у якості університетських та шкільних ресурсів з метою надання студентам додаткового матеріалу для засвоєння навчального матеріалу;
- з метою обміну знаннями про програмування, нові технології та рішення;
- при обміні досвідом серед інженерів, розробників та ІТ-спеціалістів;
- у мовній освіті для отримання матеріалів з розвитку літературних та мовленнєвих навичок;
- у літературознавчих та філософських онлайн спільнотах для організації обговорення творів, течій та теорій;
- при організації онлайн-курсів з історії, психології, соціології та інших гуманітарних наук.

Наукова новизна. Робота присвячена розробці та дослідженню освітнього блогу, що використовує потужності React Native для забезпечення ефективної та зручної платформонезалежної експлуатації. Використовується гнучкість компонентів React та пакети багаторазового коду.

Висновки. Розроблено особистий освітній блог на базі фреймворку React Native для Android та iOS та проаналізовано особливості його роботи. В межах дослідження реалізована взаємодія з API за умови забезпечення зручного інтерфейсу та безпеки користувачів. Отримані результати сприятимуть подальшому розвитку освітнього контенту та навчальних платформ.

УДК 004.4'2

Г.Л. Хара¹, Н.О. Соколова¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ІНФОРМАЦІЙНА СИСТЕМА ТЕСТУВАННЯ ВЕБ-ДОДАТКІВ

Анотація. Дана робота спрямована на розробку інформаційної системи, спеціалізованої на тестуванні веб-додатків. Основною метою цього проекту є створення ефективного та надійного інструментарію для проведення тестів веб-застосунків з використанням сучасних методів інтеграції, аналізу та звітування результатів.

Ключові слова: *тестування, програмне забезпечення, автоматизація, архітектура, технології, тестові сценарії, результати, API-інтерфейс, JSON, X-path, Selenium, TestNG, WebDriverManager.*

Вступ. У епоху цифрових технологій розробка програмного забезпечення є важливою частиною поточної цифрової революції. Розробка інформаційної системи тестування веб-додатків є важливим етапом життєвого циклу програмного забезпечення. Архітектура повинна задовольняти сучасним вимогам, в тому числі і правовим аспектам [1], підлаштовуватися під вимоги, які накладаються на веб-додатки, та бути гнучкою аби тестування було максимально ефективним.

Оскільки процес розробки стає безперервним, потреба в тестуванні стає все більш важливою. Розробка програмного забезпечення, тестування полягає не лише в тому, щоб запобігти виникненню помилок, але й упевнитися, що розробляється правильна функція, яка задовольняє потреби клієнтів і досягає повернення інвестицій[2].

Тестування веб додатків визначається необхідністю гарантувати високу якість, ефективність та продуктивність під час їх використання [3].

Одним із ключових аспектів є створення модульної структури, що дозволяє легко розширювати систему для адаптації до різних видів тестувань. Це дозволяє охопити більшу частину функціоналу. Використання автоматизованих тестів дозволяє ефективно виявляти помилки та забезпечує швидке внесення змін для поліпшення продукту. Інтеграція із засобами ведення версій та засобами звітності забезпечує зручний моніторинг та аналіз результатів, що в свою чергу позитивно впливає на розробку продукту [4, 5].

Також тестування має й економічний вплив. Виявлення помилок на ранніх етапах проектування або розробки дозволяє зменшити витрати ресурсів та коштів, так як їх виправлення на ранніх етапах більше дешевша.

Постановка задачі. Мета роботи полягає в аналізі підходів розробки інформаційної системи тестування веб-додатків.

Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- Аналіз вимог.
- Проектування архітектури.
- Вибір технологій.
- Розробка інструментарію.
- Методологія тестування.
- Інтеграція і тестування.
- Оцінка та вдосконалення.

Основний зміст роботи. В роботі розглядався широкий спектр етапів процесу розробки системи тестування веб-додатків. Починаючи з аналізу вимог та проектування архітектури, було звернуто увагу на вибір оптимальних технологій, що відповідають потребам проекту. Розробка інструментарію, базуючись на методології тестування та використанні моделі Page Object, забезпечила ефективну інтеграцію та тестування. Модель "Ланцюжок викликів" в контексті Page Object дозволила послідовно взаємодіяти з елементами на сторінках веб-додатків, спрощуючи тестові сценарії та забезпечуючи чітку структуру коду.

Під час роботи були використані наступні методи та інструменти:

- TestNG: Фреймворк для написання та виконання автоматизованих тестів.
- Selenium: Набір інструментів для автоматизації веб-додатків.
- JSON: Формат обміну даними.

- XPath: Мова запитів для вибору елементів на веб-сторінках для локалізації елементів для тестів у Selenium.
- WebDriverManager: Інструмент для автоматичного керування драйверами браузера в Selenium.
- Maven: Інструмент для управління проектами на Java, зокрема для збирання, завантаження бібліотек та управління залежностями.
- GitHub: Система керування версіями.
- Модель Page Object
- Модель Ланцюжок викликів

Під час розробки використовувалися передові інструменти, такі як TestNG та Selenium, для автоматизації тестів веб-додатків. Використання формату JSON для обміну даними, XPath для локалізації елементів у Selenium та WebDriverManager для керування драйверами браузера сприяло зручній та ефективній роботі з веб-елементами.

Крім цього, інструменти, які спрямовані на управління проектами Maven та система керування версіями GitHub були використані для організації робочих процесів та спільної роботи.

Висновки. В результаті роботи була розроблена інформаційна система для тестування веб-додатку. Були знайдені невідповідності з технічним завданням та відправлені відповідні доклади. Дослідження в цій роботі дозволяє існуючим та майбутнім розробникам веб-додатків зрозуміти, як вдосконалити процес тестування для забезпечення високої якості продукту. Отримані знання можуть бути використані в освітніх програмах для навчання принципам та практиці тестування веб-додатків. Дані результати також можуть стати корисними для компаній та комерційних розробників, сприяючи покращенню їхніх тестових стратегій та продуктів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Жирова, Т. ., Котенко, Н. ., Токар, В. ., Хорольська, К. ., & Бебешко, Б. . (2022). Тестування доступності Web-додатків: Testing The Accessibility. Computer Systems and Information Technologies, (3), 89–95. <https://doi.org/10.31891/CSIT-2021-5-12>
2. Кодола Г. М. Автоматизоване тестування веб-додатків з різномірною архітектурою / Г. М. Кодола, Н. С. Волинець, І. В. Сербулова // Вісник Національного технічного університету "ХПІ". Сер. : Нові рішення в сучасних технологіях = Bulletin of the National Technical University "KhPI". Ser. : New solutions in modern technology : зб. наук. пр. – Харків : НТУ "ХПІ", 2019. – № 5 (1330). – С. 91-100.
3. Чому тестування важливо?. syntho. URL: <https://www.syntho.ai/uk/why-is-testing-important/>
4. Тестування веб-проектів: основні етапи та поради. <https://qalight.ua/baza-znaniy/testuvannya-veb-proektiv-osnovni-etapi-ta-poradi/>
5. Тестування та Автоматизація Тестування Веб-Сайтів: Шлях до Високоякісних Додатків. <https://it-rating.ua/testuvannya-ta-avtomatizatsiya-testuvannya-veb-saytiv-shlyah-do-visokoyakisnih-dodatkiv>

КОМП'ЮТЕРНА ТЕХНОЛОГІЯ МОНІТОРИНГУ СТАНУ ЛІСОВОГО ПОКРИВУ ПРИРОДНОГО ЗАПОВІДНИКА «ДРЕВЛЯНСЬКИЙ» ЗА ДАНИМИ SENTINEL-2

Анотація. Представлено комп'ютерну технологію, яка використовується для виявлення та аналізу змін у лісовому покриві заповідника. У роботі детально представлені складові цієї технології, які дозволяють ідентифікувати природні та антропогенні фактори, впливаючи на дану екосистему. Здійснена апробація технології за допомогою часового ряду зображень Sentinel-2 за період з 2016 по 2023 роки на території природного заповідника «Древлянський».

Ключові слова: інформаційна технологія, моніторинг лісів, комп'ютерне моделювання, Sentinel-2.

Вступ. Ситуація з лісовими ресурсами в Україні не відповідає сучасним еколого-економічним вимогам суспільства. Надмірне використання лісів, несанкціонована вирубка, а також втрата лісових територій під неекологічні цілі спричиняють загальне погіршення ситуації [1]. Отримання і обробка даних про ліси не відповідають потребам управління природними ресурсами. Не визначено ефективні методи моніторингу, контролю та управління лісовими масивами на різних рівнях управління. Узгодженість процедур, форматів, термінів та обмін інформацією між органами влади щодо цих питань відсутня. Інформаційна база про кількість, якість та стан лісових ресурсів не відповідає потребам управління. Щоб вирішити ці проблеми, потрібно створити ієрархічну систему управління та інформаційного забезпечення, яка б відображала різні рівні управління належним чином.

В Україні поки не створено державної програми для дистанційного моніторингу лісів, тому зараз спостереження проводяться в основному на місцях. Впровадження технологій дистанційного моніторингу може спростити інвентаризацію, знизити витрати та майже усунути вплив суб'єктивних факторів. Використання розвинутої геоінформаційної системи, побудованої на базі даних від дистанційного зондування, може стати потужним інструментом для швидкого та точного аналізу інформації, необхідної для прийняття управлінських рішень на будь-якому рівні управління.

Метою роботи є розробка методики моніторингу стану лісового покриву природного заповідника «Древлянський».

Постановка задачі. Актуальною науково-прикладною задачею для досягнення мети потрібно вирішити наступні завдання:

- дослідити сучасний стан лісового покриву природного заповідника «Древлянський»;
- проаналізувати існуючі методи дистанційного зондування Землі для моніторингу лісів;

- виконати попередню обробку, радіометричне калібрування та атмосферну корекцію первинних даних;
- розробити технологію на основі неконтрольованої та контрольованої класифікації знімків Sentinel-2 для дослідження стану та незаконних вирубок лісів.

Методи досліджень. Запропонована в роботі комп'ютерна технологія моніторингу стану лісового покриву складається з наступних етапів:

- отримати доступ до різночасових супутникових знімків з Sentinel-2;
- розрахувати індекси рослинності (NDVI, RVI, GRVI) за формулами з табл.1 [2];
- виконати контрольовану класифікацію для отримання міської рослинності (алгоритм максимальної правдоподібності у цьому дослідженні використовувався алгоритм максимальної правдоподібності)
- виконати неконтрольовану класифікацію індексів рослинності з використанням комплексного методу K-means та головних компонентів (PCA) [3];
- виявити порушення деревостанів;
- виявити зміни у неконтрольованій класифікації на основі аналізу головних компонент та побудувати карти змін по роками;
- побудувати карту змін лісового покриву з 2016 року по 2023 рік.

Таблиця 1

Формули індексів рослинності [2]

Індекс рослинності	Формула для супутника Sentinel-2
NDVI	$NDVI = \frac{B8 - B4}{B8 + B4}$
RVI	$RVI = \frac{B8}{B4}$
GRVI	$GRVI = \frac{B8}{B3}$

Для реалізації запропонованої технології було використано QGIS та мову програмування Python. На рис.1 наведено результат контрольованої класифікації, в якій визначаються класи рослинності та не рослинності на зображеннях. Вибираються зразки для навчання класифікаційних моделей. Важливо вибрати зразки, які репрезентативно відображають кожен клас і включають різноманітні спектральні характеристики (різні типи рослин, ґрунти, нерослинні покриття тощо).

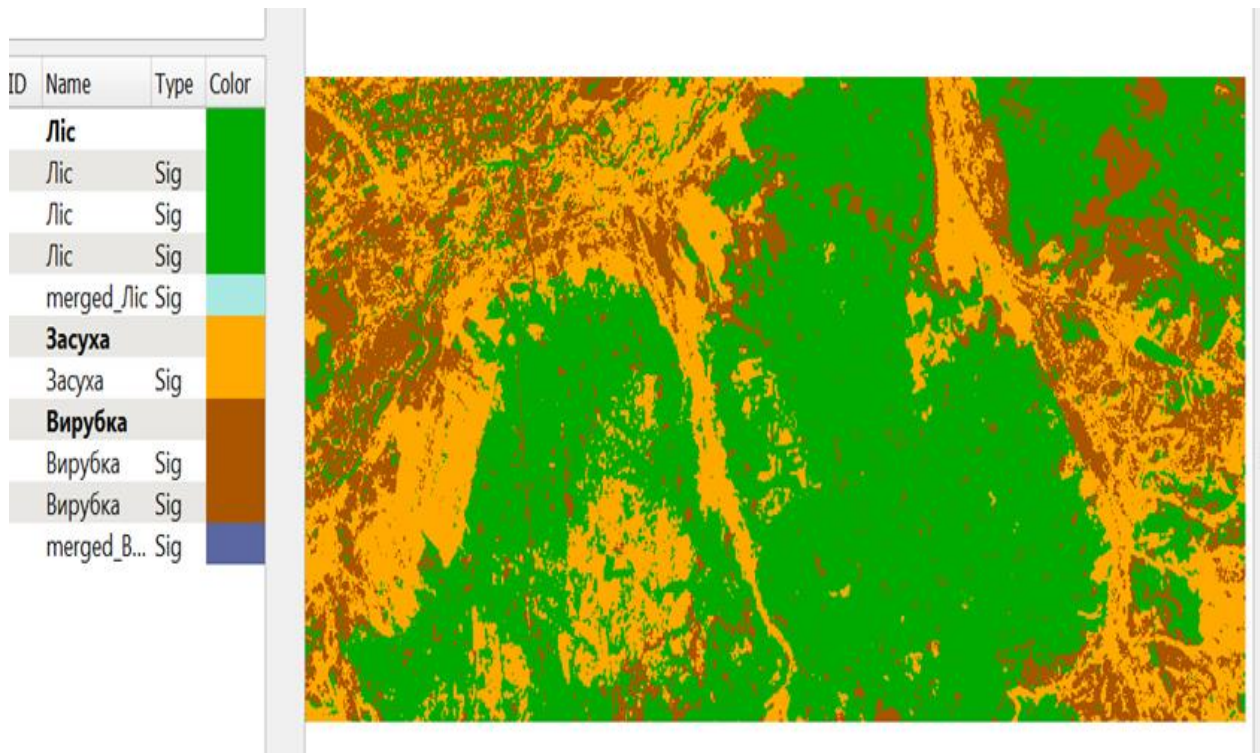


Рис. 1. Контрольована класифікація

Обговорення результатів. Аналізуючи отримані результати з рис.2, можна побачити суттєві зміни лісового покриву. Градієнт синього кольору на рис.2 вказує на зміни лісового покриву протягом семи років, зелений – змін в цій частині не відбулось.

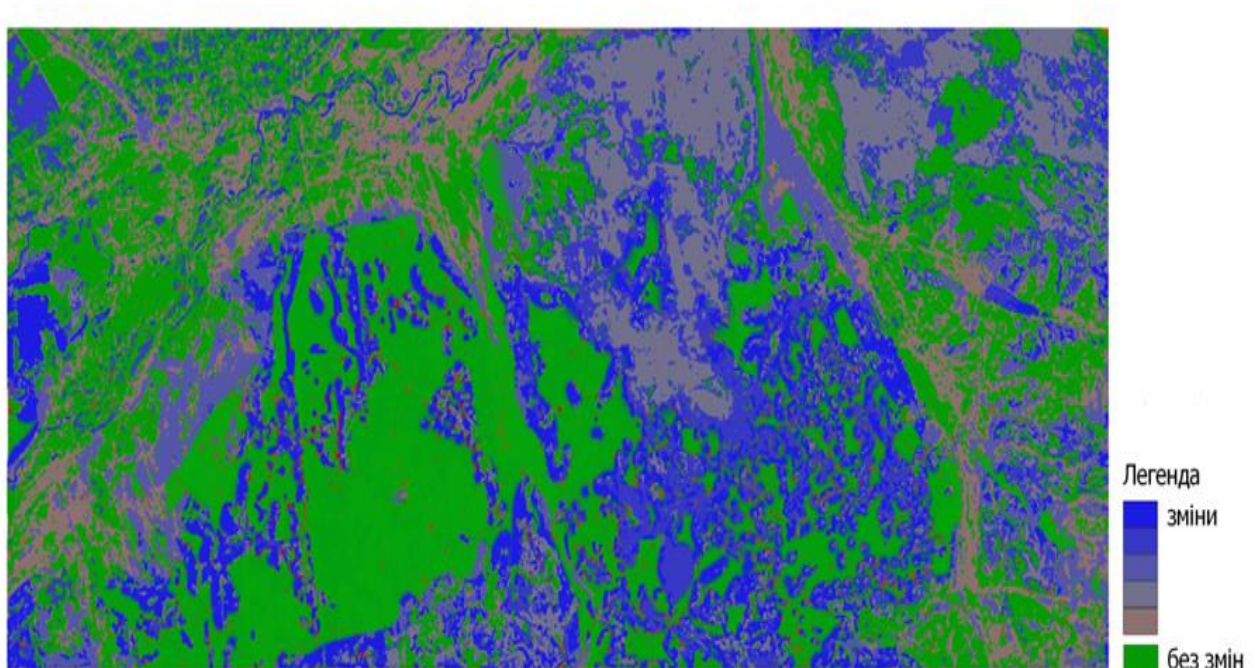


Рис. 2. Зміни лісового покриву заповідника «Древлянський»

Зміни в структурі і стані лісового покриву у заповіднику за 7 років були результатом впливу таких факторів як:

- лісові пожежі;
- усихання дерев, спостерігалось надмірне усихання дерев, впливаючи на стан лісової екосистеми. Передчасне усихання та втрата листя спричинили зміни у візуальному вигляді лісового покриву та його структури;
- санітарні рубки, додатковим фактором, що вплинув на структуру лісового покриву, були санітарні рубки. Ці заходи були призначені для видалення хворих або відмерлих дерев з метою запобігання поширенню хвороб та паразитів.

Наукова новизна полягає у використанні комплексних методів класифікації (контрольованої та неконтрольованої), а також методів класифікації усохлих деревостанів на основі РСА та K-means для детального аналізу лісового покриву та виявлення змін.

Висновки. В роботі розроблено і описано комп'ютерну технологію моніторингу стану лісового покриву, яка включає в себе аналіз різних видів супутникових знімків, обчислення вегетаційних індексів та класифікацію рослинності.

Результати дослідження підкреслюють важливість постійного моніторингу та аналізу стану лісового покриву у природних заповідниках. Вони надають інформацію для прийняття рішень щодо збереження та відновлення лісів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Shimizu K., Ota T., Mizoue N. Detecting Forest Changes Using Dense Landsat 8 and Sentinel-1 Time Series Data in Tropical Seasonal Forests. *Remote Sens.* 2019, 11, 1899.
2. Wang W., Liu R., Gan F., Zhou P., Zhang X., Ding L. Monitoring and Evaluating Restoration Vegetation Status in Mine Region Using Remote Sensing Data: Case Study in Inner Mongolia, China. *Remote Sens.* 2021, 13, 1350.
3. Bullock E.L., Woodcock C.E., Olofsson P. Monitoring tropical forest degradation using spectral unmixing and Landsat time series analysis. *Remote Sens. Environ.* 2018, 238.

РОЗРОБКА ТА ДОСЛІДЖЕННЯ МЕССЕНДЖЕРА НА БАЗІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Анотація. Дана робота присвячена розробці месенджера на базі технології блокчейн з використанням мови програмування Python, акцентуючи увагу на безпеці даних, децентралізації та прозорості комунікацій.

Ключові слова: месенджер, python, блокчейн. миттєве повідомлення.

Вступ. Сучасний світ цифрових комунікацій постійно розвивається, і з розвитком технологій з'являються нові виклики та можливості. Однією з передових технологій, яка пропонує нові підходи до безпеки та приватності в цифровому просторі, є технологія блокчейн. Розробка месенджера на базі блокчейну є актуальною, оскільки може відкрити нові горизонти у спілкуванні, забезпечуючи високий рівень безпеки, невідстежуваність повідомлень та гарантії незмінності даних.

Постановка задачі. Сучасний технічний прогрес динамічно розвивається, щодня поповнюючи ринок програмного забезпечення новими системами, які переслідують абсолютно різноманітні цілі. Кожного дня в сучасному світі активно розвивається ніша різноманітних месенджерів та різного роду онлайн-чатів. Люди активно користуються усіма запропонованими продуктами для організації ефективної комунікації. Головною проблемою у сфері цифрових месенджерів на сьогодні є питання безпеки та конфіденційності. Більшість існуючих рішень залежать від централізованих серверів, які можуть стати мішенню для хакерських атак, а також становити ризик для конфіденційності користувачів. Додатково, існують питання щодо цензури та контролю над інформацією з боку центральних авторитетів. Вирішенням даної проблеми може бути різного роду шифрування й кодування, але далеко не кожен розробник інтегрує такі технології в свій продукт. Враховуючи вищезгадане слід зазначити на актуальності даних досліджень.

Основний зміст роботи. Технологія обміну миттєвими повідомленнями (IM) - це тип онлайн-чату, що дозволяє передавати текст у режимі реального часу через Інтернет або іншу комп'ютерну мережу. Повідомлення зазвичай передаються між двома або більше сторонами, коли кожен користувач вводить текст і ініціює передачу одержувачу(ам), які всі підключені до спільної мережі [1]. Відмінність від електронної пошти полягає в тому, що розмови через обмін миттєвими повідомленнями відбуваються в режимі реального часу (тому "миттєво"). Більшість сучасних програм миттєвого обміну повідомленнями (іноді їх називають «соціальними месенджерами», «додатками для обміну повідомленнями» або «додатками для чату») використовують технологію push, а

також додають інші функції, такі як смайли (або графічні смайли), передача файлів, чат-боти, передача голосу через IP або відео можливості чату [2].

Системи миттєвого обміну повідомленнями, як правило, полегшують з'єднання між певними відомими користувачами (часто використовують список контактів, також відомий як «список друзів») і можуть бути окремими програмами або інтегрованими, наприклад, у систему [3].

Залежно від протоколу миттєвого обміну повідомленнями технічна архітектура може бути однорангова (пряма передача точка-точка) або клієнт-сервер (центр обслуговування миттєвих повідомлень повторно передає повідомлення від відправника до пристрою зв'язку). Зазвичай він відрізняється від обміну текстовими повідомленнями, який зазвичай простіший і використовує мережі стільникового телефону [4].

Незаписні повідомлення (OTR) – це криптографічний протокол, який забезпечує шифрування розмов миттєвих повідомлень. OTR використовує комбінацію алгоритму симетричного ключа AES із довжиною ключа 128 біт, обміну ключами Діффі–Хеллмана з розміром групи 1536 біт і хеш-функції SHA-1. На додаток до автентифікації та шифрування, OTR забезпечує пряму секретність і піддатливе шифрування. Основною мотивацією протоколу було забезпечення незаперечної автентифікації для учасників розмови, зберігаючи розмови конфіденційними, як приватна розмова в реальному житті або не для запису в журналістських джерелах. Це відрізняється від інструментів криптографії, які створюють вихідні дані, які пізніше можна використовувати як верифікований запис події зв'язку та особи учасників [5]

Блокчейн (англ. blockchain) – розподілена база даних, що зберігає впорядкований ланцюжок записів (так званих блоків), що постійно довшає. Кожен блок містить часову позначку, хеш попереднього блоку та дані транзакцій, подані як хеш-дерево Merkle, де вузли даних представлені листками. Мітка часу доводить, що дані транзакції існували на момент створення блоку [6]. Інформація про транзакції зазвичай надається відкритою, не шифрованою. Захистом від підробки та спотворення слугує включення хешу всього блоку у наступний блок. Тому внесення змін в один з блоків вимагає відповідних змін в усіх блоках після нього, що зазвичай виявляється або дуже складно, або дуже коштовно.

Алгоритм системи працює наступним чином (рис. 1): в першу чергу система отримує відкористувача його логін і текст повідомлення. Пустий логін припустимий, в такому разі система продовжує роботу. Пустий текст повідомлення неприпустимий, в такому разі система припиняє виконання алгоритму і чекає наступної ініціації від користувача, після чого починає спочатку. Якщо ведені дані коректні - система створює новий блок, який містить в собі дані про відправника, отримувача та текст повідомлення.

Система створює POST-запит до сервера і відправляє його. Якщо сервер повертає код 200 (підтвердження) – система надсилає повідомлення усім клієнтам. Якщо сервер повертає будь-який інший код – система закінчує роботу алгоритму без відправлення повідомлення

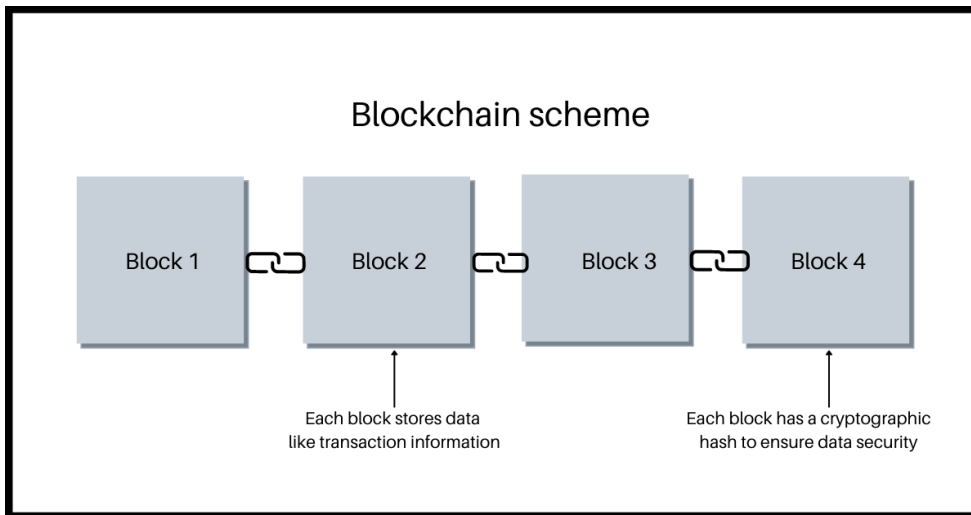


Рис. 1. Загальна схема блокчейну

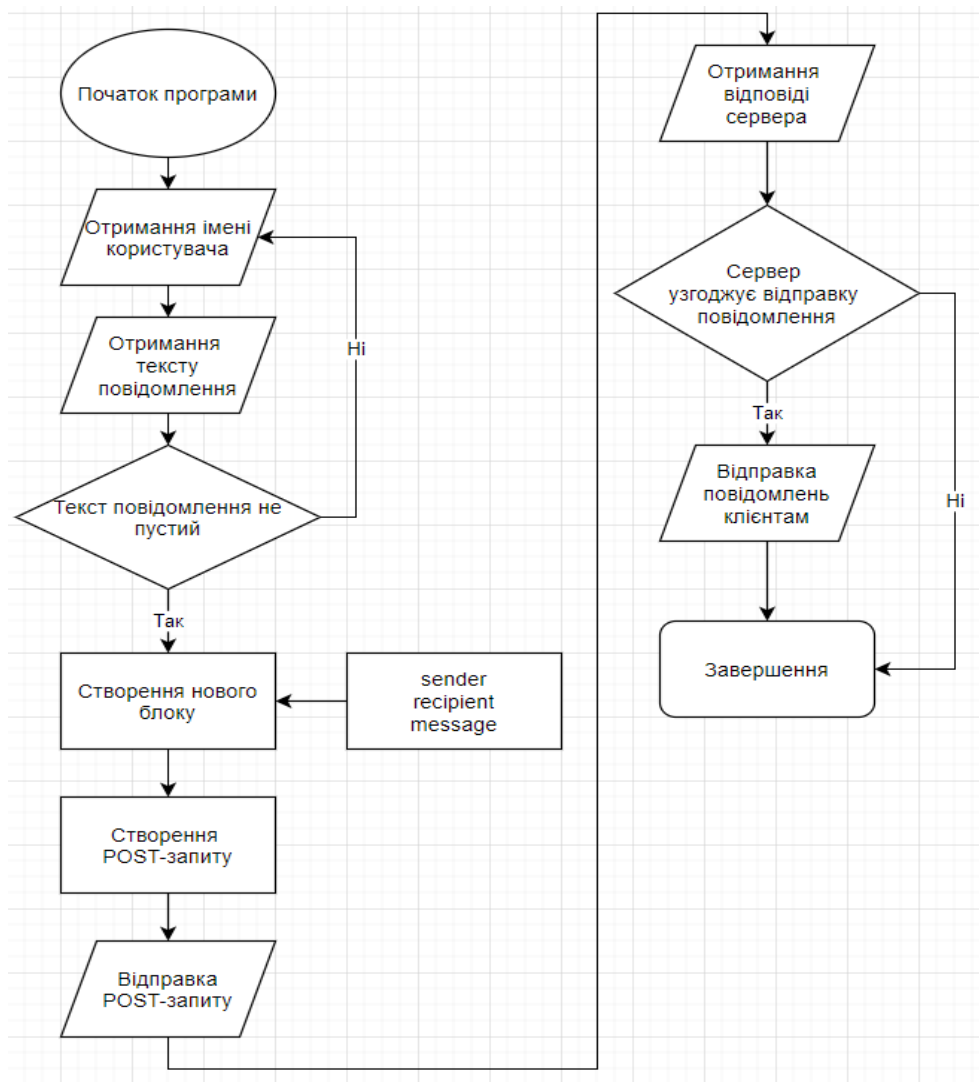


Рис. 2. Алгоритм роботи системи

В першу чергу необхідно спроектувати майбутній функціонал системи. Системні вимоги можна ефективно охопити шляхом моделювання поведінки системи за допомогою діаграм варіантів використання в UML.

UML діаграма варіантів використання зображена на рис. 3, де показано усі можливі дії користувача в системі на поточний момент

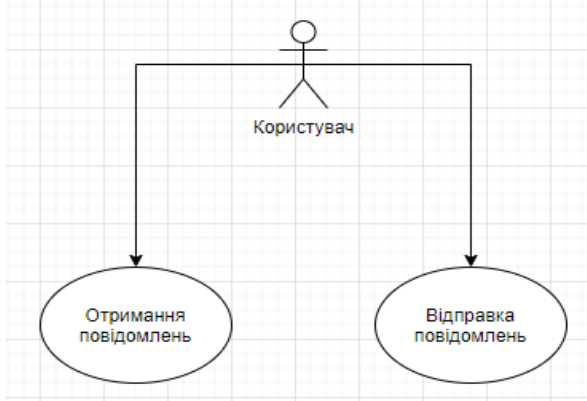


Рис. 3. Діаграма варіантів використання

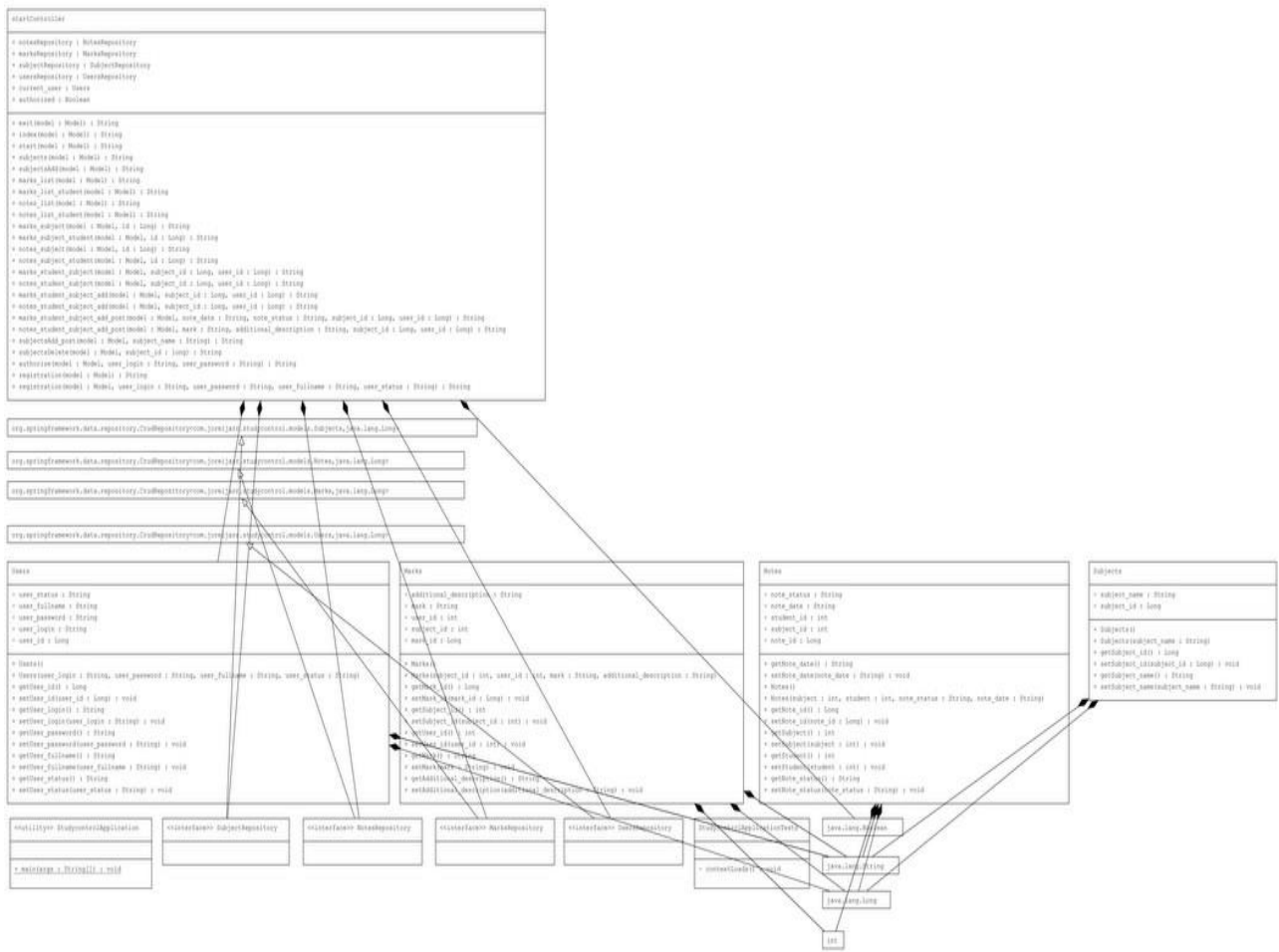


Рис. 4. Діаграма класів

Для проектування внутрішньої будови системи було розроблено UML діаграму класів, яка ілюструє внутрішню структуру проекту (рис.4). Діаграми класів є фундаментальними для процесу моделювання об'єктів і моделюють статичну структуру системи. Залежно від складності системи ви можете використовувати одну діаграму класів для моделювання всієї системи або кілька діаграм класів для моделювання компонентів системи.

Наукова новизна полягає в створенні власної реалізації простої системи месенджера з використанням технології блокчейн. В процесі виконання проекту було створено повноцінну програмну систему месенджера підвищеного рівня безпеки, яка працює через хмарне сховище. Система реалізована на мови програмування Python та бібліотеки Flask.

Висновки. Розроблено месенджер на базі технології блокчейн з використанням мови програмування Python, акцентуючи увагу на безпеці даних, децентралізації та прозорості комунікацій. На даному етапі система повністю функціональна та готова до використання в реальних умовах.

ПЕРЕЛІК ПОСИЛАНЬ

1. Instant messaging: This conversation is terminated. [Electronic resource]. – Access mode: http://news.bbc.co.uk/2/hi/uk_news/magazine/8698174.stm – Title from the screen.
2. AIM has been discontinued as of December 15, 2017 [Electronic resource]. – Access mode: help.aol.com – Title from the screen.
3. The rise of messaging platforms. // The Economist, через Chatbot News Daily. – 2018.
4. Google's Allo puts AI in a messaging app. // Engadget. – 2020.
5. Шун, Бен. RIP: Google Allo dies today, a look back at the groundwork of Google Messages and RCS. // 9to5Google. – 2019.

УДК 004.9

А.К. Пінковська¹, Н.О. Соколова¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ КОНФІГУРУВАННЯ КОМПОНЕНТІВ АВТОМАТИЗОВАНОЇ СИСТЕМИ ВИРОБНИЦТВА

Анотація. Актуальність даної роботи полягає у розробці інформаційної технології для конфігурування компонентів автоматизованих систем виробництва з метою оптимізації та підвищення ефективності виробничих процесів. Робота включає аналіз сучасних тенденцій у галузі інформаційних технологій та розробку нових підходів до конфігурування систем.

Ключові слова: інформаційні технології, конфігурування, автоматизовані системи, виробництво, оптимізація.

Вступ. Упродовж існування інформаційних систем, їхні зміни безперервно відбуваються відповідно до розвитку технологій та корпоративного бізнесу. Виникнення персональних комп'ютерів надало новий поштовх для розвитку

інформаційних технологій, сприявши ефективному зв'язку в межах інформаційних систем і впровадженню нових методів управлінської діяльності [1]. Аналіз розвитку інформаційних систем підкреслив постійні прагнення підприємств поліпшити обробку інформації для управлінських рішень, що призвело до створення нових програмних засобів та їхньої модернізації.

Керівництво підприємств постійно ставило завдання підвищення ефективності обробки інформації, що призвело до виникнення якісних програмних засобів та їх оновлення. Важливим кроком є впровадження інформаційної технології, яка гармонійно вписується в структуру управління конкретного підприємства. Складність впровадження інформаційної технології пропорційна розміру підприємства та рівню його розвитку.[1]

В умовах швидкого технологічного прогресу і високої конкуренції виробництва стає важливим завданням розробка ефективних методів конфігурування компонентів автоматизованих систем для досягнення оптимальної продуктивності та зниження витрат.

Підвищення продуктивності бізнесу здійснюється за допомогою впровадження технологій ERP, MIS і BI. Застосування цих інструментів призводить до досягнення синергетичних ефектів, автоматизації та координації роботи всіх підрозділів підприємства, успішної реалізації стратегічних програм та підвищення конкурентних переваг. Це сприяє гармонійному розвитку та зміцненню позицій компанії на ринку.[1]

Постановка задачі. Метою даної роботи є розробка інформаційної технології конфігурування, яка дозволить оптимізувати виробничі процеси шляхом ефективного вибору та налаштування компонентів автоматизованої системи.

Основний зміст роботи. Проведений аналіз поняття "конфігуратора" як системи показав, що в будь-якому конфігураторі можна виділити три частини: система, яка піддається конфігурації (конфігуруєма або сконфігурована система); система, яка визначає конфігурацію (аналітична система модельної конфігурації); система, яка виконує конфігурацію (конфігуруюча система).

Web-сторінки виконують роль конфігуруючої системи, тоді як система, яку конфігурують, найчастіше є лише аналітичним відображенням реальних об'єктів, наприклад, конфігуратори автомобілів, комп'ютерів, інтер'єру, меблі та інше. У цьому випадку аналітичною системою модельної конфігурації є інформація про те, які компоненти, в якій кількості та з якими характеристиками використовуються для опису конкретного автомобіля, складеного ПК або обладнаної меблею кімнати. Потім, як правило, ця інформація (вже як сконфігурована система) зберігається і відправляється менеджеру для узгодження з клієнтом. Розглянемо більш складні конфігуратори - ті, що забезпечують конфігурацію реальних ІТ-систем. У таких конфігураторах всі три складові (конфігурована система, аналітична система модельної конфігурації та конфігуруюча система) є повноцінними і явно вираженими. До них, наприклад, відносяться: конфігуратори веб-додатків, такі як: конфігуратори Інтернет-форумів, конфігуратори Інтернет-магазинів, конфігуратори будь-яких систем,

які базуються на веб-інтерфейсі або мають веб-інтерфейс; конфігуратори систем ІТ в цілому та розподілених систем.[2]

Обґрунтування вибору інформаційних технологій для конфігурування розпочинається аналізом сучасних тенденцій у галузі інформаційних технологій. Детальне вивчення основних напрямків розвитку інформаційних технологій у виробництві розкриває перспективи впливу цифровізації та Інтернету речей на автоматизовані системи. Слідом за цим проводиться системний аналіз попередніх досліджень у галузі конфігурування, визначення ключових критеріїв ефективності та їх впливу на виробничі показники. Важливим етапом є дослідження та аналіз сучасних методів конфігурування в автоматизованих системах виробництва, огляд існуючих підходів та їх ефективності. Це включає ретроспективний аналіз технологій конфігурування та дослідження успішних кейсів.

Визначення вимог до конфігурування в контексті автоматизованих систем ставить завдання встановлення основних критеріїв з точки зору оптимізації ресурсів та забезпечення стійкості системи. Урахування вимог до відкритості та масштабованості систем є ключовим для успішної інтеграції.

Остаточним етапом є розробка та впровадження інформаційної технології конфігурування, яка базується на попередніх аналізах та враховує всі аспекти, визначені на ранніх етапах.[3]

Створення моделі конфігурації системи:

- Розроблення математичної моделі конфігурації компонентів системи виробництва.
- Визначення ключових параметрів та залежностей між компонентами.
- Розробка програмного забезпечення для конфігурування компонентів:
- Реалізація інтерфейсу для зручного конфігурування системи операторами.
- Забезпечення можливості автоматизованого вибору оптимальних конфігурацій з урахуванням вхідних умов.

Наукова новизна. Робота є новаторською у визначенні підходів до конфігурування компонентів автоматизованих систем виробництва, включаючи розробку ефективної інформаційної технології, яка враховує сучасні технологічні та виробничі виклики.

ПЕРЕЛІК ПОСИЛАНЬ

1. Пурій Г. М. Інформаційні системи і технології в управлінні діяльністю підприємства. Ефективна економіка. 2019. № 6. С. 285-286.
2. Задоров, В. Б. До розвитку концепції «конфігураторів» для побудови архітектури інформаційних технологій організаційних антропогенних систем [Текст]/В. Б. Задоров, А. А. Васильев // Управління розвитком складних систем. - 2011. - Вип. 6. - С. 110-111.
3. Системи електронної комерції: створення, просування і розвиток : монографія / О. М. Юдін, М. В. Макарова, Р. М. Лавренюк. – Полтава : РВВ ПУЕТ, 2011. – 201 с.

РОЗДІЛ 3

ПРОГРАМНІ ЗАСОБИ УПРАВЛІННЯ, ЗБОРУ, ОБРОБКИ І ПЕРЕДАЧІ ІНФОРМАЦІЇ

УДК 004.932:528.854

І.М. Удовик¹, Л.І. Мещеряков¹, Г.Г. Швачич¹, С.О. Деменков¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РЕАЛІЗАЦІЯ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ КОДУ ДЛЯ ORM-СИСТЕМИ ACTIVEJDBC З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ANOTATION PROCESSING

Анотація. Описано процес покращення користувацького досвіду при роботі з ORM-системою ActiveJDBC. Покращення досягається за рахунок автоматичної генерації коду, що дозволяє маніпулювати з ActiveJDBC об'єктами як з POJO.

Ключові слова: *ActiveJDBC, ORM, генерація коду, Java, Dynamic Proxy, Annotation Processor.*

Вступ. Стрімкий розвиток галузі розробки програмного забезпечення призвів до появи різноманітних шаблонів та стратегій маніпуляцій з базами даних та, відповідно інструментів, що їх реалізують. Одним з шаблонів проектування, що набуває все більшої популярності є Active Record. Цей шаблон дозволяє інкапсулювати всю логіку роботи з таблицею бази даних у сам клас моделі, завдяки чому зникає необхідність створювати додаткові сервісні класи, які будуть відповідати за CRUD операції. Однією з реалізацій цього підходу є ORM-система ActiveJDBC. ActiveJDBC розвивається як відкрите програмне забезпечення і є популярним варіантом для розробки додатків на Java, які взаємодіють з реляційними базами даних.

Відповідно до документації [1], основними принципами, що сповідує ActiveJDBC є:

- він виводить метадані з бази даних;
- конфігурація здійснюється на основі домовленостей;
- немає потреби вивчати іншу мову запитів. SQL буде досить;
- код часто читається як англійський текст;
- немає сесій, немає прикріплення до сесій, немає повторного приєднання до сесій;
- немає менеджерів збереження;
- моделі є легкими, простими POJO;
- немає Proxy. Те, що ви пишете, те і отримуєте;
- немає геттерів і сеттерів. Ви можете їх писати, якщо бажаєте;
- немає DAO і DTO;
- немає анемічної моделі домену;

Хоча ActiveRecord і декларує, що класи моделей мають бути POJO, вони є занадто легкими, що не є проблемою при використанні цього інструмента у маленьких проєктах, але при зростанні кодової бази це призводить до появи великої кількості шаблонного коду. Наприклад, якщо треба отримати доступ до атрибутів кортежа відношення, ActiveRecord пропонує скористатися «get» та «set» методами, та передати у них саму назву атрибуту, що може призвести до помилок програміста по неухважності і значно ускладнює написання коду, особливо коли у проєкті, який використовує ActiveRecord у якості ORM використовуються типи даних, що не підтримуються ActiveRecord. У якості прикладів таких типів є `java.time.LocalDate` та `java.time.LocalDateTime` що були додані ще у Java 8. Одним з обхідних шляхів є використання власних методів, що будуть додаватися у ActiveRecord, і які будуть робити всі необхідні перетворення, приклад яких зображено на рис. 1.

```
2 usages
public LocalDate getBirthDate() {
    // вимушені власноруч перевіряти значення поля на null
    // та робити перетворювання
    return Optional.ofNullable(this.getDate(BIRTH_DATE)) Optional<Date>
        .map(Date::toLocalDate) Optional<LocalDate>
        .orElse( other: null);
}

2 usages
public LocalDateTime getAnotherDate() {
    // вимушені власноруч перевіряти значення поля на null
    // та робити перетворювання
    return Optional.ofNullable(this.getTimestamp(ANOTHER_DATE)) Optional<Timestamp>
        .map(Timestamp::toLocalDateTime) Optional<LocalDateTime>
        .orElse( other: null);
}

no usages
public void setAnotherDate(LocalDateTime anotherDate) {
    // вимушені власноруч перевіряти значення поля на null
    // та робити перетворювання
    this.setTimestamp(ANOTHER_DATE, Optional.ofNullable(anotherDate) Optional<LocalDateTime>
        .map(Timestamp::valueOf) Optional<Timestamp>
        .orElse( other: null));
}
```

Рис. 1. Приклад написання гетерів та сетерів з `java.time.LocalDate` та `java.time.LocalDateTime`

Наступною особливістю є те, що базовий абстрактний клас для ActiveJDBC об'єктів Model не імплементує методи equals() та hashCode(), що призводить до того, що не можливо порівняти два ActiveJDBC об'єкти між собою, а також унеможливорює використання таких об'єктів в якості ключів для хеш колекцій. Тому, якщо вони потрібні, то знову ж таки, користувачі вимушені їх написати власноруч.

```
@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || getClass() != o.getClass()) return false;
    FancyTable that = (FancyTable) o;
    return Objects.equals(this.getFirstName(), that.getFirstName()) &&
        Objects.equals(this.getBirthDate(), that.getBirthDate()) &&
        Objects.equals(this.getAnotherDate(), that.getAnotherDate());
}

@Override
public int hashCode() {
    return Objects.hash(this.getFirstName(), this.getBirthDate(), this.getAnotherDate());
}
```

Рис. 2. Перевизначені методи equals() та hashCode()

Ще одним потенційним обмеженням є те, що метод toString() також не перевизначено, що іноді може бути не зручно, наприклад, коли треба вивести якусь інформацію у лог.

```
@Override
public String toString() {
    return "FancyTable{firstName = " + (this.getFirstName() == null ? null : "" + this.getFirstName() + "")
        + ", birthDate = " + (this.getBirthDate() == null ? null : "" + this.getBirthDate() + "")
        + ", anotherDate = " + (this.getAnotherDate() == null ? null : "" + this.getAnotherDate() + "")
        + "}";
}
```

Рис. 3. Перевизначений метод toString()

Також, необхідно зазначити що ще одним недоліком цього інструмента являється відсутність можливості використовувати породжувальний патерн Будівельник для створення нових об'єктів.

Тому, залишається актуальним розробка інструменту, що буде автоматично генерувати шаблонний код для ORM-системи ActiveJDBC.

Постановка задачі. Для досягнення поставленої мети сформовані і вирішені такі завдання:

- провести аналіз існуючих підходів до генерації шаблонного коду у мові програмування Java;
- розробити інструмент генерації шаблонного програмного коду для ORM-системи ActiveJDBC;
- забезпечити повноцінне функціонування заявлених функцій, таких як:
 - перевизначення методів, toString(), equals() та hashCode();
 - генерація «гет» та «сет» методів, назви яких відповідають специфікації JavaBeans [2];
 - генерація коду реалізації шаблону Будівельник.

Основний зміст роботи. Так як розроблюваний інструмент повинен мати можливість згенерувати потрібний користувачу код, то спочатку, потрібно з'ясувати, яким чином можливо досягти цієї мети у мові програмування Java. Наразі існує декілька етапів, впродовж яких Java дозволяє втручатися у програмний код і робити з ним щось додаткове, що не було явно зазначено програмістом. Їх можна умовно поділити на: генерація коду під час виконання програмного коду, та генерація коду на етапі компіляції програмного коду.

Генерація коду під час виконання є доволі розповсюдженим явищем і використовується у багатьох різних фреймворках та бібліотеках. У якості прикладу такого фреймворку можна навести Spring Framework. Відповідно до дослідження проведеного Б. Вермером [3], що провів опитування більш ніж двох тисяч респондентів, Spring використовують 60% з них.

Генерація коду під час виконання може бути у свою чергу умовно поділена на:

- генерація коду під час завантаження класу;
- генерація коду під час виклику відповідних методів-генераторів;

Обидва підходи використовують завантажувач класів (ClassLoader) у якості основного інструмента.

Генерація коду від час компіляції може виконуватись завдяки Maven плагінам, але зазвичай використовується підхід обробки анотацій, як найбільш універсального та не залежного від засобу автоматизації роботи з програмними проектами «Apache Maven».

Обробник анотацій (Annotation Processor) – це інструмент в середовищі розробки Java, який дозволяє розробникам створювати власні анотації та визначати логіку, пов'язану з ними. Він грає важливу роль у сучасній Java розробці, дозволяючи автоматизувати багато рутинних завдань і забезпечувати додатковий функціонал для написаного коду.

Існує цикл, якому слідкує компілятор javac, що дозволяє кожному обробнику анотацій генерувати код для анотацій, які відкриті саме для нього. Ось порядок цього процесу: [4]

1. Позначаємо елемент анотацією, яка відкрита для обробника анотацій.

2. javac розпочинає компіляцію класів проекту. Він вже обізнаний з усіма наявними процесорами анотацій, оскільки ми включили їх у classpath.

3. javac створює файли з розширенням «.class» під час першого проходу обробки. Оскільки javac вже обробив файли з розширенням «.java», то він вже знає про анотації, які були використані. Після цього він передає керування відповідному процесору анотацій.

4. На цьому кроці процесор анотацій починає виконувати свою роботу. Тут він може генерувати додаткові файли з розширенням «.java», або робити будь яку іншу роботу.

5. На цей момент всі початкові файли «.java» були скомпільовані, але тільки-но згенеровані «.java» файли ще не пройшли обробку. Тим не менш, javac виявить це і розпочне інший раунд обробки з першого кроку.

Схематичне зображення роботи обробника анотацій зображено на рис. 4.

Раунди обробки

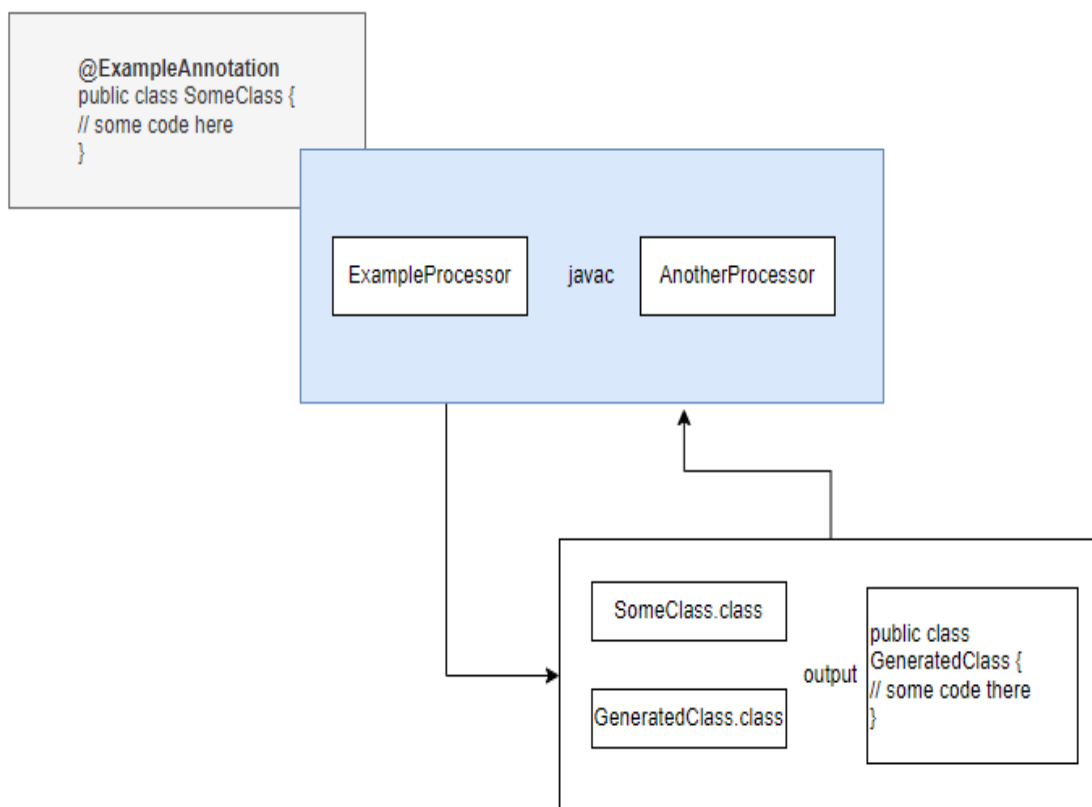


Рис. 4. Схематичне зображення роботи обробника анотацій

Результати аналізу існуючих підходів до генерації коду наведено у таблиці 1.

Таблиця порівняльних характеристик методів генерації коду

Характеристика	Генерація коду від час виконання	Генерація коду під час компіляції
Створює додаткове навантаження на процес компіляції	Ні	Так
Створює додаткове навантаження на систему при виконанні коду	Так	Ні
Дозволяє перевизначити метод equals()	Так	Так
Дозволяє перевизначити метод hashCode()	Так	Так
Дозволяє перевизначити метод toString()	Так	Так
Дозволяє згенерувати нові методи	Так, але цей код неможливо використати програмістом при написанні нового коду	Так
Дозволяє згенерувати класи-будівельники	Так, але цей код неможливо використати програмістом при написанні нового коду	Так

Грунтуючись на даних з вищенаведеної таблиці робимо висновок, що для досягнення мети найоптимальнішим рішенням буде використовувати підхід опрацювання анотацій.

Після проведення генерації класів-обгортки для моделей ActiveJDBC були отримані наступні результати залежностей кількості строк коду, що необхідно написати без, та з використанням генератора шаблонного коду. Результати наведено у таблиці 2.

Наукова новизна отриманих результатів полягає у тому, що вперше було розроблено програмне забезпечення, яке полегшує взаємодію з ActiveJDBC компонентами, та дозволяє додавати функціонал, який не був закладений у ActiveJDBC його розробником.

Таблиця залежності кількості строк коду від кількості очікуваних атрибутів таблиці

Кількість атрибутів таблиці	Без використання генератора шаблонного коду та без потреби у перетворенні даних	Без використання генератора шаблонного коду та з перетворенням даних	Отримана кількість строк коду з використанням генератора шаблонного коду
1	59	63	12
2	74	82	13
3	89	101	14
4	104	120	15
5	119	139	16
6	134	158	17
7	149	177	18
8	164	196	19
9	179	215	20
10	194	234	21

Висновки. В результаті цієї роботи теоретично і на практиці були досліджені підходи генерації програмного коду у мові програмування Java. Був розроблений обробник анотацій, що дозволяє автоматично на етапі компіляції згенерувати код, що буде взаємодіяти з об'єктами ORM системи ActiveJDBC.

Розроблений інструмент дозволяє:

- вивільнити робочий час програміста від написання шаблонного рутинного коду, та зосередитись на вирішенні поставлених перед ним задач, а не на обслуговуванні інструмента взаємодії з базою даних;
- зменшити обсяг коду, що повинен пройти валідацію при проходженні процесу Code Review;
- підвищити читабельність отриманого коду, завдяки використанню геттерів та сеттерів, назви яких відповідають специфікації JavaBeans, та можливістю використання патерна Будівельник;
- підвищити стабільність отриманого коду завдяки тому, що інструмент автоматично зробить всі потрібні перевірки та перетворення даних, що будуть отримуватися з бази даних, або навпаки, зберігатися у ній;
- унеможливує завчасне використання ActiveJDBC об'єкту завдяки використанню патерна Будівельник;

ПЕРЕЛІК ПОСИЛАНЬ

1. ActiveJDBC / URL: <https://javalite.io/activejdbc>. дата звернення: 28.10.2023.

2. JavaBeans(TM) Specification 1.01 Final Release / URL: <https://download.oracle.com/otndocs/jcp/7224-javabeans-1.01-fr-spec-oth-JSpec/>. дата звернення: 28.10.2023
3. Spring dominates the Java ecosystem with 60% using it for their main applications / URL: <https://snyk.io/blog/spring-dominates-the-java-ecosystem-with-60-using-it-for-their-main-applications/>. дата звернення: 28.10.2023.
4. All About Annotations and Annotation Processors / URL: <https://medium.com/swlh/all-about-annotations-and-annotation-processors-4af47159f29d>. дата звернення: 28.10.2023.

УДК 004.422

В.В. Спирінцев¹, О.В. Хазова¹, Д.В. Садиченко¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСЛІДЖЕННЯ ТА ПОРІВНЯННЯ ЕФЕКТИВНОСТІ БАЛАНСУВАЛЬНИКІВ НАВАНТАЖЕННЯ

Анотація. Здійснено аналіз існуючих на сучасному ІТ-ринку балансувальників навантаження порівняння та аналіз різних балансувальників навантаження, що знаходяться на сучасному ринку інформаційних технологій. Отримані результати дозволять визначити переваги та недоліки кожного із розглянутих балансувальників навантаження, а також визначити оптимальні варіанти для різних сценаріїв використання.

Ключові слова: балансувальник навантаження, AWS, ELB, GCP, Azure, Google Cloud, Load Balancer, Load Balancing.

Вступ. Сучасне інформаційне середовище визначається високим рівнем складності та динамічності, що вимагає від компаній наявності ефективної та надійної інфраструктури для обробки та управління великим обсягом даних. Одним із ключових аспектів забезпечення стійкої та надійної роботи інформаційних систем є використання балансувальників навантаження [1-8]. Ці інструменти стають все більше важливими у контексті оптимізації роботи серверів, мереж та додатків, забезпечуючи розподіл та оптимізацію навантаження. З урахуванням стрімкого розвитку технологій та динамічного зростання обсягів обробки даних, актуальним стає вибір оптимального балансувальника навантаження для забезпечення ефективності та надійності інфраструктури. Розгляду даного питання присвячені проведені дослідження.

Постановка задачі. Балансування навантаження (Load Balancing) застосовується для оптимізації виконання паралельних обчислень за допомогою паралельної обчислювальної системи, яка передбачає рівномірне навантаження обчислювальних вузлів. В сфері створення високонавантажених додатків, що розгорнуті в хмарі гостро стоїть питання максимально ефективного

використання ресурсів. Дисбаланс навантаження може виникнути з кількох причин [1]:

- неоднорідність структури паралельного застосування (різні запити процесу вимагають різних обчислювальних потужностей);
- неоднорідність структури обчислювального комплексу (різні обчислювальні вузли володіють різною продуктивністю);
- неоднорідність структури взаємодії між вузлами обчислювальної мережі (лінії зв'язку між вузлами можуть мати різні характеристики пропускної спроможності).

На сьогоднішній день універсального методу боротьби з дисбалансом навантаження не існує. Метою даної роботи є узагальнення та систематизація результатів досліджень по розв'язанню поставленої задачі аналізу різних балансувальників навантаження в контексті їхньої здатності адаптуватися до різноманітних сценаріїв використання. Для досягнення поставленої мети в роботі необхідно:

- здійснити аналіз основних балансувальників навантаження, які пропонуються на ринку;
- дослідити технічні характеристики виявлених балансувальників, їх функціональні можливості та архітектурні особливості;
- здійснити порівняння різних балансувальників за певними критеріями. В якості критеріїв порівняння виступають: пропускна спроможність, час відгуку, масштабованість та можливості налаштувань.
- визначити, як кожен балансувальник навантаження пристосовується до різних типів завдань та обсягів навантаження;
- дослідити функціональні можливості балансувальників, включаючи розподіл трафіку, обробку виняткових ситуацій та можливості моніторингу;
- дослідити рівень надійності та стійкості кожного балансувальника під час специфічних умов експлуатації та атак.

Основний зміст роботи. У сучасному цифровому світі, де об'єми даних невинно зростають, ефективне управління та розподіл навантаження стає стратегічно важливим завданням для підприємств та організацій. Для цього використовуються балансувальники навантаження, які забезпечують оптимальний розподіл трафіку та забезпечують стабільну роботу інформаційних систем.

В цьому дослідженні ми спрямовуємо увагу на три ключові сервіси, які надають балансування навантаження та володіють значним авторитетом на ринку: Amazon Web Services (AWS) Elastic Load Balancer (ELB), Google Cloud Platform (GCP) Load Balancing та Microsoft Azure Load Balancer. Ці три платформи визначають стандарти в області хмарних технологій та пропонують рішення для забезпечення високої продуктивності та надійності інфраструктури.

Проведений аналіз [2-8], показав наступне. AWS Elastic Load Balancer є надійним сервісом, розподіляючи трафік між різними екземплярами EC2 для оптимальної продуктивності (рис.1).

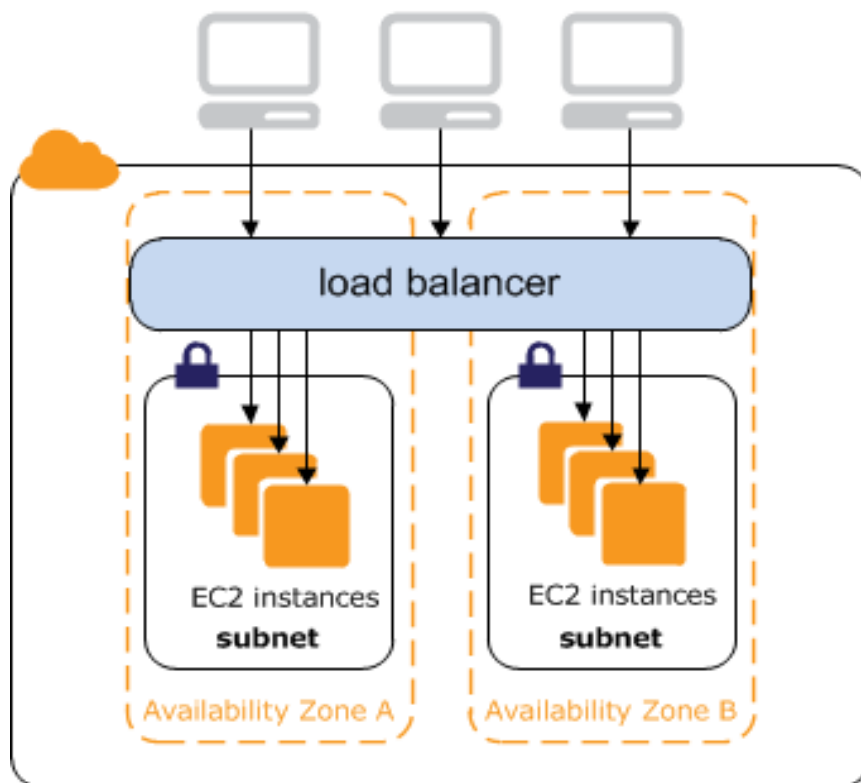


Рис.1. Принцип роботи AWS Elastic Load Balancer

Google Cloud Load Balancing забезпечує глобальне балансування навантаження для ефективного доставки контенту (рис. 2).

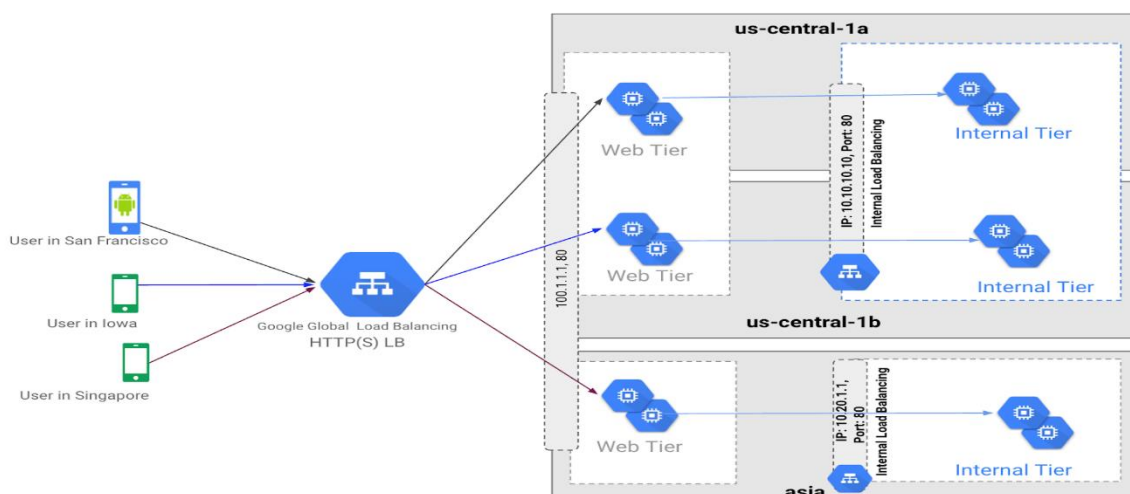


Рис.2. Принцип роботи Google Cloud Load Balancing

Тоді як Azure Load Balancer [7] організовує роботу з даними на рівні сеансів та використовується як для внутрішньої мережі, так і для зовнішнього трафіку (рис. 3).

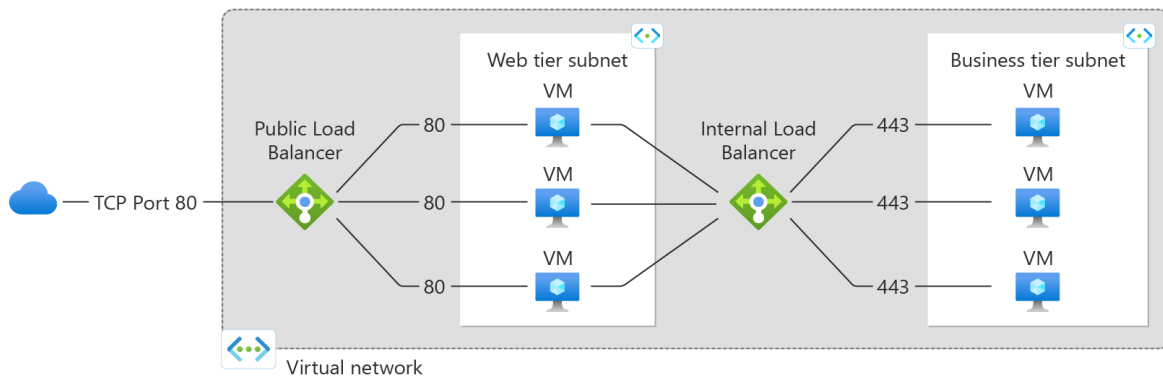


Рис. 3. Принцип роботи Azure Load Balancer

Порівняння технічних характеристик вказує на різницю у пропускній здатності, де AWS ELB має до 10 Гбіт/с, Google Cloud Load Balancing більше 1 Тбіт/с, а Azure Load Balancer до 1 Тбіт/с. Щодо часу відгуку, AWS ELB відзначається низьким часом завдяки оптимізації маршрутизації, Google Cloud Load Balancing має швидкий час відгуку за рахунок глобального балансування, а Azure Load Balancer демонструє надійний та ефективний час відгуку.

Щодо масштабованості: AWS ELB пропонує гнучку масштабованість з можливістю динамічного збільшення пропускнуої спроможності. Google Cloud Load Balancing автоматично масштабується відповідно до потреб, в той час як Azure Load Balancer здатний обробляти зростаючий обсяг трафіку шляхом додавання ресурсів.

Аналіз функціональності розкриває, що всі три балансувальники мають розподіл трафіку, проте AWS ELB робить це на різні зони доступності та екземпляри EC2 (див. рис.1), Google Cloud Load Balancing забезпечує глобальне балансування з розподілом трафіку за регіонами (див. рис. 2), а Azure Load Balancer розподіляє трафік між віртуальними машинами та ресурсами (див. рис. 3). У контексті обробки виняткових ситуацій, AWS ELB володіє вбудованою підтримкою відновлення та перерозподілу трафіку під час збоїв, Google Cloud Load Balancing може виявляти та управляти ситуаціями недоступності, а Azure Load Balancer надає можливості моніторингу та автоматичного виправлення помилок.

Один з ключових аспектів вибору балансувальника навантаження - це його рівень надійності та стійкість до різних умов експлуатації та потенційних атак. Оцінимо ці параметри для кожного з розглянутих сервісів:

- AWS Elastic Load Balancer (ELB) - володіє високим рівнем надійності завдяки автоматичному розподілу трафіку та можливості відновлення під час збоїв. AWS забезпечує захист від різних типів атак, використовуючи сервіси WAF (Web Application Firewall) та AWS Shield для виявлення та мітигації потенційних загроз

- Google Cloud Platform (GCP) Load Balancing - є високонадійним сервісом з можливістю автоматичного масштабування та відновлення під час збоїв. GCP пропонує інструменти, такі як Cloud Armor, для виявлення та захисту від DDoS-атак та інших загроз в мережі.

- Microsoft Azure Load Balancer - забезпечує стабільну роботу та можливості автоматичного відновлення. Azure має вбудовані інструменти для виявлення та мітігації DDoS-атак, а також можливості моніторингу та автоматичного виправлення помилок.

Наукова новизна складається у запропонованому підході щодо вибору найоптимальнішого балансувальника навантаження з урахуванням специфічних вимог та характеристик розробляемого застосунку, що стає все більш важливим у контексті оптимізації роботи серверів, мереж та додатків, забезпечуючи ефективний розподіл та оптимізацію навантаження.

Висновки: Результати ретельного порівняльного аналізу балансувальників навантаження AWS, Google Cloud Platform та Azure вказують на ключові особливості, які слід враховувати при виборі оптимального рішення для конкретних потреб і програм.

AWS Elastic Load Balancer (ELB) оптимально використовувати для масштабних веб-застосунків з великою кількістю користувачів, оскільки AWS ELB володіє гнучкою масштабованістю та високою пропускнуою здатністю. Google Cloud Platform (GCP) Load Balancing відзначається високою пропускнуою здатністю та глобальним балансуванням, роблячи його ефективним для застосунків, що вимагають швидкої та надійної доставки контенту на всіх континентах. Microsoft Azure Load Balancer ефективно організовує роботу з даними на рівні сеансів, що робить його популярним для внутрішньої мережі та застосунків, які використовують сеансовий рівень комунікації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Load Balancing in Parallel Computers. [Electronic resource]. – Access mode: <http://www.inspireignite.com/load-balancing-in-parallel-computers/> – Title from the screen.
2. How Elastic Load Balancing works [Electronic resource]. – Access mode: <https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html> – Title from the screen.
3. Elastic Load Balancing features [Electronic resource]. – Access mode: <https://aws.amazon.com/elasticloadbalancing/features/> – Title from the screen.
4. Cloud Load Balancing overview [Electronic resource]. – Access mode: <https://cloud.google.com/load-balancing/docs/load-balancing-overview> – Title from the screen.
5. Load balancer feature comparison [Electronic resource]. – Access mode: <https://cloud.google.com/load-balancing/docs/features> - Title from the screen.
6. Internal Load Balancer [Electronic resource]. – Access mode: <https://www.cloudskillsboost.google/focuses/1910?parent=catalog> – Title from the screen.
7. What is Azure Load Balancer? [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/azure/load-balancer/load-balancer-overview> – Title from the screen.
8. Networking services compared: AWS vs Azure vs Google Cloud [Electronic resource]. – Access mode: <https://www.pluralsight.com/resources/blog/cloud/networking-services-compared-aws-vs-azure-vs-google-cloud> – Title from the screen.

В.М. Куваєв¹, Л.І. Мещеряков¹, А.Т. Харь¹, А.І. Політов²

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Дніпровський Національний університет імені Олеся Гончара, Дніпро, Україна

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ ПРИ ПІДВИЩЕНИХ ВИМОГАХ ДО РЕАКЦІЇ ОПЕРАТОРА НА ПОЗАШТАТНІ СИТУАЦІЇ

Анотація. Описані особливості підходів до реалізації людино-машинного інтерфейсу при підвищених вимогах до реакції оператора на позаштатні ситуації, що виникають при протіканні технологічного процесу, на прикладі системи керування швидкісним режимом безперервної прокатки.

Ключові слова. інформаційно-керуючі системи, інтерфейс оператора, позаштатні ситуації.

Вступ. На поточний час найбільш широке застосування при розробці людино-машинних інтерфейсів систем технологічної автоматизації знайшли програмні засоби SCADA [1,2,3]. Таки системи надають широкий спектр інструментарію для програмування інтерфейсу операторів, який є, за суттю, віртуальним відображенням пультів операторів і щитів диспетчерського керування. Але їх застосування передбачає строгу ієрархічну структуру апаратних і програмних засобів АСК ТП – SCADA системи використовуються на рівні станцій оператора і призначені для підтримки режимів «м'якого» реального часу. Цей рівень керування не передбачає реалізації, хоча б в обмеженому обсязі, задач критичних до режиму реального часу. Одночасно рівень контролерів передбачає взаємодію з оператором через виключно через інтерфейс верхнього рівня, що уповільнює реакцію АСК ТП на лії оператора. У той же час виникнення позаштатних ситуацій при протіканні технологічних процесів часто потребує більш швидку реакцію, ніж це забезпечує стандартний підхід до організації людино-машинного інтерфейсу.

Постановка задачі. Виходячи з викладеній у вступі проблемі виникає задача у пошуку методів підвищення швидкості реакції АСК ТП на позаштатні ситуації усунення яких здійснюється оператором технологічного процесу.

Основний зміст роботи. Особливістю систем що керують технологічним процесом з жорсткими часовими рамками по втручанні оператора до їх усунення є те, що, як правило, він аналіз якості настроювання технологічного процесу здійснює візуально контролюючи технологічний процес і, в більший мірі, використовуючи інформацію, яка відображається на дисплеї. Взаємодія з останнім здійснюється через інтерфейс оператора, що надає можливість доступу до всього спектру інформації, що наявна в АСК ТП. Це диктує свої підходи до формування інтерфейсу оператора технологічного процесу в частині структури і щільності розміщення інформації на дисплеї і вирішення задачі створення ергономічного людино-орієнтовного інтерфейсу. До таких задач відносяться,

перше за все, забезпечення швидкого доступу до потрібної інформації і надання інформації у найбільш зручному вигляді для різних режимів настроювання і ведення технологічного процесу.

Як приклад вирішення таких задач розглянемо процес розробки такого інтерфейсу оператора на прикладі системи керування швидкісним режимом прокатки дрібносоротно-дротового стана ДСДС 250/150 ПАТ «АрселорМіттал Кривий Ріг». З функціональною структурою і основними програмними рішеннями цієї системи можна ознайомитися в [4].

Перше за все зазначимо, що процес керування швидкісним режимом прокатки здійснюється оператором комбіновано: як з через інтерфейс оператора верхнього рівня так і через органи керування, що розташовані на пульті оператора і підключені безпосередньо до контролерів. Це пов'язано з критичним часом реакції оператора на низку технологічних ситуацій при яких більш оперативне втручання оператора в хід технологічного процесу через органи керування, що розташовані на пульті, дозволяє запобігти його порушення або знизити негативні наслідки такого порушення. Практика показала доцільність і ефективність такого рішення. З цього можна зробити висновок, що при розробці інтерфейсу оператора-технолога інформаційно-керуючих систем слід враховувати вимоги реального часу не тільки для функцій, що виконуються в автоматичному режиму «жорсткого» реального часу, а і автоматизованих функцій які відносяться до функцій «м'якого» реального часу. Така ситуація характерна для технологічних процесів з високою швидкістю їх протікання. Наприклад, швидкість прокатки на стані ДСДС 250/150 досягає 18м/с.

Загальна структура інтерфейсу, що розміщено на дисплеї орієнтована на мінімізацію часу доступу оператора до потрібної інформації при керуванні технологічним процесом в режимі реального часу. і наведена на рис.1.

Загальний простір екрану розбито на зони, функціонал яких пов'язаний з найбільш зручним зосередженням уваги оператора на тієї чи іншої ділянці екрану. Найбільш актуальна інформація відображається у центральній області екрану і ліворуч неї, що відповідає природній звички читання і перегляду малюнків, меню розташована в зоні заголовків, справа розташована загальна інформація про робочу систему, поточний профіль та загальні параметри виробництва. У нижньому рядку представлена додаткова інформація.

Головне меню містить усі інструменти керування системою, а також дублює всі закладки блокнотів. Меню кнопок призначене для швидкого виклику тих чи інших функцій системи – функції кнопок відповідають функціям аналогічних пунктів головного меню.

Поле індикації режимів і додаткової інформації включає п'ять панелей для відображення загальносистемних параметрів, таких як: тип системи, режим управління клітями, профіль, що прокочується та ін.

Рядок стану містить такі поля: індикація порушення технологічного режиму, кнопка швидкого переходу до перегляду журналу порушень, поточний час, категорія користувача, зареєстрованого в системі, номер вузла.

Сторінки блокнотів містять зведені таблиці базових (еталонних), змінних та контрольованих параметрів клітей і механізмів, згрупованих відповідно до їх

індивідуальних властивостей та призначення. При проектуванні системи відмовились від представлення інформації в вигляді мнемосхем технологічного процесу на користь більш щільного розміщення інформації та мінімізації сторінок блокноту. Це дає певне скорочення кількості перемикачів між сторінками блокноту при настроюванні, аналізі та корекції швидкісного режиму прокатки оператором.

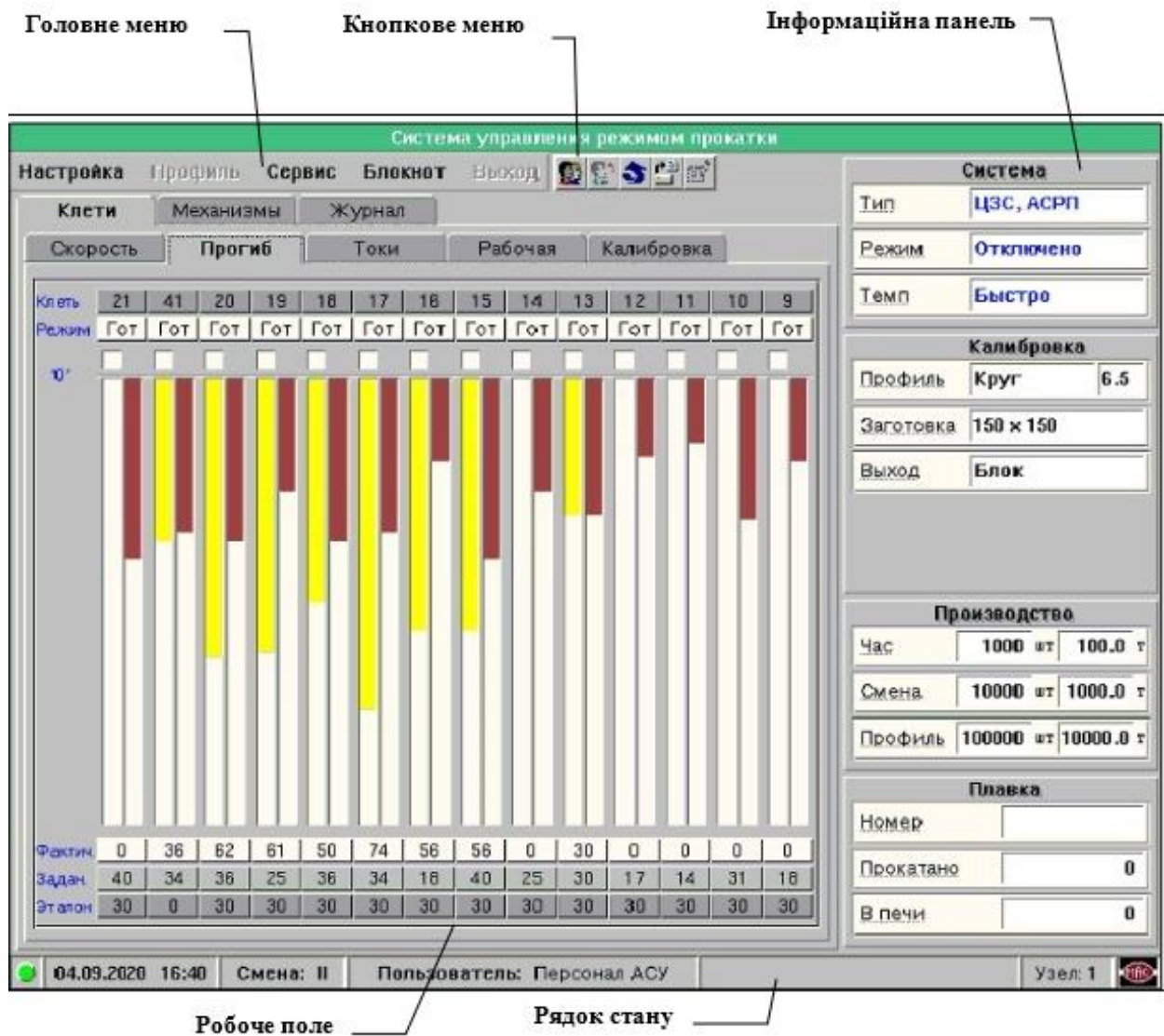


Рис. 1. Загальна структура інтерфейсу, що розміщено на дисплеї.

Відеограми у яких основними інформаційними елементами є стовпчикові діаграми, теж формуються за табличним принципом (див. рис.1)

Таке подання інформації дозволяє значно легше аналізувати якість налаштування швидкісного режиму прокатки для окремих міжклетьових проміжків та і контролювати прокатку заготовок у лінії клітей в цілому.

Запропонований комплекс апаратно-програмних рішень дозволив підвищити точність налаштування швидкісного режиму безперервної прокатки і забезпечити виробництво прокату з підвищеною точністю геометричних розмірів його перетину, скоротити час налаштування швидкісного режиму прокатки до прокатки першої

заготовки та скоротити число випадків переростання позаштатних ситуацій в аварійні, що дозволило зменшити непродуктивні втрати металу.

Наукова новизна полягає у обґрунтуванні рішень по організації людино-машинного інтерфейсу що скорочують час реакції АСК ТП на позаштатні ситуації.

Висновки. При розробці інтерфейсу оператора-технолога інформаційно-керуючих систем слід враховувати вимоги реального часу не тільки для функцій, що виконуються в автоматичному режиму «жорсткого» реального часу, а і автоматизованих функцій які відносяться до функцій «м'якого» реального часу. Керування оператором технологічним процесом з високою швидкістю протікання доцільно здійснювати комбіновано: як з через інтерфейс оператора так і через органи керування пульта оператора

Організація інтерфейсу оператора у вигляді дворівневого блокнота дозволяє організувати швидкий доступ до актуальної інформації, а таблична організація даних на сторінках блокноту – зробити їх інформаційно насиченими.

Досвід впровадження і експлуатації системи керування швидкісним режимом прокатки підтвердив ефективність запропонованих рішень організації інтерфейсу оператора.

ПЕРЕЛІК ПОСИЛАНЬ

1. Пупена О.М. Розроблення людино-машинних інтерфейсів та систем збирання даних з використанням програмних засобів SCADA/HMI: Навч. посіб. Київ : Видавництво Ліра-К, 2020. — 594 с.
2. Bailey D., Wright E. Practical SCADA for Industry. - Newnes, 2005. – 304 p.
3. Автоматизована система диспетчерського керування WindEx. URL: https://activolt.com.ua/wp-content/uploads/2021/04/windex-.2_2021.pdf (дата звернення 24.04.2023).
4. Kuvaiev, V., Ishchuk, P., Politov, A., Buriak, V. (2021). Prohramni rishennia po zabezpechenniu nadiinoho funktsionuvannia skladnykh informatsiino-keruiuchykh system krytychnykh do rezhymu realnoho chasu [Software solutions to ensure the reliable operation of complex information and control systems critical to real-time]. Information Technology: Computer Science, Software Engineering and Cyber Security, 1, 16–24. [in Ukrainian]. doi: <https://doi.org/10.32782/IT/2021-1-3>

ЗАБЕЗПЕЧЕННЯ ЕФЕКТИВНОГО МЕНЕДЖМЕНТУ СТАНУ ДОДАТКУ: РОЛЬ STATE MANAGEMENT В РОЗВИТКУ МОБІЛЬНИХ ДОДАТКІВ FLUTTER

Анотація. Розглянуто ключові аспекти розробки мобільних додатків за допомогою технологій управління станом. У сфері розвитку мобільних додатків Flutter важливу роль відіграє state management для забезпечення ефективного управління станом додатків. Підкреслюється значення використання state management для оптимізації роботи зі станом додатків у зв'язку з їх взаємодією з різними компонентами та елементами. Проаналізовано різноманітні підходи до state management у мобільних додатках Flutter. В основі дослідження покладено вивчення сучасних методів та технологій управління станом (Provider, Bloc, MobX) та їх вплив на розвиток додатків. Окрема увага приділяється аналізу викликів, з якими стикаються розробники при виборі підходу до управління станом, а також ефективності цих підходів у контексті розробки мобільних додатків. Результати дослідження роблять акцент на важливості правильного вибору та застосування підходу до управління станом для забезпечення оптимальної продуктивності та швидкості реакції додатку.

Ключові слова: *Flutter, State Management, Provider, Bloc, MobX, управління станом.*

Вступ. Зараз спостерігаємо ситуації, як мобільні додатки Flutter відкривають нові можливості для створення надзвичайно функціональних та захоплюючих застосувань. Проте, успіх цих додатків вимагає більш глибокого розуміння ролі, яку відіграє state management у їхньому розвитку. Дані дослідження спрямовані на розкриття ключового значення state management в забезпеченні ефективного управління станом мобільних додатків у середовищі Flutter. Розглядаючи цей аспект розвитку, досліджуються різні підходи до управління станом, визначаються їхні переваги і вплив на якість та продуктивність додатків. Велика кількість мобільних додатків розроблених за допомогою Flutter змушує розробників поглибитись в технічні деталі, які лежать в основі їхнього успіху. І саме state management виступає тим ключовим елементом, який визначає зручність, швидкість та функціональність додатків. Актуальним питанням є не лише з'ясувати, як state management впливає на розвиток додатків, але й визначити кращі практики для побудови ефективних та легко масштабованих мобільних рішень у середовищі Flutter.

Постановка задачі. Мета цього дослідження полягає у створенні оптимальної стратегії керування станом мобільних додатків, розроблених за допомогою Flutter. Це включає в себе: аналіз існуючих методів управління станом, визначення найбільш ефективного підходу для конкретного типу додатка, реалізацію та тестування обраної стратегії, а також аналіз результатів для подальших покращень. Для реалізації поставленої мети в роботі необхідно здійснити аналіз різних методів керування станом, порівняти їхню ефективність та простоту використання, а також обрати найбільш ефективний метод для

оптимізації роботи додатків. Основні завдання включають розробку та впровадження обраної стратегії управління станом в мобільних додатках, їхнє тестування для перевірки покращень у продуктивності та зручності використання, а також аналіз результатів для виявлення можливостей подальшого удосконалення.

Основний зміст роботи. Flutter - це кросплатформовий набір інструментів інтерфейсу користувача, який призначений для повторного використання коду в операційних системах, таких як iOS і Android, а також дозволяє додаткам безпосередньо взаємодіяти з базовими сервісами платформи (рис. 1). Ціль полягає в тому, щоб надати розробникам можливість створювати високопродуктивні додатки, які виглядають природно на різних платформах, враховуючи відмінності там, де вони існують, і одночасно спільно використовуючи якомога більше коду.

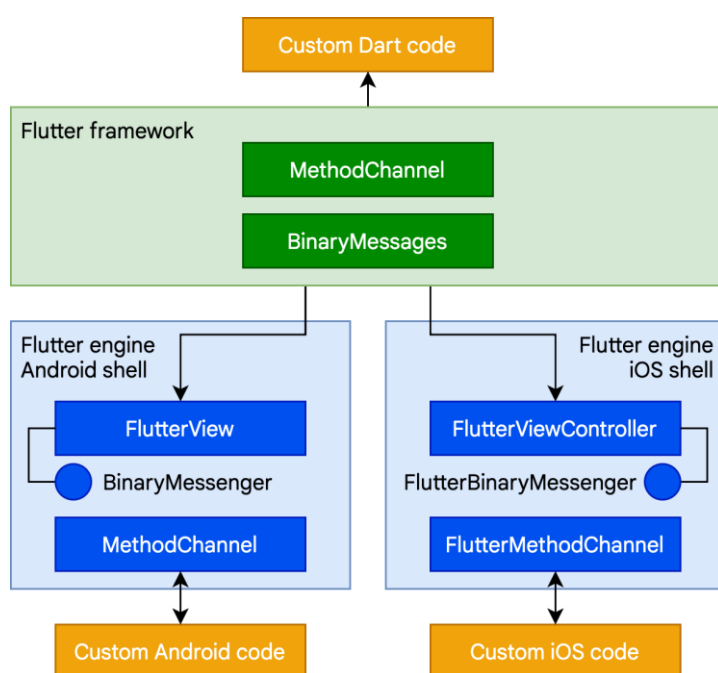


Рис.1. Архітектура Flutter

Наразі у Flutter існує кілька популярних бібліотек для управління станом (рис. 2).

Розглянемо більш ретельно деякі із них:

- **Provider.** Базується на концепції "розміщення". Це легкий у використанні і досить розширюваний інструмент, який дозволяє зручно передавати дані по всьому додатку без необхідності великої кількості коду.

- **Bloc** (реалізовано через пакет `flutter_bloc`). Основна ідея полягає в розділенні бізнес-логіки від UI. Це забезпечує чіткість коду, оскільки стан додатка управляється за допомогою різних блоків.

- **GetX.** Бібліотека пропонує все-в-одному рішенні для стану, маршрутизації та інших потреб додатку. Вона надає простий синтаксис та швидкість роботи.
- **Riverpod.** Це рішення, яке дозволяє працювати зі станом у вигляді провайдерів, що надає більш гнучкий підхід до управління станом.
- **Redux.** Це реалізація патерна Redux у Flutter. Вона пропонує централізований стан за допомогою механізму диспетчеризації дій (actions) та змінювачів (reducers).

➤ Available State Management

Name	# Popularity	# Likes	# Pub Point	Version	
Provider	100%	4767	130	5.0.0	
GetX	99%	5603	120	4.3.4	
BLoC / Rx (Flutter Bloc)	99%	2567	130	7.1.0	
flutter_hooks**	98%	735	130	0.18.0	
Riverpod (flutter_riverpod)	98%	490	130	0.14.0+3	
MobX (flutter_mobx)	98%	372	130	2.0.1	
Stacked**	97%	785	120	2.2.3	
Redux (flutter_redux)	97%	290	120	0.8.2	
states_rebuilder	96%	314	120	4.3.0	

Рис. 2. Порівняння state management бібліотек

Ці бібліотеки мають власні переваги та підходять для різних типів додатків та сценаріїв розробки. Обираючи одну з них, важливо враховувати специфіку проекту та особливості розробки, щоб забезпечити оптимальну роботу зі станом у вашому додатку.

Розглянемо основні переваги та недоліки найпопулярніших state management бібліотек (Provider, Bloc, GetX).

Provider.

Переваги:

- Простота використання: має простий API, що дозволяє легко передавати та оновлювати дані у додатку.
- Гнучкість: забезпечує різні методи передачі стану, такі як Provider, ChangeNotifierProvider, та багато інших, що робить його гнучким для різних потреб.
- Добре поєднується з Flutter: розроблено спеціально для Flutter і має хорошу інтеграцію з фреймворком.

Недоліки:

- Може стати складним для великих проектів: у деяких випадках, коли додаток має великий обсяг стану, може виникнути необхідність управління багатьма провайдерами, що може зробити код складнішим.

Використання:

- Добре підходить для невеликих та середніх за розміром додатків, де простота та гнучкість важливі.

Bloc.

Переваги:

- Розподіл бізнес-логіки: забезпечує чіткий розподіл бізнес-логіки та UI, що спрощує розробку та тестування.

- Централізований стан: можливість централізувати стан додатку робить код більш передбачуваним та легшим для розуміння.

- Розширюваність: має широкі можливості для розширення за допомогою подій, станів та трансформацій.

Недоліки:

- Вимагає додаткового вивчення: може вимагати більшого часу для засвоєння, особливо для початківців.

Використання:

- Ідеальний для складних додатків, де потрібний чіткий розподіл між бізнес-логікою та UI, а також коли важлива розширюваність коду.

GetX.

Переваги:

- Швидкість та простота: має простий синтаксис та дозволяє швидко розробляти додатки без великої кількості коду.

- Все-в-одному: надає все необхідне для розробки, включаючи стан, маршрутизацію та багато іншого.

Недоліки:

- Може бути менш гнучким: для деяких проектів може бути менш гнучким порівняно з іншими бібліотеками.

Використання:

- Підходить для швидкого прототипування, невеликих та середніх за розміром додатків, де простота є ключовою.

Наразі бібліотека Bloc (з використанням пакету `flutter_bloc`) залишається однією з найпопулярніших та найбільш потужних бібліотек для управління станом у Flutter додатках. На рис. 3 наведено компоненти бізнес логіки.

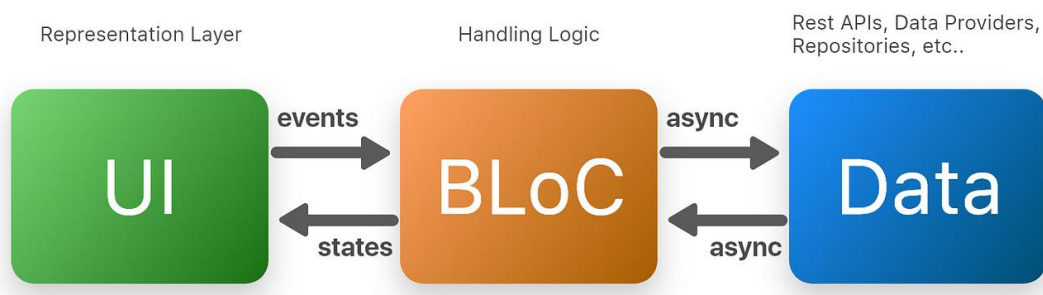


Рис. 3. Компоненти бізнес логіки

Ось деякі причини, чому вона може залишатися першим вибором для розробників у 2023 році. Bloc сприяє чіткому розподілу між логікою додатку та його представленням, що дозволяє підтримувати код більш структурованим та легким для розуміння. Централізований підхід до керування станом дозволяє зробити управління даними більш передбачуваним та контрольованим. Bloc має широкі можливості для розширення через події, стани та трансформації, що дозволяє гнучко пристосовувати логіку додатку під конкретні потреби. Ця бібліотека активно розвивається та має активне співтовариство розробників, що забезпечує підтримку, оновлення та вдосконалення. Bloc вже зарекомендував себе як стабільна та надійна бібліотека, яка допомагає уникнути багатьох проблем з управлінням станом. Багато навчальних матеріалів, документаційних ресурсів та підтримки спрямовано на використання Bloc, що полегшує його вивчення та розвиток для розробників.

Звісно, кожна бібліотека має свої власні переваги та контекст застосування. Але Bloc залишається важливим інструментом для розробників Flutter, які вважають чіткість, розширюваність та контроль за станом додатку ключовими аспектами розробки.

Наукова новизна. Новизна досліджень полягає у комплексному рішенні щодо огляду різних підходів до управління станом, визначенні їх переваг та недоліків у зв'язку з використанням Flutter для мобільних розробок. Дані дослідження надають комплексний аналіз методів управління станом з фокусом на платформі Flutter, що дозволяє розробникам зрозуміти, як вибір певного підходу до управління станом може впливати на якість, продуктивність та масштабованість мобільних додатків на даній платформі.

Висновки. В даній роботі було досліджено різні підходи до управління станом мобільних додатків Flutter, визначаючи їх переваги та вплив на продуктивність застосунків. Виходячи з цього дослідження, стає очевидним, що вибір оптимального методу управління станом має вирішальне значення для досягнення успішного та ефективного розвитку додатків. Ефективний state management дозволяє покращити якість коду, забезпечити швидкість реакції додатку та сприяє його масштабованості.

ПЕРЕЛІК ПОСИЛАНЬ

1. Flutter Architecture Overview. [Electronic resource]. – Access mode: <https://docs.flutter.dev/resources/architectural-overview>– Title from the screen.
2. Top Flutter State Management Libraries in 2023. Canadian Software Agency Inc. article. LinkedIn. – Access mode: https://ca.linkedin.com/company/canadian-software-agency?trk=article-ssr-frontend-pulse_publisher-author-card– Title from the screen.
3. List of state management approaches. Flutter Docs. [Electronic resource]. – Access mode: <https://docs.flutter.dev/data-and-backend/state-mgmt/options>– Title from the screen.
4. Comparative study on Flutter State Management. [Electronic resource]. – Access mode: <https://blog.alifakbar.com/post/658377568699957248/comparative-study-on-flutter-state-management>– Title from the screen.

С.Д. Приходченко¹, Н.Л. Гресь¹, А.Г. Шпорта¹, А.Є. Тріско¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСЛІДЖЕННЯ СПОСОБІВ ТА ПРИНЦИПІВ ВЗАЄМОДІЇ З ARTIFICIAL REALITY НА ПРИКЛАДІ ІГРОВОГО ДОДАТКУ

Анотація. В роботі описано аналіз AR технологій та принципи взаємодії з ними для використання в ігровому додатку. В результаті розроблено продукт, що являє собою ігровий застосунок з AR елементами.

Ключові слова: *Artificial Reality, ігровий додаток, ігровий двигун, користувач, Unity, Vuforia.*

Вступ. У сучасному світі швидкого технологічного розвитку взаємодія з штучною реальністю (Artificial Reality, AR) стає невід'ємною частиною нашого цифрового життя. Сфери застосування AR дедалі розширюються, і ця технологія проникає все далі. Від віртуальних вітрин та візуалізаторів в інтернет-магазинах до допомоги нашим медичним службам за допомогою технологій, таких як AssuVein, можливості, які надає доповнена реальність, виходять далеко за межі очікувань піонерів цієї технології.

Однією зі сфер, яка активно впроваджує принципи AR, є розробка ігрових додатків. Високий рівень інтерактивності та можливість взаємодії з віртуальним середовищем роблять AR ігри захоплюючими та зацікавлюючими для користувачів.

Дослідження способів та принципів взаємодії з AR на прикладі ігрового додатку стає важливим етапом у вдосконаленні цих технологій. В контексті ігор AR, ключовою є розробка ефективних методів взаємодії гравця з віртуальним світом, що дозволяє створювати неперевершені геймплей-враження.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- Проаналізувати існуючі принципи взаємодії з AR;
- визначити найбільш підходящі для використання в ігровому додатку;
- спроектувати взаємодію ПЗ та доданої реальності для найбільш гармонійного поєднання;
- спроектувати та розробити ігровий додаток, що буде використовувати Augmented Reality для вдосконалення;
- зробити висновки щодо доцільності використання AR в ігрових додатках.

Основний зміст роботи. Доповнена реальність (AR) є одним із провідних іммерсивних досвідів XXI століття. AR викликала революцію в різних сферах, включаючи сфери охорони здоров'я і медицини, навчання та освіти, туризм, дизайн, виробництво та інші схожі галузі.

Доповнена реальність надає користувачам комбінований вид, накладаючи комп'ютерно-генерований віртуальний контент, такий як звук, графіка, текст або відео, на об'єкти реального світу. Основні компоненти процесу в AR - відстеження положення для розміщення віртуальних об'єктів у реальному середовищі та відображення віртуального контенту користувачеві.

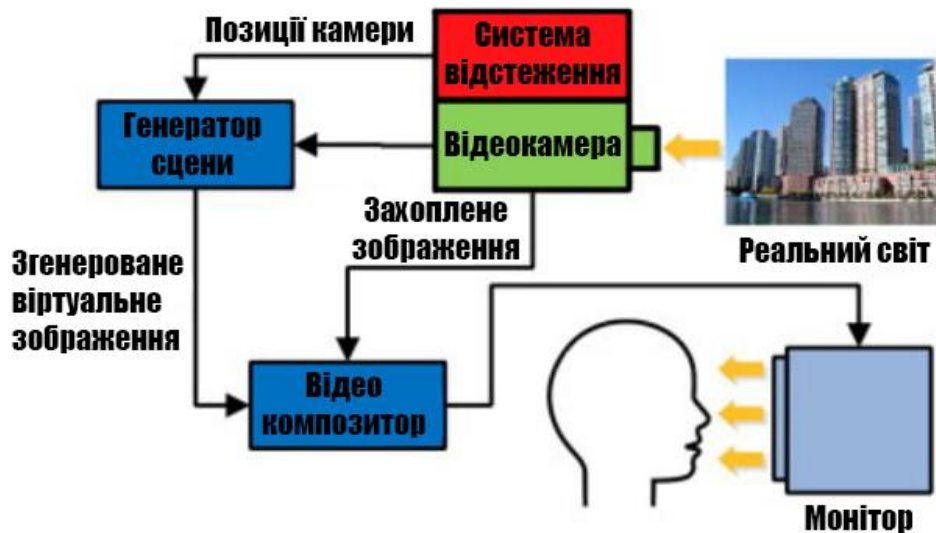


Рис. 1. Процес накладання доповненої реальності

Технології відстеження вводять в світ розширеної реальності відчуття руху, що можна використовувати для досягнення різноманітних цілей та виконання завдань. Якщо система відстеження правильно вибрана та належно встановлена, вона дозволяє особі переміщатися віртуальним та розширеним середовищем. Крім того, це надає можливість взаємодіяти з людьми та об'єктами у розширених середовищах. Вибір технології відстеження залежить від типу середовища, типу даних і доступності необхідних коштів.

Для розуміння чому були імплементовані конкретні технології слід розглянути їх основні типи, призначення, переваги та недоліки.

Технологія магнітного відстеження: Ця технологія включає в себе джерело відстеження та два сенсори: один для голови та інший для руки. Джерело відстеження створює електромагнітне поле, в якому розташовані сенсори. Потім комп'ютер розраховує орієнтацію та положення сенсорів на основі затухання сигналу у полі. Це надає можливість повного діапазону рухів у 360 градусів, тобто дозволяє нам дивитися та рухатися у всіх трьох осях ординат[1]. Однією з основних проблем магнітного відстеження є обмежений діапазон розпізнавання.

Відстеження за допомогою інерційних сенсорів: Магнітометри, акселерометри та гіроскопи є прикладами інерційних вимірювальних блоків (ІВБ), які використовуються в інерційному відстеженні для оцінки швидкості та орієнтації відстежуваного об'єкта. Інерційна система відстеження використовується для визначення трьох ротаційних ступенів свободи відносно гравітації.

Переваги інерційного відстеження: воно не потребує прямого видимості та не має обмежень дальності. Воно не схильне до оптичних, акустичних, магнітних та електромагнітних джерел перешкод. Крім того, воно має незначний час затримки і може бути оброблене так швидко, як це потрібно.

Недоліки інерційного відстеження: воно схильне до дрейфу орієнтації та положення з часом, основний вплив має на вимірювання положення[2].

Візійне відстеження визначається як підходи до відстеження, які встановлюють положення камери за допомогою даних, отриманих з оптичних сенсорів. Основною перевагою цієї технології є її доступність завдяки покращеній обчислювальній потужності споживчих пристроїв та всеосяжності мобільних пристроїв. Недоліком є низька чіткість відстеження близьких об'єктів та залежність від технічних характеристик пристрою користувача.

Відстеження тривимірної структури: Для визначення тривимірних точок у сцені можуть використовуватися різні типи сенсорів. Найбільш поширеними є структуроване світло або принцип часу польоту[3]. Ці технології працюють на принципі аналізу глибини. У цьому випадку глибинна інформація про реальне оточення витягається за допомогою розмітки та відстеження. Інформація про тривимірну структуру стала дуже доступною завдяки розвитку комерційних сенсорів, здатних виконувати цю задачу.

Інфрачервоне відстеження: Об'єкти, які випромінюють або відбивають світло, є одними з найраніших методів відстеження, які використовуються в технологіях розширеної реальності. Їх висока яскравість порівняно з навколишнім середовищем робить це дуже простим.[4] Самостійно випромінюючі мішені також були байдужі до навколишнього середовища, наприклад, тіні або поганого освітлення. Відстеження на основі моделі: Раніше відстеження тривимірної моделі об'єкта зазвичай створювалося вручну. У цій системі лінії, циліндри, сфери, кола та інші примітиви об'єднувалися для ідентифікації структури об'єктів[5]. Структурна інформація сцени вилучалася за допомогою крайових фільтрів. Крім того, для визначення позиції порівнювалися інформація про краї та примітиви. Подібний підхід дозволяє досягти стабільного відстеження об'єктів навіть у відкритих середовищах. Мінусами є необхідність розробки алгоритмів стеження та моделей об'єктів.

GPS-відстеження: Ця технологія відноситься до позиціонування за допомогою прив'язки до реальної позиції на Землі. Поточна точність системи GPS становить до 3 метрів, однак можливі покращення завдяки прогресу у супутниковій технології та кількох інших розробках. Один із прикладів - це реальний кінематичний метод (RTS), який використовує несучу хвилю сигналу GPS. Основна перевага його полягає в тому, що він може підвищити точність до рівня сантиметрів. Військові, ігрові[6] та додатки для проведення екскурсій використовують GPS-відстеження для досвіду в доповненій реальності. Оскільки воно має низьку точність відстеження позиції, його можна використовувати тільки у гібридних системах відстеження або в застосунках, де не важлива точна реєстрація позиції.

Гібридне відстеження: Гібридні системи були розроблені для досягнення наступних цілей: покращення точності відстеження, усунення вад задіяних систем, додавання більшої свободи руху та огляду. Наприклад, поєднання GPS відстеження з візійним дозволяє нівелювати неточність позиціонування першої.[7] Мінусом такого підходу є необхідність задіювати декілька технологій в межах однієї системи, що є серйозним викликом.

Fiducial Tracking: Штучні орієнтири, які додаються до середовища для допомоги в відстеженні та реєстрації, відомі як фідуціали. Складність відстеження фідуціалів значно варіюється в залежності від використовуваної технології та застосування. З розвитком робіт над концепцією та складністю фідуціалів були введені додаткові можливості, такі як багаторядкове розпізнавання фідуціалів на значно більших відстанях. Для визначення положення спостерігача потрібно мінімум чотири точки з відомою позицією.[8]

Для розробки ігровго додатку було обрано фідуційний та візійні підходи як такі, що є найбільш доступними та зручними для сумісного використання, адже саме завдяки цьому тандему можна отримати найбільш оптимальний результат в рамках наявних ресурсів.

Наукова новизна полягає в дослідженні та порівнянні методів взаємодії з Artificial Reality, що дозволить імплементувати їх в ігровий додаток та надати користувачам більш іммерсивні відчуття від гри, роблячи її більш захоплюючою.

Висновки. В результаті було проаналізовано існуючі технології та принципи взаємодії з AR, описані їх переваги та недоліки. На основі отриманої інформації було розроблено ігровий додаток на базі ігрового двигуна Unity з використанням AR SDK Vuforia, що дозволило перевірити теоретичні міркування на практиці та підтвердити їх доречність.

ПЕРЕЛІК ПОСИЛАНЬ

1. Santoni, F.; De Angelis, A.; Moschitta, A.; Carbone, P. MagIK: A Hand-Tracking Magnetic Positioning System Based on a Kinematic Model of the Hand. *IEEE Trans. Instrum. Meas.* 2021, 70, 1–13.
2. Heidemann, G.; Bax, I.; Bekel, H. Multimodal interaction in an augmented reality scenario. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, State College, PA, USA, 13–15 October 2004; pp. 53–60.
3. Wang, X.; Dunston, P.S. Comparative effectiveness of mixed reality-based virtual environments in collaborative design. *IEEE Trans. Syst. Man Cybern. Part C* 2011, 41, 284–296.
4. Danielsson, O.; Holm, M.; Syberfeldt, A. Augmented reality smart glasses for operators in production: Survey of relevant categories for supporting operators. *Procedia CIRP* 2020, 93, 1298–1303.
5. Gibson, L.; Hanson, V.L. Digital motherhood: How does technology help new mothers? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Paris, France, 27 April–2 May 2013; pp. 313–322.
6. Henrysson, A.; Ollila, M. UMAR: Ubiquitous mobile augmented reality. In *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia*, College Park, MD, USA, 27–29 October 2004; pp. 41–45.
7. Höllerer, T.; Wither, J.; DiVerdi, S. “Anywhere augmentation”: Towards mobile augmented reality in unprepared environments. In *Location Based Services and TeleCartography*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 393–416.

8. Evennou, F.; Marx, F. Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. EURASIP J. Adv. Signal Process. 2006, 2006, 1–11.

УДК 004.415.3:681.6

О.В. Спирінцева¹, І.Д. Різниченко¹

¹Дніпровський Національний університет імені Олеся Гончара, Дніпро, Україна

РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ КЕРУВАННЯ ТЕХНОЛОГІЧНИМИ ПРОЦЕСАМИ З ВИКОРИСТАННЯМ СПЕЦИФІКАЦІЇ OPC UA

Анотація. В роботі розроблено та досліджено веб-додаток, який використовує комунікацію OPC UA та має функціонал, що дозволяє користувачеві контролювати, відстежувати та аналізувати виробничий процес. Розробка таких специфікацій, як OPC UA, підтверджує наявність тенденції в промисловій автоматизації на використання сучасних технологій.

Ключові слова: OPC UA, Java, React, промисловість, веб-додаток, Spring Framework.

Вступ. В сучасному індустріальному середовищі, де технології та автоматизація відіграють визначальну роль, управління технологічними процесами стає ключовим елементом для ефективного та надійного функціонування виробництва. З метою оптимізації та підвищення продуктивності виробничих систем розробка сучасних інструментів для керування стає актуальною задачею.

Одним із перспективних напрямків є використання технології OPC UA (OLE (Object Linking and Embedding) for Process Control Unified Architecture) для забезпечення стандартизованого зв'язку та обміну даними між різноманітними пристроями та системами керування. У зв'язку з цим розробка веб-додатку для керування технологічними процесами, який базується на технології OPC UA, може стати вагомим кроком у напрямку створення ефективного та інтегрованого інструменту для вирішення завдань автоматизації та контролю виробничих процесів.

Постановка задачі. Для досягнення поставленої мети в роботі необхідно вирішити такі завдання, як аналіз принципів роботи та специфікації OPC UA, проектування та розробка програмної та апаратної частин проекту з урахуванням сумісності компонентів зі специфікацією OPC UA; здійснити відповідні дослідження роботи розробленого проекту.

Основний зміст роботи. У роботі були використані такі технології та інструменти, як специфікація OPC UA, мова програмування TypeScript, бібліотека React, фреймворк Spring, мова програмування Java.

Технологія OPC UA є розширенням оригінального стандарту OPC (OLE for Process Control), яке виходить за межі промислового контролю та включає в себе багато новаторських можливостей. Ключові аспекти технології OPC UA:

- OPC UA є відкритим стандартом, що розробляється для забезпечення інтероперабельності між різними системами автоматизації та промисловими пристроями;
 - забезпечує спрощену інтеграцію між різними платформами та пристроями незалежно від їхніх виробників;
 - враховує питання безпеки на високому рівні, зокрема підтримує сучасні методи шифрування та автентифікації для забезпечення конфіденційності та цілісності даних;
 - завдяки своїй архітектурі OPC UA дозволяє реалізацію розподіленого доступу, що полегшує взаємодію між великою кількістю пристроїв та систем;
 - архітектура OPC UA розглядається як модульна та розширювана, що дозволяє легко впроваджувати нові функції та розширювати можливості стандарту;
 - використовує стандартизовані інформаційні моделі для опису пристроїв та процесів, що полегшує роботу з даними та їх інтерпретацію;
 - підтримує множини транспортних протоколів, зокрема HTTP, MQTT, AMQP.

Для створення веб-додатку для керування технологічними процесами з використанням специфікації OPC UA обрані наступні компоненти: Сервер OPC UA, Клієнт OPC UA, База даних, Backend (на базі Java та Spring Framework), Frontend (на базі React).

В якості апаратного носія серверу OPC UA був обраний людино-машинний інтерфейс, реалізований на програмному забезпеченні для візуалізації та керування процесами Siemens WinCC, який також може використовуватися як сервер OPC UA для обміну даними з іншими системами та пристроями в індустріальному середовищі.

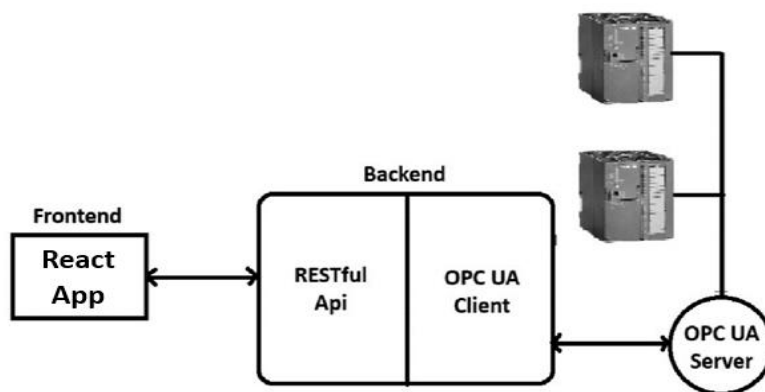


Рис. 1. Загальна модель веб-додатку

Для реалізації веб-додатку було вирішено використовувати Java у поєднанні зі Spring Framework для створення backend, оскільки OPC UA має набір інструментів SDK (*software development kit*) для інтеграції в додатки на мові Java. Для frontend було обрано React оскільки він відповідає сучасним вимогам

та дозволяє використовувати односторінкову архітектуру, що особливо корисно для відображення та керування технологічним процесом. Загальна модель веб-додатку зображена на рис. 1.

Висновки. Використання технології OPC UA в розробці веб-додатку для керування технологічними процесами дозволяє створити ефективний та стандартизований інструмент для взаємодії з пристроями та системами автоматизації в промисловому середовищі.

Впровадження стандарту OPC UA в систему керування дозволяє забезпечити стандартизований та безпечний обмін даними між різними пристроями та додатком, що є важливим фактором в контексті індустрії 4.0. Використання цього стандарту дозволяє ефективно інтегрувати пристрої в єдиний інтерфейс управління.

Дослідження можливостей OPC UA демонструє рівень розвитку промислових систем обміну даними та перспективи інтеграції з сучасними веб-технологіями.

ПЕРЕЛІК ПОСИЛАНЬ

1. The OPC Unified Architecture (UA) is a platform independent service-oriented architecture. [Електронний ресурс] Режим доступу: <https://opcfoundation.org/about/opc-technologies/opc-ua/>
2. SIMATIC WinCC Open Architecture Documentation [Електронний ресурс]. Режим доступу: https://www.winccoa.com/documentation/WinCCOA/3.18/en_US/OPC_UA/opc_ua_basics.html
3. Spring Boot Reference Documentation [Електронний ресурс]. Режим доступу: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
4. React Native official documentation [Електронний ресурс]. Режим доступу: https://reactnative.dev/docs/getting-started.com/documentation/WinCCOA/3.18/en_US/OPC_

УДК 004.415.3:681.6

М.Г. Бердник¹, О.В. Іванченко¹, О.С. Коваленко¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ РОЗРАХУНКУ РАЦІОНУ ХАРЧУВАННЯ ДЛЯ ХВОРИХ НА ЦУКРОВИЙ ДІАБЕТ ДРУГОГО ТИПУ

Анотація. В роботі побудовано математична модель, метод , а також розроблено програмне забезпечення для розрахунку раціону харчування хворих на цукровий діабет 2 типу, з можливістю моніторингу рівня цукру в крові шляхом проведення вимірювань та порівняння їх результатів з існуючими нормами.

Ключові слова: раціон харчування, хлібна одиниця, оптимізація, програмний додаток, програмні засоби C#, рівень цукру.

Вступ. Актуальність роботи полягає в розробці та апробації програмного забезпечення, здатного ефективно розраховувати оптимальне споживання поживних речовин, зокрема вуглеводів, для пацієнтів з цукровим діабетом 2

типу, оскільки це трудомісткий процес, що вимагає складних розрахунків, які враховують багато параметрів, таких як вік, стать, вага, кількість хлібних одиниць в прийомах їжі тощо. Багато людей, які змушені самостійно перевіряти рівень цукру в крові, стикаються з проблемою визначення вмісту кошика в супермаркеті.

Одним з ключових аспектів актуальності є економічний. Програма розраховує не тільки кількість спожитих вуглеводів, а також має функцію підрахунку витрат, що підвищує практичну цінність для кінцевого користувача. На основі наданих даних розроблений алгоритм враховує бюджетні обмеження користувача і мінімізує загальну вартість раціону. Ця стратегія враховує не лише ефективність харчування, але й реальні обмеження та економічні міркування, які можуть суттєво вплинути на здатність людини дотримуватися дієтичних рекомендацій.

Відсутність аналогічних інструментів у відкритому доступі в українській індустрії програмного забезпечення, а також поширення альтернативних продуктів під власницькими ліцензіями та акцент на характерних для інших країн продуктах харчування, дозволяють цьому програмному забезпеченню заповнити значні прогалини в лікуванні діабету та мати позитивний соціальний вплив. Високим практичним значенням є можливість впливати на процес розробки раціону харчування та дотримання відповідної дієти для пацієнтів з діабетом 2 типу. Розроблене програмне забезпечення може також використовуватися для моніторингу рівня цукру в крові та порівняння його з медичними нормами.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- розробити математичну модель, метод розв'язку та програмне забезпечення мовою програмування C# для розрахунку раціону харчування для хворих на цукровий діабет другого типу.

Програмне забезпечення повинно відповідати наступним функціональним вимогам:

- наявність зрозумілого та дружнього графічного користувальницького інтерфейсу;
- наявність можливості ведення даних про користувачів та продукти харчування для проведення подальших обчислень;
- наявність можливості аналізувати введені дані рівню цукру у крові впродовж заданого проміжку часу та будувати графік змін;
- наявність можливості обчислення раціону харчування на основі вказаних вимог (максимальної допустимої суми витрат, наявності або відсутності певних продуктів харчування, рекомендацій щодо споживання вуглеводів).

Додаток використовує дані користувача (вік, стать, ступінь фізичної активності), дані про продукти харчування (ціна, вага, кількість продуктів) та максимальну суму витрат як вхідні дані.

Основний зміст роботи. Людина з діабетом 2 типу повинна споживати певну кількість вуглеводів (вимірюється в хлібних одиницях) щодня, отже, вона повинна споживати певну кількість їжі, щоб отримати ці вуглеводи (хлібні одиниці). Згідно з дослідженнями, кожен продукт харчування має різний рівень вмісту вуглеводів через природу свого походження, що призводить до ситуацій, коли споживання одного продукту призводить до недотримання норми споживання хлібних одиниць, тоді як споживання іншого продукту призводить до перевищення цієї норми. Як наслідок, необхідно розробити обмеження на математичну модель, які б гарантували, що загальна кількість хлібних одиниць у спожитій їжі знаходиться в діапазоні від мінімального до максимального зазначеного значення, залежно від стану конкретного пацієнта. Такі обмеження можуть бути виражені математично у вигляді нерівності:

$$Crh_1 < \sum_{i=0}^{n-1} x_i \cdot BU_i < Crh_2, \quad (1)$$

Де Crh_1 – мінімальна денна рекомендована до спожиття кількість хлібних одиниць;

Crh_2 – максимальна денна рекомендована до спожиття кількість хлібних одиниць;

x_i – i -й продовольчий продукт;

BU_i – вміст хлібних одиниць в i -му продукті;

$i = 0, 1, 2 \dots n - 1$ – номер обраного продукту харчування;

n – загальна кількість обраних продуктів харчування.

На додаток до різного вмісту хлібних одиниць, кожен продукт має свою ціну, яка може коливатися в залежності від різних змінних, таких як неврожай, військові дії, неможливість імпорту або експорту тощо. Враховуючи необхідність розподілу матеріальних ресурсів та складність їх використання виключно для придбання продуктів харчування, варто додати до математичної моделі обмеження, щоб загальна кількість придбаних продуктів харчування не перевищувала суму, яку людина виділила б на бюджетні покупки. Математично, це можна представити у вигляді співвідношення:

$$\sum_{i=0}^{n-1} p_i x_i = h, h > 0, \quad (2)$$

де x_i – кількість придбаного i -го продукту харчування;

p_i – вартість придбаного i -го продукту харчування;

h – загальна сума, виділена на закупки продовольчих товарів.

Важливо також обмежити максимальну добову норму споживання кожного продукту, який буде включено до розрахунків, щоб уникнути сценаріїв з ірраціональними або навіть безглуздими цифрами, наприклад, 1 кілограм хліба на день. Обмеженням, яке дозволить це регулювати, є умова:

$$0 \leq x_i \leq Am_{max}, \quad (3)$$

де x_i – кількість придбаного i -го продукту харчування в грамах; Am_{max} – максимальна рекомендована кількість щоденного споживання продукту.

Варто зазначити, що максимальна добова норма споживання в грамах варіюється залежно від продукту. Наприклад, 300 г для хліба, 450 г для картоплі, 1,5 г для солі та цукру тощо. Оскільки сума, що витрачається на обрану пацієнтом їжу, не повинна становити більшу частину бюджету, доцільно мінімізувати цю функцію:

$$h \rightarrow \min$$

Розглянута функція може слугувати цільовою для створюваної математичної моделі: знайти $\min h$ при умовах (1)-(3).

Для вирішення цієї проблеми було розглянуто декілька методів оптимізації, включаючи метод найменших квадратів, лінійну оптимізацію, метод Пауелла та метод Нелдера-Міда. В результаті, для оптимізації рішення цієї задачі було обрано метод найменших квадратів, який використовується, щоб вирішити проблему визначення значень змінної x таким чином, щоб кожне з них на першій ітерації $i = 1, \dots, t$ було якомога ближче до заданого значення y_i . Математично, це описується виразом:

$$\sum_i e_i^2 = \sum_i (y_i - f_i(x))^2 \rightarrow \min_x$$

При розв'язанні задачі оптимізації:

$$\min_x f(x)$$

з обмеженнями

$$b(x) \geq 0,$$

$$c(x) = 0$$

метод найменших квадратів шукає значення функції Лагранжа:

$$L(x, \lambda, \sigma) = f(x) - \lambda^T b(x) - \sigma^T c(x),$$

де λ та σ – значення множників Лагранжа.

Для побудови програми використовується патерн проектування Model-View-ViewModel (MVVM). Він є умовним нащадком патернів MVC та MVP, а його ключовою перевагою є відокремлення візуальної складової від логіки її роботи.

Рівні коду в MVVM виглядають наступним чином:

1. Model відповідає за абстрагування джерел даних. Для доступу та збереження даних Model та ViewModel працюють разом. Моделі - це класи, які відображають предметну область програми. Моделі містять дані та бізнес-логіку програми. Це бізнес-об'єкти, які не мають нічого спільного з візуальним представленням програми.

2. ViewModel отримує повідомлення про дії користувача через View. Цей рівень не містить прикладного коду і лише керує ViewModel. Вся взаємодія з користувачем відбувається в межах View, яке відповідає за виявлення введення (кляцання миші, введення з клавіатури тощо) і передачу його до Model представлення за допомогою зв'язування даних. Фізичним зв'язком між View і ViewModel є зв'язування даних, яке може бути досягнуто за допомогою зворотних викликів або властивостей.

3. ViewModel - пропонує потоки даних, що стосуються подання. Вона також слугує зв'язком між моделлю та представленням. ViewModel відповідає за реалізацію властивостей і дій, які можуть бути пов'язані з представленням. Ці властивості та дії визначають функціональність, яку може надати користувачеві представлення, але відображення цієї функціональності повністю залежить від представлення. ViewModel також відповідає за передачу даних з класів Model до View. Для цього ViewModel може безпосередньо надавати доступ до View класам Model, у цьому випадку клас Model повинен підтримувати прив'язку даних та події сповіщення про зміни.

Інтерфейс користувача програмного додатку створено з використанням технології Windows Presentation Foundation. Для підключення до бази даних PostgreSQL використано технології Entity Framework Core.

Створений програмний продукт має зручний традиційний інтерфейс користувача, який дозволяє користувачеві налаштовувати розміри програмних вікон.

Програма містить механізм реєстрації та авторизації, створений з дотриманням сучасних заходів безпеки для збереження результатів обчислень та персоналізації облікового запису користувача.

Наукова новизна. Розроблено метод і програмне забезпечення розрахунку оптимального раціону харчування хворих на цукровий діабет другого типу в залежності від індивідуальних даних хворих та із врахуванням максимальної можливої суми витрат на продукти харчування.

Висновки. За результатами дослідження ефективності розробленого програмного забезпечення були зроблені наступні висновки:

– програма є ефективним інструментом для розрахунку раціону харчування з урахуванням поживності та кількості вуглеводів, необхідних для задоволення щоденних потреб користувача, виходячи з його особливостей та ступеня фізичної активності відповідно до медичних рекомендацій;

– правильна конфігурація обмежень дозволяє сформувати кошик, який відповідає потребам добового споживання вуглеводів і водночас має найнижчу вартість для заданого набору товарів.

Створене програмне забезпечення дозволяє користувачеві контролювати рівень глюкози в крові та порівнювати його з медичними нормами, а також з переліком розрахованих продуктових кошиків.

УДК 004.77

О.Г. Галушко¹, Н.О. Соколова¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА СЕРВЕРНОГО КЛАСТЕРУ НА ОСНОВІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВІДКРИТИМ КОДОМ

Анотація. Організація групи серверів у спільну мережу (кластер), для підвищення продуктивності та доступності служби шляхом розподілу робочого навантаження між декількома серверами та забезпечення резервування на випадок збою одного з серверів.

Ключові слова: сервер, програмне забезпечення, мережа, MySQL, LAMP, Linux, Apache, PHP, WordPress, сайт, технології, кластеризація.

Вступ. В сучасному світі надважливими є комунікації. Перерви в роботі серверів можуть призвести як найменше до невдоволення клієнтів, навіть до фінансових втрат, негативного впливу на репутацію бізнесу, а в умовах пандемій та війн і до катастрофічних наслідків, тому для будь-якого бізнесу, який пропонує свої послуги в Інтернеті, надзвичайно важливо мати високий рівень доступності серверів. Одним з методів вирішення цієї проблеми є володіння компаніями кластерами серверів, щоб забезпечити постійну доступність для клієнтів відповідних сервісів.

Кластер серверів – це група серверів, які працюють разом для виконання одного завдання, наприклад обслуговування веб-сторінок або розміщення бази даних. Кластери серверів використовуються для підвищення продуктивності та доступності служби шляхом розподілу робочого навантаження між декількома серверами та забезпечення резервування на випадок збою одного з серверів [1].

Організація мережі об'єднаних серверів є ключовою для забезпечення безвідмовної роботи веб-додатків та серверів. Це дозволяє розподіляти навантаження між серверами, зменшує ризик відмови при перевантаженні та забезпечує резервне зберігання даних, забезпечуючи високу доступність і надійність системи. Така організація дозволяє ефективно керувати трафіком, запобігає відмовам через одиночні точки виходу та забезпечує швидку відновлюваність у випадку непередбачуваних ситуацій [2].

Забезпечення безпеки систем об'єднаних серверів включає в себе заходи, спрямовані на захист від кіберзагроз та несанкціонованого доступу до даних. Це охоплює використання шифрування для захисту передачі даних, встановлення міцних систем аутентифікації та авторизації, а також регулярне оновлення програмного забезпечення для запобігання використанню вразливостей.

Подібні системи відіграють ключову роль у сучасних інформаційних технологіях, забезпечуючи надійність доступу до даних та сервісів, ефективне використання ресурсів, сегментацію та захист даних, можливість масштабування відповідно до потреб та ефективне моніторингове планування. Вони надають структуровану та стійку основу для роботи з даними, забезпечуючи стабільність та безпеку у важливих аспектах функціонування інформаційних систем [3].

Постановка задачі. Мета поставленого проекту полягає у формуванні і вирішенні таких завдань:

- Аналіз задач і вимог.
- Проектування мережі
- Підбір технологій і програмного забезпечення
- Налаштування і підготовка віртуальних серверів до роботи
- Організація групи серверів у спільну мережу
- Оцінка і аналіз покращення системи

Основний зміст роботи. Під час виконання роботи використовувались наступні інструменти та програмне забезпечення:

- Ubuntu 22.04: операційна система для серверів.
- Apache: веб-сервер обробки запитів для передачі через HTTP статичних та динамічних вебсторінок.
- Nginx: вебсервер і проксі-сервер для Unix-систем.
- Rsync: утиліта для синхронізації файлів і резервного копіювання
- WordPress: система керування вмістом з відкритим кодом
- MariaDB: реляційна система керування базами даних
- phpMyAdmin: веб-додаток з відкритим кодом із графічним веб-інтерфейсом для адміністрування бази даних MySQL або MariaDB.
- Secure Shell, SSH – мережевий протокол прикладного рівня, що дозволяє проводити віддалене управління комп'ютером і тунелювання TCP-з'єднань

Цей проект складався з широкого переліку етапів процесу створення кластеру. На початку аналізу задач та проектування мережі, було розглянуто оптимальні технології, інструменти та підходи, що відповідають основним вимогам вирішення задач. Створення кластеру з використанням програмного забезпечення з відкритим кодом дозволило забезпечити безвідмовну та стабільну роботу розташованого сайту у мережі Інтернет.

Під час виконання проекту було використано сучасні інструменти для роботи, такі як: веб-сервери Apache та Nginx – для обробки запитів, MariaDB – для роботи з базами даних на віддаленому сервері, SSH – один з головних інструментів, який надавав змогу налаштовувати та встановлювати все необхідне ПЗ за допомогою тунелювання.

Ця система виділяється своєю здатністю до автоматичної стійкості до відмови завдяки наявності резервного сервера в іншій локації, що забезпечує безперервність роботи навіть при недоступності основного сервера. Поділ функцій між серверами, балансування навантаження та централізоване

зберігання даних підвищують ефективність та надійність системи, забезпечуючи більш стабільне функціонування та покращене управління ресурсами.

Архітектура та налаштування цієї системи надають змогу зберігати внесені зміни на сайті або в базі даних, навіть під час недоступності основного серверу. Оскільки резервний сервер використовується не тільки для зберігання резервної копії сайту, а виконує синхронізацію даних, щоб відображати внесені зміни, у випадку недоступності основного серверу. Тобто резервний сервер бере на себе роль основного, під час недоступності основного серверу - завдяки налаштуванням балансувального серверу.

Синхронізація даних допомагає зберігати та тримати актуальними всі внесені зміни на сайті шляхом синхронізації файлів, а віддалена база даних – зміни у базі даних, оскільки основний і резервний сервер мають віддалений доступ до бази даних.

Висновки. Використання обраних технологій дозволило створити систему, що може підтримувати роботу сайту або веб-додатка у мережі, надавати до нього безперебійний доступ та розподілити навантаження на декілька серверів, зі змогою використання резервного сервера, у випадку недоступності основного без втрати даних і простою у роботі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Що таке кластер серверів: усе про цей тип хостингу?, digiexe. URL: <https://www.digiexe.com/uk/blog/server-clusters/>
2. Building Microservices: Designing Fine-Grained Systems. Sam Newman, 2015. 278 с.
3. Jarret B. Linux Cluster – Basics. <https://medium.com/@OpsTalk/build-a-web-server-cluster-using-docker-linux-and-microsoft-in-1-hour-def320735b76>

ЗАБЕЗПЕЧЕННЯ ЕФЕКТИВНОЇ ІНТЕГРАЦІЇ: РОЛЬ RESTFUL API В РОЗВИТКУ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ OPENCART

Анотація. Розглянуто ключові аспекти взаємодії мобільних додатків із системою управління контентом (CMS) Opencart за допомогою технології RESTful API. Досліджуються сучасний стан синтезу мовлення, методи глибинного навчання та алгоритми FastSpeech, що визначають швидкість синтезу голосу. Особлива увага приділяється проблемам взаємодії "один до багатьох", авторегресійної послідовності та альтернативним не авторегресійним моделям. У статті також розглядаються різні методи та технології взаємодії мобільних застосунків з RESTful API, аналізуються виклики та проблеми цього процесу, а також оцінюється ефективність наявних підходів до інтеграції мобільних додатків із CMS системою Opencart.

Ключові слова: мобільні додатки, RESTful API, Opencart, CMS, взаємодія програмного забезпечення, інтеграція додатків, технології взаємодії, аналіз ефективності, альтернативи.

Вступ. В сучасних реаліях ІТ-технологій мобільні додатки стають невід'ємною частиною ведення електронного бізнесу. Забезпечуючи зручний інтерфейс та доступ до продуктів та послуг, вони стають потужним інструментом для покращення користувальницького досвіду та збільшення обсягів продажів. Одним із найбільш популярних рішень для створення та управління електронними магазинами є використання CMS Opencart. Проте, ключовим завданням сьогодення є не просто створення мобільного додатка, а й його ефективна взаємодія з основною системою управління контентом. У цьому контексті виникає питання інтеграції мобільних додатків з CMS, особливо через використання RESTful API. В даній роботі висвітлено сучасний стан взаємодії мобільних додатків з CMS Opencart, а також досліджено ключові аспекти цього процесу. Розглядаються технічні виклики, що виникають при інтеграції, а також проаналізовано різні методи та технології для забезпечення ефективної взаємодії, для забезпечення максимальної продуктивності, безпеки та зручності в управлінні електронним бізнесом.

Постановка задачі. Аналіз взаємодії мобільних застосунків із системою Opencart за допомогою RESTful API розкриває велику кількість ключових викликів та потенційних проблем. Однією з центральних складнощів є забезпечення оптимальної продуктивності та надійності у великому обсязі обміну даними. Серед основних викликів можна виділити ефективне керування сесіями, яке є важливим для забезпечення безпеки та конфіденційності взаємодії. Проблеми можуть виникнути при неадекватній обробці та передачі великого обсягу інформації, що вимагає ретельної оптимізації. Крім того, необхідно розглядати питання сумісності та взаємодії різних версій мобільних додатків та CMS. Також важливо вирішити завдання автоматизованого тестування для

забезпечення високої якості інтеграції. В даних дослідженнях будуть враховані виявлені проблеми для розробки стратегій подолання та оптимізації інтеграційного процесу, з метою досягнення максимальної ефективності взаємодії між системами та задоволення вимог користувачів із врахуванням особливостей системи OpenCart.

Основний зміст роботи. На рис. 1 наведено порівняльний аналіз методів інтеграції JSON Web Tokens (JWT), OAuth 2.0, Webhooks.

Метод/Технологія	Переваги	Недоліки
JSON Web Tokens (JWT)	- Висока безпека даних	- Потребує додаткового оброблення на стороні сервера
	- Компактність передачі даних	- Обмежена підтримка окремими версіями браузерів
	- Широке використання в інтернет-просторі	
OAuth 2.0	- Дозволяє делегувати доступ до ресурсів	- Складний для реалізації
	без передачі облікових даних	- Високий ризик зловживання правами доступу
	- Забезпечує стандартизований механізм авторизації	- Вимагає управління токенами та їх оновлення
Webhooks	- Миттєве сповіщення про події	- Залежність від стабільності та доступності сторонніх серверів
	- Зменшення навантаження на сервер	
	- Інформативність та легкість відлагодження	- Потребує налагодження на обох сторонах

Рис. 1. Порівняння методів інтеграції

Аналіз вищезазначених методів/технологій показав наступне:

- JSON Web Tokens (JWT) надає високий рівень безпеки даних та компактність передачі, що робить його популярним серед розробників. Однак йому притаманні певні недоліки, такі як потреба у додатковому обробленні на серверному боці та обмежена підтримка певними версіями браузерів.
- OAuth 2.0 дозволяє ефективно делегувати доступ до ресурсів без передачі облікових даних. Незважаючи на його переваги, реалізація цього протоколу може бути складною, а висока ризик зловживання правами доступу вимагає уважного управління токенами та їх оновлення.
- Webhooks є методом сповіщення про події, забезпечуючи миттєвий обмін інформацією між системами. Використання цього методу допомагає зменшити навантаження на сервер і забезпечити надійне функціонування.

Подана таблиця порівняння дозволяє зробити об'єктивний вибір методу/технології, враховуючи конкретні потреби та характеристики проекту. Ретельний аналіз дозволяє замовнику визначитися з оптимальним підходом для забезпечення успішної інтеграції мобільних додатків з RESTful API системи Opencart.

Для оцінювання ефективності різних підходів була використана формула (1):

$$\text{Ефективність} = \frac{\text{Кількість успішних інтеграцій}}{\text{Загальна кількість тестових запитань}} \times 100\% \quad (1)$$

Де "Кількість успішних інтеграцій" визначалася як кількість випадків, коли мобільний застосунок успішно обмінювався даними з Opencart через RESTful API. "Загальна кількість тестових запитань" представляла в собі сукупність тестових сценаріїв, які охоплювали різні аспекти взаємодії. В таблиці 1 наведено ефективність різних підходів для обрання оптимального методу взаємодії, враховуючи конкретні потреби та вимоги проекту.

Таблиця 1.

Ефективність підходів інтеграції

Метод/Технологія	Кількість тестів	Успішні інтеграції	Ефективність, %
JWT	50	45	90
OAuth 2.0	50	48	96
Webhooks	50	50	100

Процес розробки передбачав створення функціоналу для взаємодії з ключовими елементами Opencart, такими як каталог товарів, корзина покупок та інші. Додаток був детально протестований для перевірки правильності інтеграції та забезпечення його коректної роботи.

Під час апробації застосунку, були визначені його ключові можливості та обмеження. Оцінка ефективності включала аналіз часу відгуку системи, надійності передачі даних, та інших показників, необхідних для забезпечення стабільної роботи додатку в реальних умовах.

Отримані результати стали основою для подальших висновків та рекомендацій з удосконалення взаємодії мобільних додатків із CMS Opencart через RESTful API.

Під час оцінки продуктивності та надійності RESTful API в системі управління контентом Opencart були вивчені основні показники, що впливають на ефективність даного інтерфейсу. Аналізувалися такі аспекти, як швидкодія передачі даних, відповіді сервера на запити, а також обробка великого обсягу транзакцій та завантаження сервера під час великих навантажень.

Отримані дані щодо продуктивності RESTful API дозволили визначити його потужності та обмеження. Було оцінено рівень масштабованості системи при великому обсязі запитів, а також стабільність роботи API в умовах збільшення навантаження.

На основі проведених досліджень можна зробити висновки щодо того, як оптимізувати та покращити роботу RESTful API в Opencart для забезпечення найвищого рівня продуктивності та надійності при взаємодії з мобільними додатками.

При розгляді підходів до крос-платформної розробки мобільних застосунків в контексті взаємодії з системою управління контентом Opencart, був проведений аналіз використання крос-платформних рішень та їхнього впливу на ефективність інтеграції.

Визначалися переваги та недоліки використання різних фреймворків для крос-платформної розробки, зокрема React Native, Flutter, Xamarin тощо. Розглядалися аспекти швидкодії, надійності та можливостей адаптації до специфіки Opencart.

Отримані результати дозволяють зробити висновки щодо оптимального вибору інструментарію для крос-платформної розробки мобільних додатків з урахуванням особливостей Opencart та забезпечення максимальної ефективності взаємодії між додатками та системою управління контентом.

На рис.2 наведено результати оцінки продуктивності та надійності RESTful API в Opencart за наступними показниками: кількість запитів, середній час відповіді сервера та відсоток успішних відповідей.

Показник	Значення
Кількість запитів на сервер	1000
Середній час відповіді сервера, мс	150
Відсоток успішних відповідей	98%

Рис. 2. Основні показники оцінки продуктивності

Отримані дані дозволяють здійснити аналіз ефективності взаємодії та визначити можливості для подальшого вдосконалення системи.

Наукова новизна. У контексті наукової новизни, дослідження взаємодії мобільних додатків з CMS Opencart через RESTful API вносить вагомий внесок у розуміння ефективності таких взаємодій. В роботі запропоновано детальний аналіз існуючих методів, що визначає їхню ефективність та ідентифікує можливі проблеми в контексті використання Opencart. Запропонований розгляд різних методів/технологій інтеграції мобільних додатків з RESTful API, а також розгляд крос-платформної розробки для Opencart, становлять вагомий внесок у розгляд використання сучасних рішень для покращення взаємодії та розвитку мобільних додатків в екосистемі Opencart.

Висновки. В результаті проведеного дослідження взаємодії мобільних додатків з CMS Opencart через RESTful API було виявлено важливі аспекти та розроблені рекомендації для оптимізації цього процесу. Отримані результати визначають нові можливості та сприяють покращенню взаємодії мобільних додатків з Opencart, що має актуальним для розвитку електронної комерції та покращення користувацького досвіду.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ranganathan, S. (2019). RESTful Web API Design with Node.js 10: Learn to create robust RESTful web services with Node.js, MongoDB, and Express.js. Packt Publishing. – 276 с.
2. Richardson, L., & Amundsen, M. (2013). RESTful Web APIs: Services for a Changing World. O'Reilly Media. – 408 с.
3. Tilkov, S., & Vinoski, S. (2010). REST und HTTP: Einsatz der Architektur des Web für Integrationsszenarien. dpunkt.verlag. – 300 с.
4. O'Reilly, T. (2007). What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. O'Reilly Media. – 144 с.
5. Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine. – 200 с.

РОЗДІЛ 4

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ ДЛЯ ВИРШЕННЯ ЗАВДАНЬ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ

УДК 004.94+519.7

В.В. Слесарєв¹, О.С. Сенченко², М.І. Притула²

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Інститут прикладної математики і механіки НАН України, Слов'янськ, Україна

МОДЕЛЮВАННЯ ЛІНГВІСТИЧНОГО ПРЕДСТАВЛЕННЯ ДЕТЕРМІНОВАНИХ ГРАФІВ З ВИКОРИСТАННЯМ ПРОГРАМНИХ ЗАСОБІВ

Анотація. Описано результат реалізації програмної моделі лінгвістичного представлення детермінованих графів. Наведено метричні оцінки програмного коду. Показано переваги наведеної реалізації та окреслено можливий напрямок її використання.

Ключові слова: *детермінований граф, програмна метрика, лінгвістичне представлення, моделювання.*

Вступ. На сьогодні програмне моделювання є ефективним інструментом аналізу багатьох технічних задач та широко використовується у загальній теорії систем. Однією з таких задач є зручне задання як скінченних, так і нескінченних об'єктів. Ефективним способом такого задання є представлення об'єкту за допомогою твірних елементів та визначальних співвідношень між ними. У математиці відомо задання таким способом груп, напівгруп, та автоматів без виходу. Існує ряд теоретичних та прикладних задач, які безпосередньо пов'язані із указаним вище способом задання. Наразі таке представлення розповсюджено на детерміновані графи, які можуть бути ефективно використані в якості середовища у багатьох прикладних задачах. При цьому, детерміновані графи задаються словами в алфавіті міток їх вершин, тому можна говорити саме про лінгвістичне представлення детермінованих графів. Оскільки на даний час графи є концептуальним інструментом технічної документації та, крім того, є широко вживаною програмною абстракцією, тому важливу роль відіграє програмне моделювання результатів з метою їх підтвердження, аналізу, уточнення та удосконалення практичною реалізацією. Крім того, можливе застосування лінгвістичного представлення детермінованих графів у складних інженерних системах потребує розробки програмних прототипів алгоритмів, які запропоновано у цьому представленні, з метою подальшого аналізу можливості їх впровадження в промислові стандарти.

Основні означення. Розглядаються неорієнтовані, скінченні, непорожні, зв'язні прості графи з розміченими вершинами $G = (V, E, X, \xi(V))$, де V – множина вершин графа, E – множина його ребер, $\xi(V): V \rightarrow X$ – всюдिवизначена функція розмітки вершин графа символами скінченного алфавіта $X = \{x_1, \dots, x_p\}$. Розмічений граф G називається детермінованим (Д-графом) [1], якщо всі вершини у відкритому околі кожної його вершини мають різні мітки: $\forall v, v_1, v_2 \in V (v_1, v_2 \in E(v) \wedge \xi(v_1) = \xi(v_2)) \rightarrow v_1 = v_2$. Шлях $p = v'_1 \dots v'_k$ розглядаємо як послідовність $\xi(p) = \xi(v'_1) \dots \xi(v'_k)$ міток вершин, що входять до p . У випадку, коли відома карта графа та початкова вершина шляху, за послідовністю міток $\xi(p)$ однозначно відтворюється послідовність вершин, що входять до шляху, тому Д-графи є дуже зручними для їх дослідження за допомогою агентів, що переміщуються всередині графу. Зафіксуємо вершину $v_0 \in V$, яку будемо називати ініціальною, далі вважаємо, що всі шляхи в графі починаються з ініціальної вершини; ініціальну вершину будемо додавати у позначення графа. Шлях (слово) $p = x'_1 \dots x'_k$ назвемо припустимим для вершини $v'_1 \in V$, якщо $\xi(v'_1) = x'_1$, та існують такі вершини $v'_2 \dots v'_k \in V$, що $\xi(v'_2) = x'_2, \dots, \xi(v'_k) = x'_k$ і $(v'_1, v'_2), \dots, (v'_{k-1}, v'_k) \in E$.

Лінгвістичне представлення детермінованих графів. Теорія лінгвістичного представлення Д-графів активно розробляється співробітниками відділу теорії керуючих систем Інституту прикладної математики і механіки НАН України, опубліковано низку наукових статей, присвячених цій тематиці. Наведемо теоретичні відомості, присвячені лінгвістичному представленню детермінованих графів, за останньою опублікованою роботою [2].

Було розроблено представлення Д-графа $G = (V, E, X, \xi(V), v_0)$, у якого $\xi(v_0) = x'$ парою $\{C, L\}(x')$ скінченних множин слів $C, L \in X^*$, для якої всі слова з C та L є припустимими для v_0 , слова множини C описують цикли графа G , а слова L описують його висячі вершини. Під терміном «пара» $\{C, L\}(x')$ розуміємо дві скінченні множини слів C та L , що відповідають умовам:

- 1) будь-яке слово з множини C починається та закінчується символом x' ;
- 2) будь-яке слово з множини L починається на символ x' ;
- 3) довжина будь-якого слова з множини C більша ніж 2.

Множини C та L називають компонентами пари $\{C, L\}(x')$.

Було висунуто вимоги, за виконанням яких детермінований граф $G = (V, E, X, \xi(V), v_0)$, у якого алфавіт X співпадає з множиною всіх символів, які присутні в словах з обох компонентів пари $\{C, L\}(x')$, вважається представленням за цією парою:

- а) $\xi(v_0) = x'$;
- б) всі слова з C та L є припустимими для вершини v_0 ;
- в) кожне слово r з C визначає у G цикл $v_0 r = v_0$;
- г) для кожного слова $q \in L$ у G вершина $v_0 q$ є висячею;
- д) для кожної висячої вершини $v \in V$, що є відмінною від ініціальної вершини v_0 , існує хоча б одне таке слово $q \in L$, що $v_0 q = v$;

е) для будь-якої пари $\{C, L\}(x')$ або не існує представлення, або це представлення визначається однозначно.

При цьому декільком різним парам може відповідати одне й те ж представлення.

Нехай $X = \{x_1, \dots, x_m\}$ і $x_1 \leq x_2 \leq \dots \leq x_m$ – довільно зафіксований лінійний порядок P на X . На множині всіх слів в алфавіті X вводиться лінійний порядок \preceq :

- 1) $x_i \preceq x_i$;
- 2) якщо $d(p) < d(q)$, то $p \preceq q$, де $d(p)$ – довжина слова p ;
- 3) $x_1' \dots x_s' = p \preceq q = x_1'' \dots x_s''$, якщо $x_k' \leq x_k''$ для деякого $k \leq s$, в той час як $x_k' = x_k''$, \dots , $x_{k-1}' = x_{k-1}''$.

Нехай $G = (V, E, X, \xi(V), v_0)$ – довільний граф з розміченими вершинами. Редукцією G називається процедура, яка перетворює G у детермінований граф $[G]$ за таким алгоритмом (AP):

0) Покладемо $V' = V, E' = E, \xi'(V') = \xi(V)$, позначимо $G' = (V', E', X, \xi'(V'), v_0)$.

1) Для кожної вершини $v \in V'$ знаходимо найменше за \preceq слово w , що відповідає шляху від v_0 до v . Пов'язуємо v з w .

2) Впорядковуємо вершини графа G' за пов'язаними з ним словами за порядком \preceq . Поточною вершиною v_c призначимо першу за вказаним порядком вершину, а поточною міткою призначимо $v = x_1$.

3) Якщо існують такі різні вершини $v'_1 \dots v'_q$, для яких одночасно виконуються умови $v'_1 \dots v'_q \in E'(v_c)$ та $\xi'(v'_1) = \dots = \xi'(v'_q) = x$, то позначимо $U = \{v'_1 \dots v'_q\}$ та виконуємо таку послідовність дій:

- 3.1. до множини V' додаємо нову вершину v' та покладемо $\xi'(v') = x$;
- 3.2. з множини E' вилучаємо ребра (v_i, v_j) , де $v_i, v_j \in U$;
- 3.3 для кожного ребра (v_i, v_j) , де $v_i \in U$, до множини E' додаємо ребро (v', v_j) ;
- 3.4. з множини E' вилучаємо ребра (v_i, v_j) , де $v_i \in U$, з множини V' вилучаємо вершини v , де $v \in U$;
- 3.5 якщо $v_0 \in U$, то вершину v' перейменовуємо на v_0 та вважаємо далі цю вершину ініціальною;
- 3.6. вилучаємо повтори ребер, залишаючи по одному екземпляру;
- 3.7. переходимо на крок 1.

4) Якщо існує $x' \in X$, який є наступним за x за порядком $<$ (тобто, $x \neq x_p$), то покладемо $x = x'$ та переходимо до кроку 3.

5) Якщо існує вершина v , що є наступною для v_c , то покладемо $v_c = v, x = x_1$ та переходимо на крок 3, у протилежному випадку алгоритм AP завершує свою роботу та $[G] = G'$.

Неважко бачити, що виконання алгоритму AP закінчується за скінченну кількість кроків, його результат визначається однозначно та цей результат є Д-графом.

Також у [2] визначено алгоритм (АП), який за заданою парою $\{C, L\}(x')$ або будує Д-граф $G(\{C, L\}(x'))$, або показує, що за цією парою неможливо побудувати Д-граф, що відповідає умовам (а) – (е), зазначеним вище. У процесі тестування програмної моделі було прийнято рішення про модифікацію алгоритму АП. Ця модифікація полягає у певному спрощенні процесу перевірки відповідності побудованого графу зазначеній вище умові (д). Наведемо модифіковану версію алгоритму АП.

0) Покладемо, що спочатку граф $G(\{C, L\}(x'))$ складається з однієї вершини v_0 з міткою $\xi(v_0) = x'$.

1) У відповідність кожному слову $p^i = x'x_1^i \dots x_n^i x' \in C$ додаємо до графа вершини v_1^i, \dots, v_n^i з мітками відповідно x_1^i, \dots, x_n^i та ребра $(v_0, v_1^i), (v_1^i, v_2^i), \dots, (v_{n-1}^i, v_n^i), (v_n^i, v_0^i)$. Після кожного такого додавання робимо редукцію одержаного графа.

2) Для кожного слова $p^j = x'x_1^j \dots x_n^j \in L$ виконуємо таку послідовність дій: додаємо до графа вершини v_1^j, \dots, v_n^j з мітками відповідно x_1^j, \dots, x_n^j , ребра $(v_0, v_1^j), \dots, (v_{n-1}^j, v_n^j)$ та робимо редукцію одержаного графа.

3) Розглядаємо всі слова множини L : якщо існує таке $p \in L$, що вершина $v_0 p$ не є висячою, то вважаємо, що граф $G(\{C, L\}(x'))$ не існує.

4) Для кожної висячої вершини $v \in G(\{C, L\}(x'))$ розглядаємо слова компоненти L : якщо не існує такого $p \in L$, що $v = v_0 p$, то вважаємо, що граф $G(\{C, L\}(x'))$ не існує.

Якщо в результаті виконання цієї процедури за парою $\{C, L\}(x')$ можливо побудувати граф $G(\{C, L\}(x'))$, то таку пару назвемо правильною. У алгоритмі АП перший та другий етапи створюють деякий граф, який, за умови успішної перевірки на етапах 3 та 4 є графом $G(\{C, L\}(x'))$. Якщо хоча б одна з перевірок на етапах 3 та 4 не виконується, то вважаємо, що граф $G(\{C, L\}(x'))$ не існує.

Правильну пару $\{C, L\}(x')$ називають визначальною для Д-графа G , якщо $G(\{C, L\}(x')) \cong G$.

Таким чином, алгоритм АП є частковим відображенням множин пар на множину Д-графів, за яким правильній парі відповідає Д-граф, що будується за алгоритмом АП.

Проілюструємо дію окремих етапів алгоритму АП на наступному прикладі:

Нехай $C = \{12341, 142451\}$, $L = \{152125423, 14523\}(1)$. На рис. 1А зображено граф, який отримуємо після виконання етапу 1, на рис. 1В зображено граф, який отримуємо після виконання етапу 2. У цьому графі вершини $v_0 152125423$ та $v_0 14523$ є висячими, тому перевірка на етапі 3 виконується успішно. Для виділеної вершини v з міткою 1 немає жодного слова $p \in L$, що $v_0 p = v$, тобто перевірка на етапі 4 не виконується, тому граф $\{C, L\}(1)$ не існує.

Зауважимо, що для пари $\{C, L'\}(1)$, де $L' = L \cup \{1521\}$ граф $\{C, L'\}(1)$ існує, він зображений на рис. 1.1 в.

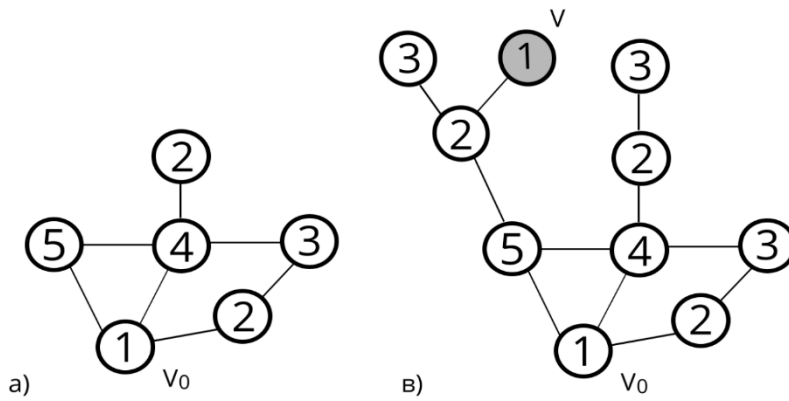


Рис. 1. Ілюстрація процедури побудови Д-графа за заданою парою

Також визначено особливу канонічну визначальну пару для будь-якого Д-графа. Нехай $G = (V, E, X, \xi(V), v_0)$ – деякий Д-граф, $V = \{v_0, \dots, v_{n-1}\}$ та $\xi(v_0) = x'$.

Визначається допоміжна множина слів в алфавіті X . Базисом досяжності \mathcal{V}_G називається така множина слів $\{w_0, \dots, w_{n-1}\}$, де $w_0 = x'$ та для кожної $v_i \neq v_0$ існує таке слово $w_i \in \mathcal{V}_G$, що виконується $v_0 w_i = v_i$, та для будь-якого $w \neq w_i$ з $v_0 w = v_i$, випливає $w_i \preceq w$. Кістякове дерево графа G , яке визначається базисом \mathcal{V}_G , позначають $T(\mathcal{V}_G)$ або $T(G, v_0)$. Для кожної вершини $v \in V$ через sp_v позначено таке слово з \mathcal{V}_G , що $v_0 sp_v = v$.

Визначено алгоритм АК побудови для Д-графа G визначальної пари $\{\Sigma_G, \Lambda_G\}$, яку, завдяки деяким її властивостям, було названо канонічною.

Спочатку покладемо $\Sigma_G = \emptyset$ і $\Lambda_G = \emptyset$. Якщо граф G складається з однієї вершини v_0 , то покладемо $\Sigma_G = \emptyset$, $\Lambda_G = \emptyset$ і $\mathcal{V}_G = \{\xi(v_0)\}$.

Нехай граф G містить більше ніж одну вершину. Спочатку до множини Λ_G додаємо усі слова $w \in \mathcal{V}_G$ такі, що вершина $v_0 w \in$ висячою вершиною графа G . Після цього для кожної двійки слів $p, q \in \mathcal{V}_G \setminus \Lambda_G$, якщо жодне з них не є початковим відрізком іншого і $v_0 p q^{-1} = v_0$, то додаємо до множини Σ_G одне з двох слів $p q^{-1}$ або $q p^{-1}$, яке є меншим за порядком \preceq (вершини $v_0 p$ та $v_0 q^{-1}$ називають твірними для того слова $p q^{-1}$ або $q p^{-1}$, що було додане до Σ_G). З алгоритмів АП та АК можна бачити, що пара $\{\Sigma_G, \Lambda_G\}$ є правильною та $G(\{\Sigma_G, \Lambda_G\}) \cong G$, тобто $\{\Sigma_G, \Lambda_G\}$ є визначальною парою для G .

Також було знайдено деякі властивості та такі метричні характеристики пари $\{\Sigma_G, \Lambda_G\}$: точне значення потужності першої компоненти, досяжні мінімальні та максимальні оцінки потужності другої компоненти, досяжні мінімальні та максимальні оцінки об'єму кожної компоненти у випадку, коли граф є деревом.

Програмне моделювання. В якості технічної специфікації для програмного моделювання алгоритмів АП, АР та АК визначено роботу [2]. Технічна реалізація [3] виконана з використанням мови програмування Python та бібліотеки для моделювання мереж NetworkX [4], з дотриманням вимог специфіки статичного та синтаксичного аналізатора MuPy та PyLint. Якість коду, згідно цих перевірок, програмно оцінена як 9.3 / 10 за стандартом PEP8. Ця

оцінка не є максимально можливою завдяки наявності в функціях збільшеної кількості, локальних змінних, а також використання назв змінних іменованих однією літерою, що є не рентабельним при написанні складних програмних модулів, але ця реалізація обрана свідомо, для відповідності термінам, використаним у роботі [2]. На рис. 2 наведено результати оцінки коду:

```
***** Module alglib
alglib.py:19:34: C0103: Argument name "G" doesn't conform to snake_case naming style (invalid-name)
alglib.py:114:4: C0103: Variable name "G_" doesn't conform to snake_case naming style (invalid-name)
alglib.py:137:13: C0103: Argument name "C" doesn't conform to snake_case naming style (invalid-name)
alglib.py:137:27: C0103: Argument name "L" doesn't conform to snake_case naming style (invalid-name)
alglib.py:137:41: C0103: Argument name "x_" doesn't conform to snake_case naming style (invalid-name)
alglib.py:137:0: R0914: Too many local variables (22/15) (too-many-locals)
alglib.py:142:4: C0103: Variable name "q" doesn't conform to snake_case naming style (invalid-name)
alglib.py:146:4: C0103: Variable name "G" doesn't conform to snake_case naming style (invalid-name)
alglib.py:159:0: C0103: Variable name "G" doesn't conform to snake_case naming style (invalid-name)
alglib.py:161:4: C0103: Variable name "q" doesn't conform to snake_case naming style (invalid-name)
alglib.py:173:0: C0103: Variable name "G" doesn't conform to snake_case naming style (invalid-name)
alglib.py:137:0: R0912: Too many branches (20/12) (too-many-branches)
alglib.py:223:4: C0103: Variable name "ni" doesn't conform to snake_case naming style (invalid-name)

-----
Your code has been rated at 9.30/10 (previous run: 9.30/10, +0.00)
```

Рис. 2. Коефіцієнт якості коду

На рис. 3 наведено шкалу складності коду за метрикою МакКейба [5] та значення цієї метрики для програмного модуля, що розглядається в даній роботі. Всі показники знаходяться на достатньому рівні, чотири з них знаходяться на найкращому рівні, з чого можна зробити висновки, що код піддається тестуванню та підтримці.

CCscore	Rank	Risk	
1-5	A	low - simple block	F 137:0 ap_graph - C (20)
6-10	B	low - well structured and stable block	F 199:0 ac_pair - C (15)
11-20	C	moderate - slightly complex block	F 59:0 check_q_node - C (13)
21-30	D	more than moderate - more complex block	F 89:0 word_pair_data_validation - C (12)
31-40	E	high - complex block, alarming	F 112:0 ar_nodes - B (8)
41+	F	very high - error-prone, unstable block	F 26:0 find_neighbours_with_the_same_labels - A (5)
			F 19:0 get_all_leaf_nodes_from_graph - A (4)
			F 41:0 walk_by_word - A (4)
			F 242:0 get_canonical_pair_metrics_from_graph - A (4)

Рис. 3. Оцінки цикломатичної складності коду за метрикою МакКейба

Також було обчислено значення метрик Halstead Complexity (складність Холстеда) [5]. Ці метрики надають інформацію щодо складності та розміру коду в файлі. Значення для параметрів difficulty, effort та bugs, що характеризують відповідно міри складності розуміння алгоритму, кількості зусиль, необхідних

для розробки й підтримки алгоритму та коефіцієнт приблизної кількості помилок, вказують на прийнятну якість програмного коду. Зауважимо, що ці метрики є евристичними вимірюваннями, і їх слід враховувати у контексті вимог проекту. На рис. 4 наведено значення параметрів цих метрик для алгоритмів АП (ap_graph) та АК (ac_pair).

ap_graph:	ac_pair:
h1: 8	h1: 8
h2: 22	h2: 20
N1: 19	N1: 16
N2: 35	N2: 27
vocabulary: 30	vocabulary: 28
length: 54	length: 43
calculated_length: 122.10749561002054	calculated_length: 110.43856189774725
volume: 264.97209216286	volume: 206.71626164847697
difficulty: 6.363636363636363	difficulty: 5.4
effort: 1686.1860410363818	effort: 1116.2678129017756
time: 93.67700227979898	time: 62.01487849454309
bugs: 0.08832403072095334	bugs: 0.06890542054949232

Рис. 4. Оцінки складності коду за метриками Холстеда

Завдяки програмній реалізації було проведено загальний аналіз процесу роботи зазначених вище алгоритмів та отримано візуалізацію результатів їх роботи. Також проведено перевірку гіпотези про те, що максимальне значення об'єму першої компоненти канонічної визначальної пари є у так званого «граф-квітки», який зображено на рис. 5. Для такого графа було знайдено це значення, а для будь-якого іншого Д-графа з такою ж кількістю вершин та ребер, що було програмно згенеровано, це значення було меншим, ніж для «граф-квітки». Крім того запропоновано можливість застосування алгоритмів АП та АК у контексті сучасної теорії побудови карти тестового покриття для програмного забезпечення, що може бути корисним для використання в процесах підвищення якості програмних продуктів.

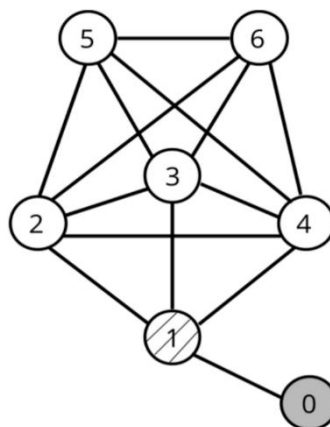


Рис. 5. «Граф-квітка», у якого 7 вершин та 14 ребер

Висновки. У роботі розглянуто програмне моделювання алгоритмів, пов'язаних з лінгвістичним представленням детермінованих графів. Наведено

метричні характеристики реалізованого коду. Це моделювання сприяло модифікації алгоритмів та підтвердженню гіпотези про граф, у якого значення об'єму першої компоненти канонічної визначальної пари є максимальним.

ПЕРЕЛІК ПОСИЛАНЬ

1. I.S. Grunskii, I.A. Mikhaylova, S.V. Sapunov (2012). Domination on the vertices of labeled graphs. *Algebra and Discrete Mathematics*, Vol. 14, Is. 2, p. 174-184.
2. О.С. Сенченко, М.І. Припула, О.А. Середа (2022). Представлення детермінованих графів визначальною парою слів. *Праці ІПММ НАН України*, том 36, №2, с. 104-116. doi: 10.32782/1683-4720-2022-36-09..
3. This repository presents the implementation of algorithms for the linguistic representation of deterministic graphs, 2023. URL: https://github.com/NickIT87/IAMM_proceedings.
4. NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks, 2023. URL: <https://networkx.org/documentation/>.
5. Introduction to Code Metrics. URL: <https://radon.readthedocs.io/en/latest/intro.html>.

UDC 004.42 (8)

Artem Vizniuk¹, Ivan Laktionov¹, Grygorii Diachenko¹
¹Dnipro University of Technology, Dnipro, Ukraine

STUDY OF THE EFFECTIVENESS OF ML ALGORITHMS IN PREDICTING THE PROBABILITY OF CROP DISEASE OCCURRENCE

Abstract. The aim of this research is to study the effectiveness of ML algorithms in predicting the probability of the corn disease «Fusarium Head Blight» in Dnipro region of Ukraine. Linear regression, feedforward neural networks and random forest models were considered for prediction. The random forest model obtained the best metric score on the testing set: $R^2=0.965$, $RMSE=3.44$. Directions for further research were substantiated in the given subject area.

Keywords: *ML, regression, time series, disease prediction, crops, agriculture.*

Introduction. According to the agricultural production statistics, provided by the globally recognized FAO [1], cereal crops, which have tripled in production over the past 20 years, are the most widely cultivated agricultural crops in open field conditions. Based on the statistical analysis of the data, accumulated by the FAO [2], it was established that the most cultivated cereal crops (according to harvested areas) in Eastern and Southern Europe, and Ukraine, in particular, are wheat, corn and barley. Trend dynamic of harvested areas over a period from 2012 to 2021 for wheat, corn and barley in Ukraine is shown in Fig. 1.

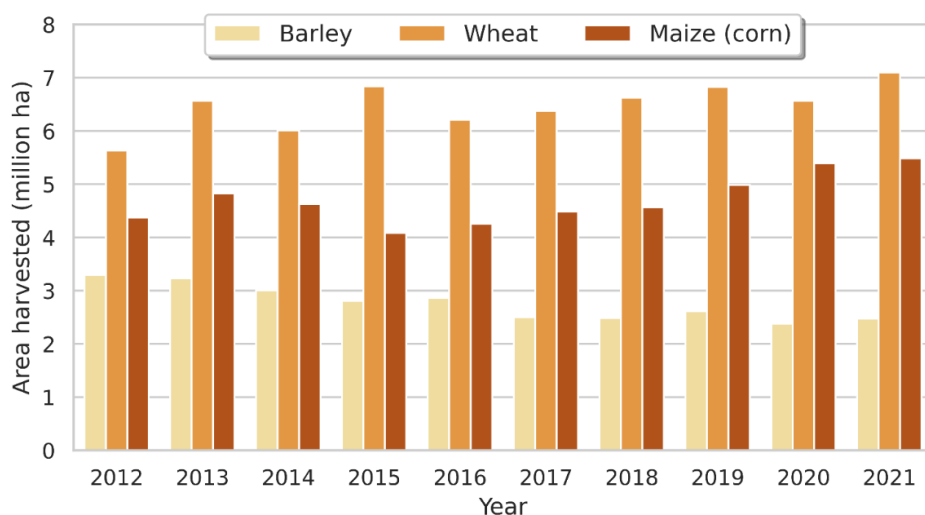


Figure 1. Area harvested for barley, wheat and corn crops in Ukraine over a period from 2012 to 2021

As can be seen in Fig. 1, wheat is the most commonly cultivated cereal crop in Ukraine, followed by corn and barley.

By analysing the last articles and research papers on the topic of crop disease prediction it was established, that fluctuations in climatic parameters, such as air temperature, relative humidity, precipitation, wind speed etc., have the biggest impact on crop disease occurrence [3–5]. Among the reviewed approaches to crop disease prediction based on climatic data ML algorithms demonstrated significant effectiveness in processing climatic data for predicting the probability of crop disease occurrence [3–5].

Problem statement. The aim of this research is to evaluate the effectiveness of ML algorithms in predicting the probability of the «Fusarium Head Blight» disease occurrence in corn in Dnipro region of Ukraine and to determine the most effective approach (according to R^2 and RMSE metrics) that will be integrated into a software application to provide real-time predictions of the disease probabilities of the observed crops. To achieve this aim, the following objectives were formulated and solved:

1. To substantiate the methods and means of research to evaluate the effectiveness of ML algorithms in predicting the probability of the «Fusarium Head Blight» disease in corn.

2. Determine the most effective regression model according to the R^2 and RMSE metric score on the testing set and the respective hyperparameters of this model.

3. Programmatically implement an algorithm for predicting the probability of corn disease using the most performant regression model.

Methodology. Climatic data with the «Fusarium Head Blight» disease probabilities for Dnipro region were downloaded from the METOS by Pessl Instruments weather station using the FieldClimate platform, which was provided by Metos Ukraine LLC. The values of air temperature, precipitation, air humidity, leaf wetness time, and disease probability were measured hourly from September 2022 to September 2023. Descriptive statistics of the collected data are shown in Table 1.

Table 1

Descriptive statistics of the collected data

	Air temperature (°C)	Precipitation (mm)	Humidity (%)	Leaf wetness duration (min)	Infection probability (%)
Count	8658				
Mean	10.62	0.07	72.8	8.7	4.84
Standard deviation	9.74	0.34	17.32	21.13	17.61
Median	9.65	0	76	0	0
Min	-10.9	0	19	0	0
Max	37	14	100	60	100

To account for the history of changes in climate parameters and disease probabilities, an additional hyperparameter timestamps is introduced, which determines how many previous values of climate parameters and disease probabilities are taken into account. For a timestamps value of 1, current climate parameters along with the climate parameters and disease probability an hour ago are used as input. For a timestamps value of 2, the climatic parameters and disease probability 2 hours ago are also provided as inputs.

Certain ML models, such as neural networks, are trained using a gradient descent algorithm, which converges faster when input values are on a similar scale. Therefore, the input values were transformed as follows:

- leaf wetness duration (min) was divided by the maximum value of 60;
- humidity (%) was divided by the maximum value of 100;
- precipitation (mm) and air temperature were standardized by the following equation: $x' = (x - \bar{x})\sigma^{-1}$, where x' – is the standardized value of a numerical feature x , \bar{x} – is the mean value of x on the training set, and σ – is the standard deviation of x on the training set;
- disease probability (%) was divided by the maximum value of 100.

The dataset was partitioned into training, validation and testing sets with a distribution of 70:15:15, corresponding to 6058 records for training, 1298 records for validation and 1299 records for testing purposes.

To predict the probability of the «Fusarium Head Blight» disease the following regression models were considered in this paper: linear regression, neural network and random forest. The research was conducted in the Google Colab environment using the programming language Python 3.10.12. Regression ML models hyperparameters which resulted in the best metric score on the testing set are presented in Table 2.

Results of the considered ML models

Model	Hyperparameters	Results on the testing set
Linear regression	timestamps=3	$R^2=0.96$, RMSE=3.61
Random forest	timestamps=3, max_depth=5, n_estimators=10	$R^2=0.965$, RMSE=3.44
Feedforward neural network	timestamps=3, optimizer=AdamW(learning_rate=0.001, weight_decay=0.003,beta_1=0.9,beta_2=0.999), layers=[Dense(10, activation=ReLU), Dense(5, activation=ReLU), Dense(3, activation=ReLU), Dense(2, activation=ReLU), Dense(1, activation=Linear)], batch_size=256, epochs=500	$R^2=0.962$, RMSE=3.59

In Table 2 it is shown that all considered ML models performed the best for the hyperparameter value timestamps=3. The random forest model demonstrated the best metric score on the testing set: $R^2=0.965$, RMSE=3.44.

The **scientific novelty** of the results obtained lies in the establishment of a set of input climatic data and the evaluation of the effectiveness of machine learning algorithms in predicting the probability of the «Fusarium Head Blight» disease in corn, which makes it possible to optimize the choice of the approach when designing specific types of systems, taking into account the R^2 and RMSE metrics.

Conclusions. In the course of the research on predicting the probability of the «Fusarium Head Blight» disease in corn, it has been established that the climatic parameters and disease probabilities in the last 3 hours along with the current climatic parameters should be taken into account for the most accurate predictions. The random forest model with hyperparameters max_depth=5 and n_estimators=10 has achieved the best metric score on the testing set. Future studies on this topic would consider recurrent neural network (RNN) architectures to address the problem of crop disease prediction based on climatic data.

REFERENCES

1. Agricultural production statistics 2000 – 2021. URL: <https://www.fao.org/3/cc3751en/cc3751en.pdf> (Accessed on 30.11.2023).
2. FAOSTAT: Crop and livestock products. URL: <https://www.fao.org/faostat/en/#data/QCL> (Accessed on 30.11.2023).

3. Fenu G., Mallocci F. M. An application of machine learning technique in forecasting crop disease. ACM International Conference Proceeding Series. 2019. No. June P. 76-82. DOI: 10.1145/3372454.3372474.
4. Xiao, Q., Li, W., Kai, Y., et al. Occurrence prediction of pests and diseases in cotton on the basis of weather factors by long short term memory network. BMC Bioinformatics. 2019. Vol. 20, No. Suppl 25. P. 1–15. DOI: 10.1186/s12859-019-3262-y.
5. Patil, R. R., Kumar, S., Rani, R. Comparison of Artificial Intelligence Algorithms in Plant Disease Prediction. Revue d'Intelligence Artificielle. 2022. Vol. 36, No. 2. P. 185–193. DOI: 10.18280/ria.360202.

УДК 519.8

С.А.Ус¹, О.Ю.Хархула¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСЛІДЖЕННЯ ТА РАНЖУВАННЯ АЛЬТЕРНАТИВ ПЛАСТИКОВОЇ УПАКОВКИ

Анотація. Проведено дослідження наявних альтернатив пластиковому пакуванню. Розв'язано задачу багатокритеріального вибору.

Ключові слова: пластикове забруднення, пакування, вторинне використання, задача вибору, метод аналізу ієрархій

Вступ. Проблема пластикового забруднення сьогодні є дуже актуальною. Об'єми пластику, що використовуються у промисловості і споживанні людьми, дедалі зростають і немає жодних передумов вважати, що ця тенденція буде змінюватися. При цьому наслідки пластикового забруднення є досить значущими і завдають великої шкоди навколишньому як водному так і наземному середовищу. Тому для відповідального виробництва і споживання пошук альтернативних варіантів пакування і їх впровадження є актуальною задачею.

Метою даної роботи є аналіз переваг та недоліків альтернативних варіантів пакування і вибір раціонального варіанта для впровадження.

Основний зміст роботи. У доповіді «Подолання пластикової хвилі: комплексна оцінка шляхів припинення забруднення океану пластиком» [1] проведено оцінювання існуючої ситуації у сфері пластикових відходів й розглянуто кілька сценаріїв її розвитку.

Головним завданням визначено скорочування споживання пластику, інакше через 20 років щорічні обсяги його відходів з 11 млн. метричних тон зростуть до 29 млн. метричних тонн. Тобто на кожному метрі берегової лінії на земній кулі лежатиме 50 кг пластикового сміття, а в океані плаватиме приблизно 600 млн. тон таких відходів.

Для виходу із ситуації запропоновано чотири варіанта. Найоптимістичніший із них – «Зміна системи» (SCS) – скорочення виробництва

та споживання пластикової продукції. Якщо він буде реалізований, то кількість відходів, що викидаються в океан, знизиться втричі в порівнянні з нинішніми показниками. Сценарії «Збір та утилізація» (CDS), «Переробка» (RES), «Скорочення та заміна» (RSS), за розрахунками вчених, допоможуть знизити рівень забруднення довкілля пластиком наполовину.

Жодна з існуючих стратегій боротьби з пластиковим забрудненням сама по собі не може дати достатнього ефекту в його усуненні, тому комплекс заходів, що вживаються для боротьби з ним повинен бути системним і включати всі стратегії в тією чи іншою мірою.

Основні стратегії та їхні складові подано у табл. 1 та відображено на рис.1.

Таблиця 1

Стратегії боротьби із пластиковим забрудненням

№	Стратегія	Складові
1	Зменшення: 130 млн. тонн (30%)	<ul style="list-style-type: none"> – Усунення, – Повторне використання (споживач), – Повторне використання (нові моделі доставки)
2	Заміна: 71 млн. тонн (17%)	<ul style="list-style-type: none"> – Папір, – Папір з покриттям, – Матеріали що компостуються
3	Переробка: 84 млн. тонн (20%)	<ul style="list-style-type: none"> – Механічна переробка – замкнутий цикл (CL), – Механічна переробка – відкритий цикл (OL), – Хімічна переробка – пластик у пластику (P2P)
4	Утилізація: 101 млн. тонн (23%)	<ul style="list-style-type: none"> – Хімічна конверсія – пластик у паливо (P2F), – Полігон ТПВ, – Спалювання

У роботі досліджено варіанти заміни пластикового пакування іншим, а саме: багаторазовим, паперовим або пакуванням із біорозкладного пластику. До уваги взято такі критерії: екологічні наслідки, складність впровадження, вартість впровадження, швидкість окупності, перспектива окупності, бар'єрні властивості.

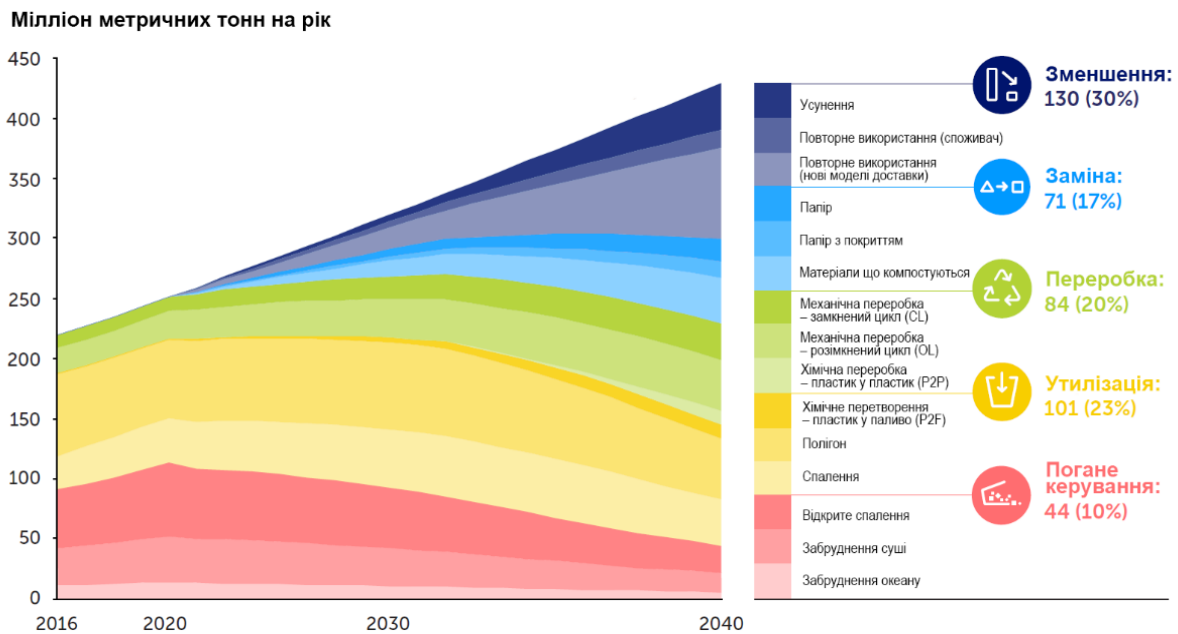


Рис. 1. Стратегія зменшення пластикового забруднення [Breaking the Plastic Wave]

Для розв’язування задачі вибору було обрано метод аналізу ієрархій [2], оскільки він дозволяє враховувати як якісні так і кількісні критерії і залучати до вибору експертів різних галузей. Побудована ієрархія проблеми показана на рис. 2. Результати порівняння критеріїв та оцінка узгодженості матриці порівнянь – у табл. 2.



Рис. 2. Ієрархія проблеми

Оцінка важливості критеріїв

	Складність впровадження	Вартість впровадження	Швидкість окупності	Перспектива окупності	Екологічні наслідки	Бар'єрні властивості	Локальний вектор пріоритетів
Складність впровадження	1	1/3	1/3	1/5	1/5	1/7	0,03
Вартість впровадження	3	1	1/3	1/3	1/7	1/7	0,05
Швидкість окупності	3	3	1	1	1/5	1/5	0,10
Перспектива окупності	5	3	1	1	1/3	1/3	0,13
Екологічні наслідки	5	7	5	3	1	1/3	0,27
Бар'єрні властивості	7	7	5	3	3	1	0,42

Як бачимо, найбільш значущим виявився критерій, який відповідає за бар'єрні функції, а найменш впливовим – складність застосування.

Після оцінки переваг альтернатив за кожним з критеріїв визначаємо їх глобальні пріоритети (табл. 3).

Таблиця 3

Розрахунок глобального пріоритету

	Складність застосування	Вартість впровадження	Швидкість окупності	Перспектива окупності	Екологічні наслідки	Бар'єрні властивості	Глобальний пріоритет
пріоритет критерію	0,03	0,05	0,10	0,13	0,27	0,42	-
Вторинне використання	0,06	0,08	0,07	0,76	0,73	0,67	0,59
Паперове пакування	0,74	0,73	0,65	0,06	0,19	0,06	0,21
Біорозкладний пластик	0,19	0,19	0,28	0,18	0,08	0,27	0,20

Отримані результати показують, що значну перевагу має вторинне використання, отже вибір багаторазового пакування є раціональним. Інші дві альтернативи незначно різняться за пріоритетами.

Наукова новизна розробки полягає у застосуванні математичних методів до дослідження альтернативних варіантів заміни пластикового пакування і вибору оптимального рішення.

Висновки. В роботі було досліджено альтернативи пластиковому пакуванню, їх переваги та недоліки і перспективи впровадження, проведено ранжування методом аналізу ієрархій.

ПЕРЕЛІК ПОСИЛАНЬ

1. PEW Charitable Trusts and SystemIQ «Breaking the Plastic Wave: a comprehensive assessment of pathways towards stopping ocean plastic pollution»
2. Трифонова О. В. Математичні моделі і методи прийняття рішень для сталого розвитку : навч. посіб. / О. В. Трифонова, Л. В. Тимошенко, С. А. Ус ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2023. – 240 с. – Режим доступу: <https://ir.nmu.org.ua/handle/123456789/165392>

УДК 004.588

С.Д. Приходченко¹, О.С. Шевцова¹, І.А. Левдик¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

НЕДОЛІКИ РОБОЧОГО ПРОЦЕСУ ТА РОБОЧОЇ ПРОДУКТИВНОСТІ З ВИКОРИСТАННЯМ ПОТОЧНИХ ЗАСОБІВ ВЕДЕННЯ ТЕХНІЧНИХ ІНСТРУКЦІЙ

Анотація. В роботі реалізовано дослідження ефективності робочого процесу з поточними методами ведення технічних інструкцій та виявлення слабких місць в існуючих рішеннях.

Ключові слова: *технічна інструкція, конструкторська документація, КРІ*

Вступ. Сучасні виробничі підприємства стикаються з необхідністю забезпечення високої продуктивності праці та мінімізації помилок у робочих процесах. З розвитком технологій у галузі віртуальної та доповненої реальності стало можливим знизити ризики та затримки в робочому процесі та переглянути традиційні методи надання – зокрема – технічних інструкцій на виробничих підприємствах.

Ця робота спрямована на проведення аналізу впливу традиційних засобів створення технічних інструкцій на продуктивність праці з використанням статистичних методів.

Основний зміст роботи. Процес інструктажу вже давно є налагодженим на багатьох великих виробництвах та має певні підходи до реалізації залежно від місця застосування.

Конструкторська документація (КД) — частина технічної документації у вигляді графічних і текстових документів, котрі в сукупності або окремо, визначають склад і будову виробу та містять необхідні дані для його розробки, виготовлення, контролю, експлуатації, ремонту і утилізації. Такий різновид інструкції є цільовим в дослідженні, але не єдиним можливим. Пропонується дослідити ефективність сьогоденних методів та підходів до ведення та використання конструкторських документацій та виявити слабкі місця в сучасних підходах для розробки нового, більш ефективного способу.

Згідно результатів опитування на тему «Оцінка якості інструктажу на робочому місці» був сформований наступний зріз зі статистичними даними щодо поточних засобів ведення інструкцій. Більшість респондентів використовують (Рис.1) усний спосіб інструктажу (87.5%), на другому місці знаходяться текстові редактори (62.5%), далі йде месенджер (50%), й лише на останньому місці – спеціальна програма (25%). Була надана опція інше, на яку припадають 12,5% відповідей.

1. Яким чином у вас проводилися інструкції щодо покрокового виконання якоїсь роботи на робочому місці?

8 ответов

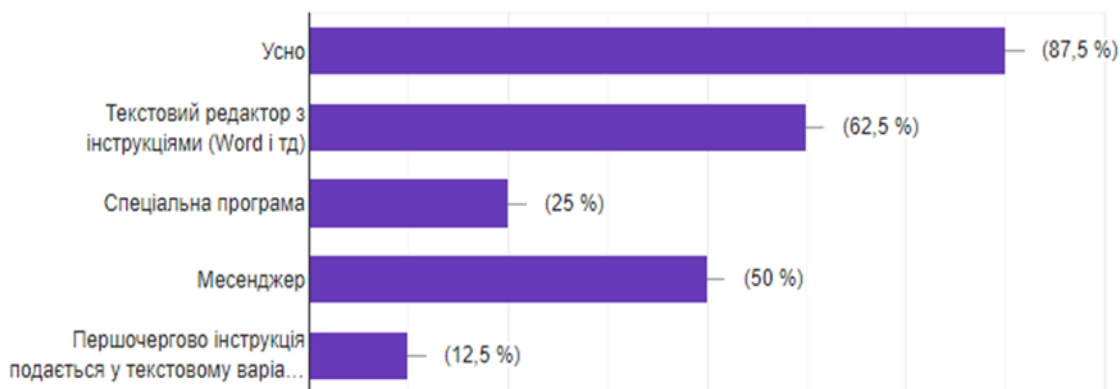


Рис.1. «Статистика використання типів ведення інструктажу»

На запитання, в якому вигляді було би зручніше надавати інструкції з великою перевагою був обраний варіант «Спеціальна програма» (Рис. 2.). На запитання про актуальність використання спеціалізованого програмного забезпечення також були надані позитивні відгуки (Рис. 3.).

5. В якому вигляді вам, як ментору, було би зручніше надавати/переглядати інструкції?

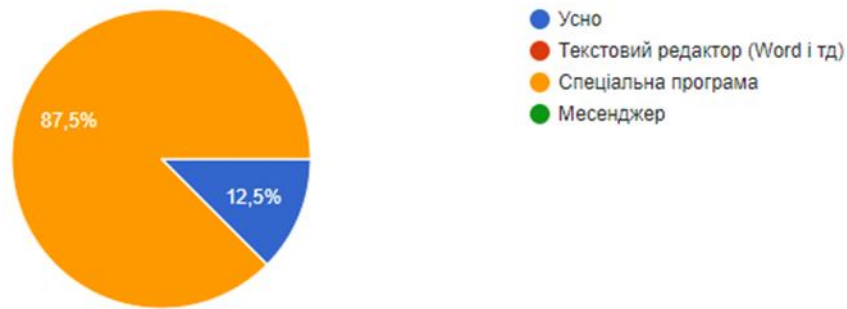


Рис. 2. «Інфографік про найбільш бажаний тип ведення інструкцій серед респондентів, що виконують менторську функцію»

4. Чи хотіли би ви мати універсальну мультимовну програму для зручного ведення інструктажних даних, можливістю їхнього розповсюдження? Якщо ні, то чому?

Так
Скоріше так
Так, це було б зручно.
Так, тому що це було б простіше з точки зору запам'ятовування
Так
Спробувати можна
Так, такий застосунок значно б полегшив й поліпшив комунікацію

Рис. 3. «Актуальність потреби в спеціалізованому програмному забезпеченні».

Таким чином була зібрана інформація про потреби співробітників. Також був поставлений ряд питань щодо недоліків в поточних способах ведення інструктажу і ось які результати були отримані:

1. Усний спосіб
 - Незручність, погана запам'ятовувальність, нема чіткості;

- Не на всі питання, що виникають є чіткі відповіді, необхідність додаткових записів обговорення, можливість щось упустити, або не розібрати до кінця;

- Не можливо запам'ятати всі інструкції, треба переписувати знову. Якщо інструкцій багато важко запам'ятати послідовність

2. Текстовий редактор

- Незручність, погана запам'ятовувальність, Нема чіткості,

- Не на всі питання, що виникають є чіткі відповіді.

3. Спеціальна програма:

- Найбільш оптимальний варіант. Але іноді не вистачає стандартизації у опису інструкцій

4. Месенджер:

- Незручність, погана запам'ятовувальність,

- Нема чіткості, можливість втрати написаних даних інструктажу, неструктурованість

5. Інше:

- Комунікація з автором завдання може потребувати час для встановлення зручного часу (наприклад через різні часові зони), витримати часу на саму комунікацію й ризик не отримати бажаної інформації

Як можна побачити, в багатьох працівників виникають проблеми з точністю та зрозумілістю інструкцій, також по зрізу можна зробити висновок, що майже ніде з опитуваних не була використана програмна система, тобто процес був не автоматизованим, що свідчить про певні затримки та ризики.

Так, наприклад, при недостатній чіткості інструкції знадобиться час для додаткового з'ясування деталей. Спосіб ведення інструкцій через текстові редактори в багатьох випадках є недостатньо зручним в плані інтерфейсу та структурованості даних. Усний спосіб передачі інформації при цьому всьому є найнадійнішим, а спосіб надання інформації через месенджер – найнебезпечнішим з точки зору конфіденційності інформації та кібербезпеки.

З точки зору бізнесу затримки і паузи в робочому процесі можуть суттєво впливати на прибуток виробництв, оцінити наявність та закономірності між затримками та використовуваними методами ведення технічної документації можна за допомогою KPI – ключових показників ефективності.

Ключові показники ефективності (англ. key performance indicators, KPI) – фінансова та нефінансова система оцінки, яка допомагає організації визначити ступінь досягнення стратегічних цілей.

Ефективність відбиває співвідношення витрат із результатами, а результативність – ступінь досягнення намічених цілей.

Виділяють чотири великі групи KPI:

1. За результатом. Показують те, що вийшло після завершення виконання того чи іншого завдання.

2. За витратами. Показують витрати на досягнення завдань.

3. За продуктивністю. Відбивають, як співвідносяться грошові чи тимчасові витрати та отриманий результат.

4. За ефективністю. Показують співвідношення показників.

Пропонується провести оцінку ефективності робочого процесу згідно системи КРІ, базуючись на метриках, пов'язаних з процесом ведення технічної документації, а саме:

1. Час створення інструкції
2. Кількість помилок і неточностей
3. Середній час пошуку необхідної інформації

Базуючись на цих метриках були проведені відповідні розрахунки, результати яких можна побачити на рис. 2:

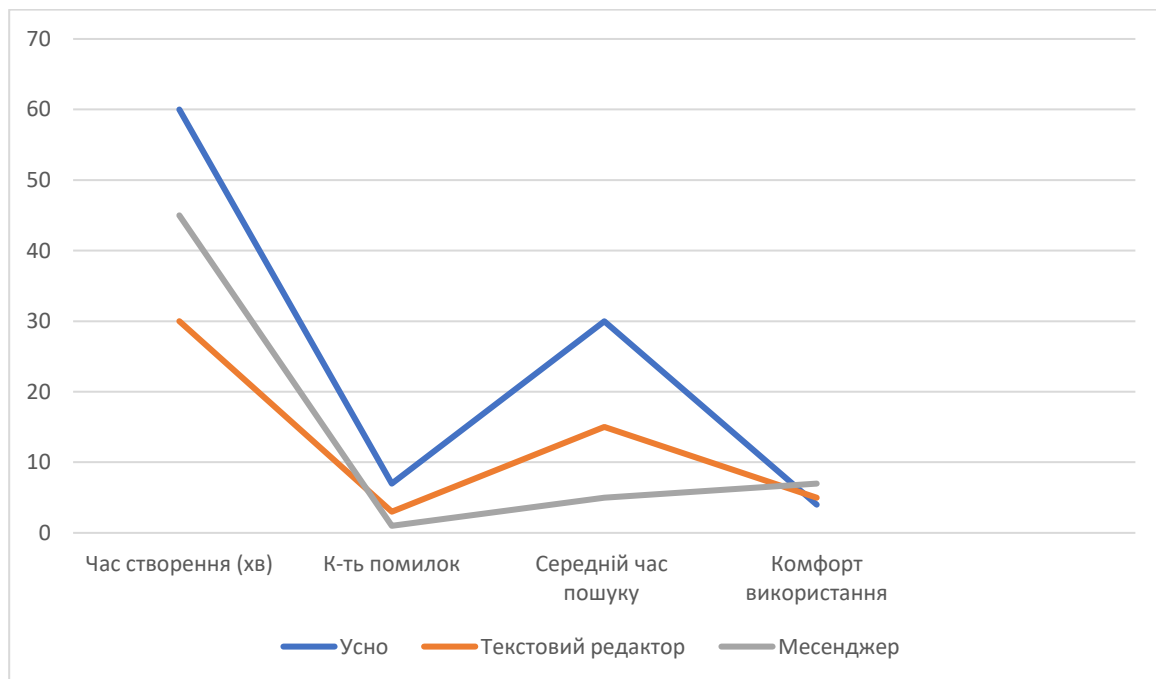


Рис.4. «Результати розрахунків згідно з метрик КРІ»

Висновки. В поточному робочому процесі є безліч підходів до передачі інформації та навчання співробітників, але в кожного з них є як переваги, так і недоліки. В ході дослідження було досліджено найрозповсюджені підходи до ведення технічних інструкцій та виявлено їхні недоліки. Також було досліджено вплив цих недоліків на продуктивність робочого процесу. Враховуючи усе вищевикладене пропонується створити універсальний програмний застосунок, який би дозволяв враховувати вищевказані недоліки та допоміг би автоматизувати процес ведення інструктажу серед співробітників підприємств.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Key Performance Indicators, KPI [Електронний ресурс]. URL: <https://www.it.ua/knowledge-base/technology-innovation/key-performance-indicators-kpi> (дата звернення: 01.11.2022).

2. СИСТЕМА КОНСТРУКТОРСЬКОЇ ДОКУМЕНТАЦІЇ [Електронний ресурс]. URL: <http://www.tsatu.edu.ua/ettp/wp-content/uploads/sites/25/3-dstu-33212003.pdf/> (дата звернення: 01.11.2022).
3. Технічна документація, 2020 [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D1%85%D0%BD%D1%96%D1%87%D0%BD%D0%B0_%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D1%96%D1%8F(дата звернення: 01.11.2022)

УДК 004.415.3:681.6

Л.І. Мещеряков¹, М.М. Одновол¹, Н.П. Уланова¹, Д.С. Ведерников¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ЕФЕКТИВНІСТЬ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РЕАЛІЗАЦІЇ МОДИФІКОВАНОГО МЕТОДУ ВІДНОВЛЕННЯ ФОРМАЛЬНИХ ГРАМАТИК

Анотація. Досліджено методи відновлення формальних граматик та розроблено модифікований метод відновлення формальних граматик. Ефективність в результаті впровадження досягається за рахунок автоматизації виводу формальних граматик з початкових послідовностей.

Ключові слова: формальні граматика, методи відновлення граматик, обробка природної мови, програмне забезпечення, автоматичний вивід граматик, теорія обчислень, генеративні алгоритми, аналіз мов, індуктивний вивід.

Вступ. В наш час, однією з ключових областей, яка знаходить своє віддзеркалення в постійному прагненні до вдосконалення і новаторства, є обробка інформації та аналіз даних. У цьому контексті, тематика досліджень методів відновлення формальних граматик набуває особливого значення. Тому, що реалізація модифікованих методів у вигляді програмного забезпечення дозволить використовувати їх на практиці для автоматичного виводу формальних граматик. Автоматизований та точний процес відновлення граматик може сприяти покращенню якості граматик, зменшити витрати, кількість помилок. Зменшення складності розробки та впровадження програмного забезпечення для відновлення граматик може зробити технології більш доступними для широкого кола розробників, включаючи початківців та розробників із різних галузей.

Такий метод може вплинути на сферу медицини, наприклад, для більш детального вивчення структур ДНК-ланцюгів, шляхом виявлення структурних особливостей та закономірностей. Допомогти науковцям формалізувати мови та алгоритми. Це особливо важливо в теорії обчислень та комп'ютерних науках, де точні та формальні специфікації мов і алгоритмів є ключовими. Розвивати нові класи граматик та досліджувати їхні теоретичні властивості. Узагальнюючи, програмне забезпечення для відновлення формальних граматик має широкий

спектр наукових застосувань у теорії обчислень та розробці програмне забезпечення, а також в різних галузях, де важлива формалізація та аналіз мов.

На жаль, наразі для входження в область формальних граматики потребує від науковця глибинних знань та експертизи в різних методах, розуміння їх особливостей та призначення.

Наразі в області формальних граматики не існує готового комплексного та доступного рішення, тому й було проведено це дослідження та розроблено програмне забезпечення для покращення ситуації в області формальних граматики.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- провести аналіз існуючих методів відновлення формальних граматики;
- визначити основні функціональні вимоги до методів та програмної системи;
- розробити модифікований метод для відновлення граматики;
- спроектувати та розробити програмну частину, шляхом аналізу та вибору найбільш підходящих технологій;
- провести порівняльний аналіз ефективності застосування алгоритму;
- зробити висновки щодо ефективності впровадження такого програмного забезпечення.

Основний зміст роботи. Під час роботи над даною роботою були розглянуті наступні методи відновлення формальних граматики:

- метод перебору;
- метод індуктивного виводу;
- метод зіставлення;
- генетичні алгоритми;
- алгоритми машинного навчання.

Метод перебору є простим, але ресурсоємким підходом до відновлення формальних граматики, що базується на систематичному переборі можливих комбінацій символів та правил підстановки. Починаючи з визначеної початкової граматики, генеруються всі можливі комбінації символів та правил. Вибираються найкращі граматики, які можуть служити для розв'язання конкретних задач. Процес генерації, оцінювання та відбору може повторюватися декілька разів для досягнення оптимального результату, включаючи випадкові зміни чи інші стратегії. Хоча метод простий, він може бути часоємним та неефективним при великій складності генерації, але використовується там, де інші методи складніше впровадити або коли точний вигляд граматики менш критичний, ніж її функціональність.

Метод індуктивного виводу відновлює формальні граматики через ітеративне покращення. Розпочинаючи з початкової граматики, аналізуються вхідні дані, і нові правила додаються та оцінюються на кожному кроці. Процес включає індуктивне виведення правил, що може базуватися на синтаксичних або частотних критеріях. Ітерації включають додавання, вилучення або модифікацію

правил. Остаточна граматики вибирається за підсумками і відповідає вимогам та показникам відновлення. Метод систематично уточнює граматику, адаптуючи її до структури вхідних даних.

Метод зіставлення відновлює формальні граматики, порівнюючи вхідні дані із зразком граматики. Аналізуються структурні елементи вхідних даних, порівнюючи їх із заданим зразком, що може бути експертно розробленим чи заздалегідь визначеним. На основі цього аналізу генерується граматики, відображаючи синтаксичні особливості зразка. Отримана граматики використовується для аналізу та відновлення нових екземплярів вхідних даних, з результатами, що валідуються та можуть бути кориговані для поліпшення точності. Метод ефективний, коли можливо визначити зразок граматики та коли структура вхідних даних пов'язана зі створенням конкретних синтаксичних конструкцій.

Метод генетичних алгоритмів використовує еволюційний підхід до відновлення формальних граматики, який використовує принципи природного відбору. Процес розпочинається зі створення початкової популяції граматики, яку оцінюють за допомогою функції придатності. Генетичні операції, такі як кросовер і мутація, використовуються для створення нових граматики у наступному поколінні. Відбір граматики для наступного покоління ґрунтується на їхній придатності, а цей процес повторюється протягом ітерацій. Генетичні алгоритми ефективні в оптимізації та при експлоративному пошуку, особливо коли структура граматики не є чіткою.

Метод машинного навчання для відновлення формальних граматики використовує алгоритми та моделі, які автоматично вивчають структуру граматики на основі великої кількості навчальних даних. Процес включає збір та підготовку навчальних даних, вибір моделі машинного навчання, векторизацію даних, тренування моделі, тестування та валідацію, а також можливий тюнінг та оптимізацію моделі для покращення результатів на конкретних завданнях відновлення граматики. Метод машинного навчання є ефективним у випадках, коли граматики складно визначається чітко, але може бути вивчена з великої кількості прикладів.

Задача полягає в розробці програмного забезпечення для відновлення формальних граматики із послідовності лексем. Програма повинна задовольняти наступні вимоги до функціональних характеристик:

- вхідні дані повинні вводитися з файлу. Програма повинна коректно читати та обробляти вхідні дані зазначеного формату;
- програма повинна мати можливість ефективно обробляти файли великого розміру, забезпечуючи швидку та оптимізовану роботу навіть у випадку великої кількості лексем;
- збереження результатів у файл, після відновлення граматики програма повинна забезпечувати можливість збереження результатів у файл. Це дозволяє користувачам зручно використовувати та аналізувати відновлені граматики на подальших етапах роботи;

- програма повинна функціонувати у режимі автоматичного відновлення граматики. Це означає, що вона самостійно аналізує вхідні дані та генерує відновлену граматику без втручання користувача.

Ці вимоги описують основу для розробки програмного забезпечення, яке забезпечить користувачам зручний та ефективний інструмент для відновлення формальних граматики з великих файлів лексем.

Основний принцип алгоритму полягає в поступовому переході від простих до більш складних правил, об'єднувати прості правила і уникати повторень. Розглянемо та аргументуємо принцип конструювання цих правил на прикладі вхідної послідовності " $((abc)+(abc)+(abc)+(abc)+(abc)+(abc))$ ". Для цього побудуємо таблицю слідувань, яка визначить всі можливі переходи. Цей підхід дозволяє ефективно створювати всі можливі переходи, при цьому граматичні конструкції не дублюються, що є особливо ефективним при обробці великої кількості повторень фрагментів у вхідній послідовності. Почнемо з побудови алфавіту та базових правил:

Алфавіт \rightarrow (,), +, a, b, c.

Побудова базових правил:

A1 \rightarrow (

A2 \rightarrow)

A3 \rightarrow +

A4 \rightarrow a

A5 \rightarrow b

A6 \rightarrow c

Наступний крок заміна всіх терміналів на прості правила: S \rightarrow A1 A1 A4 A5 A6 A2 A3 A1 A4 A5 A6 A2 A3 A1 A4 A5 A6 A2 A3 A1 A4 A5 A6 A2 A3 A1 A4 A5 A6 A2 A2.

Далі ітераційним шляхом відбувається виведення правил. Перша ітерація: будемо можливі переходи.

Таблиця 1

Таблиця слідування

	A1	A2	A3	A4	A5	A6
A1	1			1		
A2		1	1			
A3	1					
A4					1	
A5						1
A6		1				

Формулюємо нові правила по таблиці слідування:

R1 \rightarrow A1, A1

R2 \rightarrow A1, A4

R3 → A4, A5

R4 → A5, A6

R5 → A6, A2

R6 → A2, A3

R7 → A3, A1

R8 → A2, A2

Заміна на нові правила: S → R1 R3 R5 R8 R3 R5 R8 R3 R5 R8 R3 R5 R8 R3 R5 R8 R3 R5 A2.

Видалення невикористаних правил: R2, R4, R6, R7.

Друга ітерація: будемо можливі переходи.

Таблиця 2

Таблиця слідування

	R1	R3	R5	R8	A2
R1		1			
R3			1		
R5				1	1
R8		1			
A2					

Формулюємо нові правила по таблиці слідування:

P1 → R1, R3

P2 → R3, R5

P3 → R5, R8

P4 → R5, A2

P5 → R8, R3

Заміна на нові правила: S → P1 P3 P2 P5 P3 P2 P5 P3 P2 A2. Видалення невикористаних правил: P4. Третя ітерація: будемо можливі переходи:

Таблиця 3

Таблиця слідування

	P1	P2	P3	P5	A2
P1			1		
P2				1	1
P3		1			
P5			1		
A2					

Формулюємо нові правила по таблиці слідування:

F1 → P1, P3

F2 → P2, P5

F3 → P2, A2

F4 → P3, P2

F5 → P5, P3

Заміна на нові правила: S → F1 F2 F4 F5 F3.

Кінцева граMATика S → F1 F2 F4 F5 F3 використовує посилання на більш прості правила об'єднуючи їх в більш складні.

Розглянемо приклад в якому правила будуть повторно використовуватися. Для цього представимо правила в вигляді дерева.

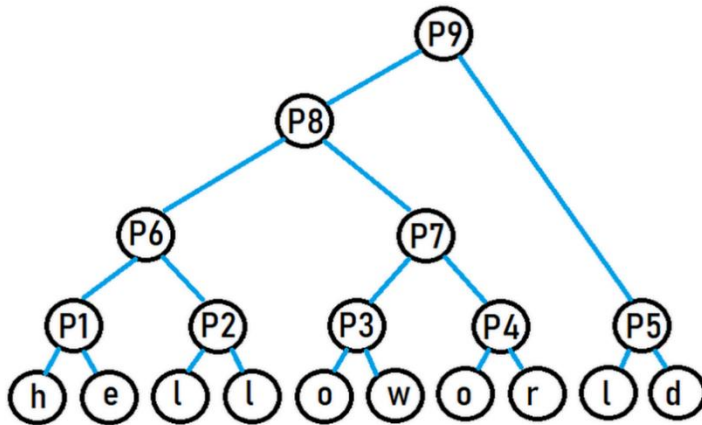


Рис.1. Приклад правила, що утворює послідовність «hello world»

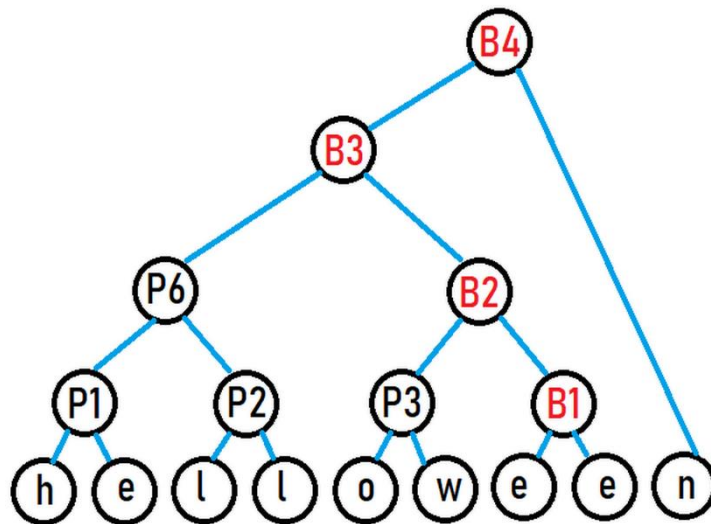


Рис. 2. Приклад правила, що утворює послідовність «helloworld»

З рисунків 1 та 2 можна відзначити, що правила P1, P2, P3 та P6 вже були використані повторно. Для включення нового слова в граMATику довелося розширити її чотирма новими правилами. Така властивість автоматично структурує та зменшує кількість правил, що може бути корисно при великій кількості правил.

Розроблений алгоритм відновлення формальних граMATик, спрямований на ефективне та систематичне створення граMATичних конструкцій на основі

простих правил. Основним принципом алгоритму є поступовий перехід від простих до складних правил, уникаючи повторень та ефективно конструюючи граматичні вирази.

Принципи розробки та проектування додатку базуються на обраній мові програмування C++, яка відповідає критеріям ефективності, розміру розроблювального додатку, та швидкості компіляції. Застосування принципів SOLID та інших принципів об'єктно-орієнтованого програмування дозволяє створити додаток, який легко підтримувати та розширювати.

Ефективність алгоритму проявляється у його ітеративному та нерекурсивному підході, що гарантує обчислювальну ефективність та передбачувану складність. Алгоритм може ефективно обробляти велику кількість повторень та створювати граматичні конструкції, підвищуючи його придатність для різноманітних застосувань у галузі обробки послідовностей та розпізнавання граматик.

Ефективність виявляється в його здатності обробляти великі об'єми даних, роблячи його корисним для різноманітних завдань обробки генетичних та мовних послідовностей. Такий підхід робить розроблений алгоритм практичним інструментом для відновлення формальних граматик у різних областях, таких як комп'ютерні науки та обробка природної мови.

Таким чином, розроблений алгоритм виявляється потужним інструментом для автоматизованого відновлення формальних граматик, що сприяє подальшим дослідженням у сфері комп'ютерних наук та обробки природної мови, а також розвитку наукових спільнот та широкого кола розробників.

Наукова новизна розробки полягає в аналізі та модифікації існуючих методів враховуючи сучасні наукові та технічні досягнення в області обробки мови та автоматичного виводу граматик.

Висновки. В результаті виконання роботи було розглянуто ключові аспекти відновлення формальних граматик та представлено різні методи, призначені для цієї задачі. Зазначено важливість формальних граматик у теорії формальних мов та їх широкий спектр застосувань у комп'ютерних науках. Досягнуті в цьому дослідженні результати свідчать про значущість вивчення та удосконалення методів відновлення формальних граматик. Взаємодія між різними підходами та використання різноманітних технік може сприяти покращенню ефективності процесу відновлення та розширенню областей їхнього застосування.

ПЕРЕЛІК ПОСИЛАНЬ

1. Automata and Formal Languages: A Practical Approach, Susan Reynolds. - O'Reilly Media, 2021. – 289с.
2. Computational Linguistics and Formal Languages: An Integrated Approach, Emma Bennett. - Pearson, 2021. – 287с.
3. Computational Linguistics and Formal Languages: An Integrated Approach, Emma Bennett. - Pearson, 2021. – 287с.
4. Elements of the Theory of Computation, Harry R. Lewis, Christos H. Papadimitriou. - Kogan Page, 2020. – 296с.

5. Formal Grammar and Automata Theory: With Applications in Software Engineering, Henry Martinez. - Springer, 2017. – 334с.
6. Formal Methods in Linguistics: An Introduction, Ryan Brooks. - Pearson, 2020. – 241с.
7. Introduction to Automata Theory, Formal Languages, and Computation, Laura Turner. - Wiley, 2017. – 276с.
8. Machine Learning for Decision Makers: Cognitive Computing Fundamentals for Better Decision Making, David Wilson. - O'Reilly Media, 2018. – 280с.
9. Natural Language Processing: Applications and Techniques, Amanda Wells. - Packt Publishing, 2022. – 398с.
10. Practical Guide to Natural Language Processing: Learn how to apply NLP to real-world projects using Python, Rachel Turner. - O'Reilly Media, 2021. – 359с.

УДК 004: 378.14

С.Д. Приходченко¹, О.В. Удовик¹, С.М. Мацюк¹, Є.В. Йолкін¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ОПТИМІЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ З ВИКОРИСТАННЯМ ХМАРНИХ СХОВИЩ

Анотація. В роботі проведено аналіз впливу використання хмарних сховищ на оптимізацію різноманітних бізнес-процесів та підвищення їх продуктивності.

Ключові слова: *хмарне сховище, хмарні обчислення, web-додаток, безпека даних, бізнес-процеси, зберігання даних, корпоративні системи, хмарна інфраструктура.*

Вступ. Сучасні підприємства в умовах стрімкого технологічного розвитку гостро потребують удосконалення та оптимізації своїх процесів. Використання хмарних сховищ є однією з потужних і перспективних технологій, яка має значний вплив на різні сфери бізнесу. Технологія забезпечує ефективне та безпечне зберігання, обробку й обмін даними, а також відкриває безмежні можливості для оптимізації бізнес-процесів.

Інтерес до даної теми пов'язаний з практичною необхідністю впровадження ефективних інструментів підвищення продуктивності та зниження витрат для бізнесу. Аналіз і дослідження показують, що хмарне сховище справді може бути ключовим елементом у досягненні цих цілей. Досягнення завдань дослідження дозволить компаніям оптимізувати свою діяльність і підвищити конкурентоспроможність на світовому ринку.

Підприємства завжди прагнуть до розвитку свої процесів та бізнесу внаслідок нестабільного навколишнього середовища. Кожна організація намагається володіти групою ресурсів і можливостей. Хмарні обчислення дозволяють клієнтам орендувати ІТ-інфраструктуру, платформу та програмні послуги в хмарі, коли це необхідно. Таким чином, клієнти хмарних обчислень можуть використовувати свої бізнес-додатки, зберігати дані та проводити аналіз через Інтернет на основі принципу "плати за використання". [0]

Основний зміст роботи. Хмарні сховища - це технологія, яка використовує Інтернет (централізовані віддалені сервери) для підтримки та обробки даних і додатків для передбачуваних користувачів, будь то приватні особи або компанії, оскільки вони надаються їм на вимогу.

Хмарні сховища є однією з найбільш суперечливих тем в наш час, оскільки кожен прагне знайти всеосяжне значення для цього вільного терміну. Це бачення майбутнього, і навіть якщо воно несе в собі певну неоднозначність технології, її безпечність і корисність змушують глобальні компанії та установи конкурувати за її використання, оскільки її переваги численні і нескінченні. Наразі глобальні компанії в галузі інформаційних технологій (IT), такі як Microsoft, Google, і понад 31 мільйон розробників та інвесторів по всьому світу розробляють хмарні технології і створюють віртуальне середовище для приватних осіб та установ, щоб значною мірою використовувати можливості гнучкого, безпечного, інтерактивного технологічного середовища. [0]

На початку сімдесятих років минулого століття концепція ефективності змінилася і стала відомою, зосередившись на здатності організації досягати своїх цілей виживання за частковими стандартами, такими як: моральний дух працівників, рівень нещасних випадків або виробничих травм, рівень відсутності на роботі та рівень плинності кадрів, серед інших понять. Для побудови організаційної ефективності будь-якої організації вона повинна бути інноваційною, що є надзвичайно важливим для успіху організації. Більше того, організаційна ефективність вважається важливою основою для досягнення організаційних цілей. Організації стикаються зі зростаючою глобальною конкуренцією, і у них немає іншого вибору, окрім як бути більш ефективними. Системи планування ресурсів підприємства забезпечують рішення, надаючи відділам краще розуміння і більшу прозорість усіх бізнес-процесів, які відбуваються в організації, так що вони стають незамінним фундаментом в організаціях взагалі. Організація завжди прагне досягти своїх цілей, незалежно від того, чи є ці цілі прибутком або певними темпами зростання, і час від часу спонукає організації вимірювати свою ефективність та результативність у досягненні цих цілей, і це для того, щоб підвищити організаційну ефективність діяльності. Складність пошуку фіксованих і точних показників для визначення стандарту або моделі, на які можна покластися при оцінці ефективності та результативності організації, враховуючи, що кожна установа має свою власну реальність, а також складність вимірювання ступеня застосування професійної етики всіма сторонами організації.

Для визначення рівня використання хмарних обчислень та його впливу на досягнення організаційної ефективності буде використано описово-аналітичний метод. Вибірка дослідження складалася з усіх адміністраторів та працівників телекомунікаційних компаній (Orange, Zain та Umniah). Вибірка дослідження складалася з 321 працівника, серед яких було розповсюджено анкети. Було отримано 251 анкету, тоді як 14 анкет, які не підходили для статистичного аналізу, були виключені, таким чином, кількість анкет, придатних для статистичного аналізу, становила 237. [0]

Інструмент Кронбаха-Альфа (Cronbach's Alpha) є статистичною метрикою, яка використовується для оцінки внутрішньої узгодженості питань у опитувальниках і шкалах. Цей коефіцієнт є одним з найбільш поширених способів вимірювання надійності опитувальників і призначений для оцінки того, наскільки всі елементи всередині шкали вимірюють той самий конструкт або властивість. Рівняння стабільності інструменту Кронбаха-альфа було застосовано до всіх областей дослідження та інструменту в цілому, це показано в таблиці 1. Кронбах-Альфа набуває значень в інтервалі від 0 до 1. Чим ближче значення до 1, тим вище внутрішня узгодженість шкали. Зазвичай вважається, що значення Кронбаха-Альфа вище 0.70 сигналізують про прийнятний ступінь надійності, хоча конкретні стандарти можуть змінюватись в залежності від контексту дослідження. [0] Формула для розрахунку Кронбаха-Альфа являє собою відношення суми дисперсії всіх елементів шкалі до суми загальної дисперсії та дисперсії, виявленої випадковими помилками. Формула виглядає таким чином(1)

$$\alpha = \frac{k}{k-1} \left(1 - \frac{\sum_{i=1}^k \sigma_i^2}{\sigma_T^2} \right) \quad (1)$$

- k - кількість елементів (питань) у шкалі,
- σ_i^2 - дисперсія відповідей кожного елемента,
- σ_T^2 - загальна дисперсія відповідей.
-

Таблиця 1

Коефіцієнт Кронбаха-альфа

Сфера	Ca
Хмарні обчислення	0.80
Організаційна ефективність	0.84

З таблиці 1 видно, що коефіцієнт стабільності для сфери хмарних обчислень склав 0,80, а організаційна ефективність – 0,84, що є високими та прийнятними значеннями для цілей дослідження.

Було отримано коефіцієнт кореляції застосування хмарних обчислень на організаційну ефективність, а також застосовано множинну регресію для виявлення впливу застосування хмарних обчислень на організаційну ефективність, і результати представлені в таблиці 2.

Коефіцієнт кореляції впливу застосування хмарних обчислень на ефективність організації

Організаційна ефективність	Хмарні обчислення	
	Кореляція	Значимість
Досягнення цілей	0.73	0.00
Продуктивність	0.71	0.00
Адаптування до робочого середовища	0.75	0.00

Таблиця 2 показує, що коефіцієнти кореляції між застосуванням хмарних обчислень та організаційною ефективністю коливаються в межах 0,71-0,75, що є позитивним зв'язком, який свідчить про позитивну кореляцію між застосуванням хмарних обчислень та організаційною ефективністю.

Висновки. У підсумку проведено аналіз впливу використання хмарних сховищ на оптимізацію різноманітних бізнес-процесів було виявлено, що використання хмарних обчислень чинить статистично значущий вплив на досягнення цілей, підвищення продуктивності та адаптацію до бізнес-середовища. [0]

Ґрунтуючись на реаліях компаній сфери телекомунікацій, бачимо що потенціалі хмарних обчислень у поліпшенні комунікаційних процесів усередині та поза організацією, сприяють координації та співпраці на різних рівнях адміністративної структури.

ПЕРЕЛІК ПОСИЛАНЬ

1. Zou S.-H., Fang N.-S., Gao W.-J. Research on online cloud storage technology. 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Xuzhou, 16–19 October 2020. [Електронний ресурс]. URL: <https://doi.org/10.1109/dcabes50732.2020.00025> (дата звернення: 11.11.2023).
2. Gilbert J. Cloud Native Development Patterns and Best Practices: Practical architectural patterns for building modern, distributed cloud-native systems. Packt Publishing, 2018. 316 с.
3. Weinman J. Cloudeconomics: The Business Value of Cloud Computing. Wiley & Sons, Incorporated, John, 2012. 416 с.
4. Bogataj Habjan K., Pucihar A. The Importance of Business Model Factors for Cloud Computing Adoption: Role of Previous Experiences. Organizacija. 2017. Т. 50, № 3. С. 255–272. [Електронний ресурс]. URL: <https://doi.org/10.1515/orga-2017-0013> (дата звернення: 12.11.2023).
5. Kassner E., Briggs B. Enterprise Cloud Strategy. Microsoft Press, 2016. 156с.

ЗАСТОСУВАННЯ МОДЕЛЕЙ ЛАТЕНТНОЇ ДИФУЗІЇ В КОНТЕКСТІ ГЕНЕРАТИВНОГО МИСТЕЦТВА

Анотація. Розглянуто теоретичні аспекти використання генеративних моделей, заснованих на дифузії, та їх практичне застосування у галузі генеративного мистецтва, шляхом виконання експериментів та аналізу їх результатів. Особлива увага приділяється оптимальним параметрам генерації для досягнення високої якості генерованих зображень.

Ключові слова: дифузійні моделі, машинне навчання, генеративне мистецтво, штучний інтелект, глибоке навчання, параметри генерації, генерація візуального контенту, *Sampling methods, Stable Diffusion, Latent Diffusion models, CLIP, U-Net, VAE, ControlNet.*

Вступ. У сучасному світі генеративне мистецтво, підсилене передовими технологіями штучного інтелекту, визначає новий етап творчого вираження та взаємодії з мистецтвом. Використання моделей, заснованих на дифузії є однією з інноваційних стратегій у цьому контексті, надаючи можливості для створення унікальних та високоякісних зображень. Однак, враховуючи теоретичні аспекти цього методу та його потенційні застосування у галузі генеративного мистецтва, важливо систематизувати знання та дослідити параметризацію методу для оптимальних результатів. Ця робота присвячена аналізу та експериментуванню з використанням моделей, заснованих на латентній дифузії в контексті генеративного мистецтва.

Таким чином, ця робота розкриває теоретичні засади та параметри методу роботи з моделями, заснованими на дифузії в контексті їх застосування у галузі генеративного мистецтва, виокремлюючи ключові аспекти, які визначають ефективність та виразність даного підходу, що дозволяють досягти найвищої якості та естетичності згенерованих зображень.

Постановка задачі. Для досягнення цілей даної роботи визначено та розгорнуто наступні завдання:

- теоретичний аналіз використання генеративних дифузійних моделей: провести докладний аналіз теоретичних принципів та основ генерації зображень із використанням моделей, заснованих на дифузії та порівняти їх із іншими методами генерації зображень;
- експериментальне дослідження параметрів методу та їх впливу на процес генерації: визначити ключові параметри генерації, провести серію експериментів, проаналізувати їх результати та визначаючи вплив різних параметрів генерації на вихідні зображення;
- розробка рекомендацій для оптимізації генерації: розробити рекомендації щодо оптимальних параметрів генерації із використанням

дифузійних моделей для різних умов та завдань генеративного мистецтва, сформувавши практичні поради щодо використання методу у реальних проектах.

Основний зміст роботи. Під час виконання даної роботи були використані наступні методи та інструменти:

- метод теоретичного аналізу;
- метод експериментального дослідження;
- метод оцінки результатів людськими експертами;
- генеративна модель латентної дифузії Stable Diffusion v1.5 від Stability AI;
- Stable Diffusion від Automatic1111 – браузерний інтерфейс, що базується на Gradio (бібліотеці для побудови інтерактивних інтерфейсів для моделей машинного навчання) для роботи із моделями латентної дифузії.

Метод використання генеративних моделей, заснованих на дифузії - це алгоритм генеративного навчання, який працює шляхом формування початкового випадкового шуму та його подальшого поступового очищення до тих пір, поки він не перетвориться на дані, які схожі на ті, що були в тренувальному наборі даних, на якому навчено модель. У контексті генеративного мистецтва моделі, засновані на дифузії можна використовувати для створення нових зображень, які не існують у реальному світі. Наприклад, алгоритм можна використовувати для створення фотореалістичних зображень, портретів людей, пейзажів або навпаки неіснуючих об'єктів та абстрактних творів мистецтва. На відміну від класичних моделей, заснованих на дифузії, які під час генерації зображення виконують обчислення у піксельному просторі зображення, моделі латентної дифузії виконують обчислення у латентному просторі, працюючи з представленням зображення меншої розмірності. Дане архітектурне нововведення дозволило значно скоротити витрати на навчання та зробило можливим масове використання даних моделей навіть серед користувачів із скромною по сьогоднішнім стандартам апаратною характеристикою.

Слід зазначити, що моделі латентної дифузії не є монолітними, а складаються з трьох основних компонентів:

VAE – нейронна модель, яка містить енкодер (конвертує вхідні дані у їх латентне представлення меншої розмірності) та декодер (реконструює інформацію у латентному представленні назад у її оригінальну форму);

U-Net – нейронна модель, яка у складі дифузійних моделей виконує дві ключові функції, які визначаються під час її навчання: прогнозування шуму у вхідному наборі даних та асоціація вхідних представлень зображень із представленням їх тестового опису, результатом навчання є модель, що здатна за рахунок очищення шуму із вхідних даних генерувати дані, подібні до тих, які були використані під час навчання;

CLIP – нейронна модель, яку було треновано на наборах зображення-текстовий опис для генерації представлень зображень та текстового опису, які

мали б високий процент збіжності, забезпечуючи семантичну відповідність між представленням та описом.

Розглянемо алгоритм генерування зображення на основі тестового опису (text-to-image) із використанням моделі латентної дифузії. Першим кроком у генеруванні зображення на основі текстового опису є створення випадкового тензора у прихованому просторі (latent space). Цей тензор містить шум, який буде використано для створення зображення, він уявляє собою Гаусівський шум (випадковий шум, який має нормальний розподіл). Приклад візуалізації початкового шуму виглядає наступним чином (рис. 1).



Рис. 1. Приклад візуалізації початкового шуму

На наступному кроці Text Encoder мовної моделі CLIP перетворює вхідний текстовий опис (Prompt) у латентний вектор, який представляє семантичне значення кожного слова/токену у запиті (кожному токену відповідає окремий вектор). Даний вектор буде об'єднано із представленням шуму та інтегровано у латентний простір для керування моделлю під час генерації, щоб забезпечити відповідність генерованого зображення текстовому опису. Тепер, отримане об'єднане латентне представлення використовується у якості вхідних даних для нейронної моделі U-Net, яка «прогнозує» або «оцінює» шум у вхідному наборі та поступово видаляє його. Тобто, входом для даної мережі є зашумлений латентний простір, а виходом – шум у даному латентному просторі. Візуалізація кроків очищення представлення генерованого зображення від шуму виглядає наступним чином (рис. 2).



Рис. 2. Візуалізація кроків очищення представлення зображення від шуму

Метод, що використовується під час видалення шуму називається методом вибірки (sampling method), оскільки на кожному кроці виконується створення нового зображення-зразка (sample). Є множина доступних методів вибірки, кожен з яких у тій чи іншій мірі впливає на процес видалення шуму та фінальне зображення. Кожен метод описується математичною формулою, яка визначає як саме буде очищено шум (частка від загального обсягу, скільки попередніх кроків обробки буде враховано, чи слід додавати новий випадковий шум із метою підвищення варіативності вихідного зображення). Приклад впливу різних методів очищення прогнозованого шуму виглядає наступним чином (рис. 3).



Рис. 3. Результати використання різних методів очищення прогнозованого шуму

Після того як оцінений шум було видалено з зображення, виконується декодування латентного простору та повернення у піксельний простір (перетворення у піксельне зображення більшої роздільної здатності).

Таким чином, використання моделей латентної дифузії є одним із найпоширеніших методів генеративного навчання, який використовується для створення реалістичних зображень. Він працює шляхом поступового розм'якшення випадкового шуму до кінцевого зображення. Спрощена схема генерації зображення виглядає наступним чином (рис. 4).

Використання дифузійних моделей не є гарантовано кращим варіантом серед усіх існуючих, адже вибір методу генерування може бути обумовлений специфічними вимогами, такими, наприклад, як швидкість генерації, якість зображень, складність реалізації та вартість генерації.

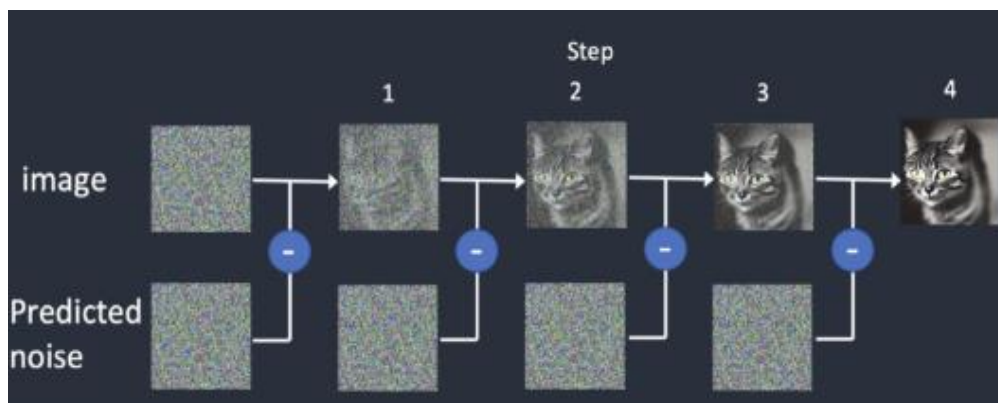


Рис. 4. Спрощена схема генерації

Однак використання моделей латентної дифузії має значну перевагу порівняно з іншими методами генерації зображень у необхідній обчислювальній потужності та задіяному обсязі відеопам'яті. Порівняння найбільш використовуваних методів генерації зображень (табл. 1).

Таблиця 1.

Порівняння методів генерації зображень

Метод	Вхід - Вихід	Переваги	Недоліки	Найкраще підходить для
Stable Diffusion	Текстовий опис - Зображення	Високоякісні зображення, гнучкість налаштування	Може повільно генерувати зображення	Генерація високоякісних зображень із текстових описів
GAN (Generative adversarial network)	Відсутній - Зображення	Дуже реалістичні зображення	Може бути складно навчати	Генерація реалістичних зображень для широкого спектра додатків
VAE (Variational autoencoder)	Дані - Нові зразки даних	Може навчати хорошу репрезентацію даних	Може бути складно навчати	Створення латентного представлення даних для таких завдань, як зменшення розмірності та виявлення аномалій
Трансформери	Дані - Зображення	Зображення високої точності	Може бути обчислювально дорогим	Генерації зображень для завдань, які потребують високої точності

Отже, можна виділити такі переваги використання моделей латентної дифузії, як: гнучкість налаштування та варіативність генерованих зображень, швидкість генерації, відкритий похідний код та, як наслідок, наявність великих онлайн-агрегаторів, які містять у відкритому доступі навчені моделі, готові та придатні до самостійного використання, це є значною перевагою, адже більшість сервісів, що виконують генерацію зображень високої якості вимагають платну

підписку, що робить їх непридатними до довгострокового практичного дослідження, а самостійне навчання моделі є занадто дорогим.

Далі розглянемо практичне використання моделей латентної дифузії для генерації зображень. Використання моделей латентної дифузії за умов оптимальних налаштувань дозволяє отримати високоякісні зображення. Для оптимального використання необхідно встановити такі параметри генерації як: Prompt (запит, тобто текстовий опис зображення, яке потрібно створити), Negative prompts (використовується для запобігання створення небажаних елементів та деталей зображень), Checkpoint (модель латентної дифузії), Sampling method (це метод, який буде використовуватися для видалення шуму під час генерації зображення), Sampling steps (кількість кроків обробки), Image size (розмір генерованого зображення), CFG scale (параметр, який контролює те, наскільки згенероване зображення зберігає баланс між відповідністю запиту користувача та «креативністю»), Seed (числове значення, що визначає шаблон початкового випадкового шуму).

Оптимальним підходом є генерація зображень розміром 512x512, або хоча б одна сторона має залишатися 512 пікселів (адже більшість доступних дифузійних моделей було треновано саме на наборах даних з даною розмірністю), та подальше використання розширення Hires.Fix, яке дозволяє користувачам на базі похідного зображення генерувати зображення більшої роздільної здатності без втрати якості. Дане розширення робить це за допомогою техніки, що називається підвищенням частоти дискретизації, яка збільшує розмір згенерованого зображення без внесення артефактів або розмиття. Приклад фрагментів зображень, згенерованих різними підходами: оригінального зображення розміром 512x512, його подальше масштабування із використанням Upscaler'a, генерація нового зображення розміром 1024x1024 та ре-генерація оригінального зображення із використанням Hires,fix виглядає наступним чином (рис. 5).

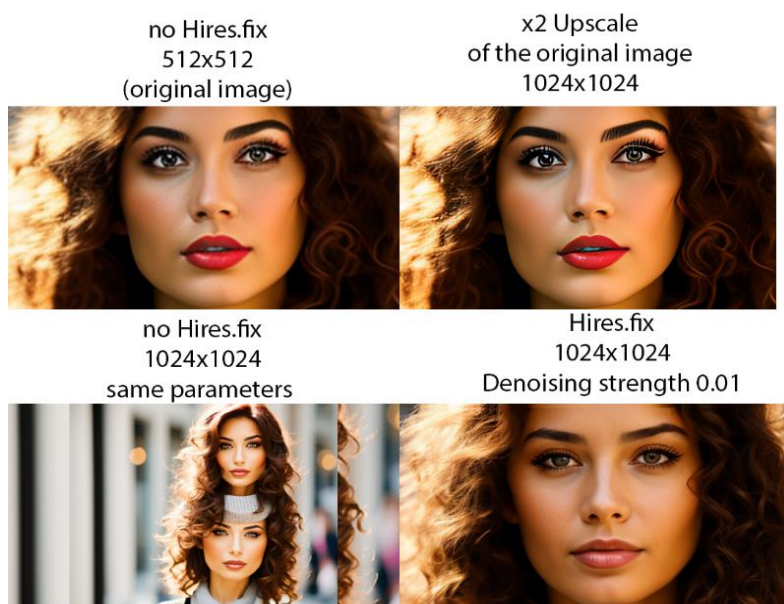


Рис. 5. Огляд фрагментів генерованих зображень

Як видно, після одного циклу використання Hires.fix зображення стає більш деталізованим та реалістичним. Однак, при генерації зображення розмірністю 1024x1024 пікселів з'являється спотворення, дане зображення не є придатним до використання.

Можна сформулювати наступні рекомендації щодо оптимального налаштування параметрів генерації при роботі із моделями дифузії:

- **Image Size:** Бажано використовувати розмір зображення 512x512 пікселів, або хоча б одна сторона має складати 512 пікселів, оскільки це відповідає наборам даних на яких тренуються моделі стабільної дифузії та забезпечує цілісність зображення;

- **CFG Scale:** Визначається еспериментально, однак значення 7 є оптимальною стартовою точкою, оскільки воно показує найбільший баланс між відповідністю текстовому запиту та «креативністю» моделі;

- **Sampling Steps:** Рекомендованими є значення у діапазоні від 20 до 30;

- **Prompt:** Використовуйте лаконічну мову для опису запиту, щоб чітко передати бажане зображення, також слід використовувати *powerful words* (ключові слова, які мали найбільшу вагу у наборі даних під час навчання моделі);

- **Negative Prompts:** Є обов'язковим для створення якісного зображення. Даний параметр містить текстовий опис зображення або його елементів, які не потрібно створювати, наприклад "blurry face, low quality, deformed limbs".

Також для розширення області можливішого використання моделей стабільної дифузії можна використовувати їх у поєднанні з ControlNet. ControlNet - це нейронна мережа, яка дозволяє користувачам керувати процесом генерації зображень з текстовими описами за допомогою моделі латентної дифузії. Вона працює шляхом додавання додаткового умовного входу до моделі латентної дифузії, який можна використовувати для керування процесом генерації зображень різними способами: керування позами та положенням об'єктів на зображенні, копіювання композиції з іншого зображення, або генерування подібного зображення.

Використання моделі латентної дифузії у поєднанні із ControlNet для створення варіацій вхідного зображення виглядає наступним чином (рис. 6).

ControlNet містить такі налаштування як:

- **ControlNet model** (визначає, яка модель ControlNet використовується для генерування зображення. Різні моделі можна використовувати для досягнення різних ефектів та точності відтворення оригінальної композиції вхідного зображення);

- **Preprocessor** (визначає, як вхідне зображення попередньо обробляється перед тим, як передати його в модель ControlNet.);

- **ControlNet Weight** (регулює вагу моделі ControlNet. Висока вага призведе до більш вираженого ефекту від моделі та більшої подібності до вхідного зображення);

- ControlMode (регулює баланс між вагами моделі латентної дифузії та моделі ControlNet).



Рис. 6. Використання моделі латентної дифузії та нейронної мережі ControlNet

Наукова новизна. У контексті наукової новизни, дослідження моделей латентної дифузії робить вагомий внесок у розумінні сучасних підходів до реалізації генеративних моделей. В роботі надані теоретичні основи роботи моделей латентної дифузії, розкриваючи ключові принципи та технічні аспекти, що визначають ефективність використання даних моделей. Також було порівняно використання моделей латентної дифузії з іншими сучасними методами генеративного мистецтва, приділивши увагу конкурентним перевагам та обмеженням використання моделей латентної дифузії.

Додатково було проведено експерименти та досліджено практичне застосування моделей латентної дифузії для: генерації зображень із текстового опису, покращення генерованих зображень та створення варіацій вхідних зображень.

Висновки. В результаті проведеного дослідження було виявлено, що використання моделей латентної дифузії демонструє великий потенціал для використання в генеративному мистецтві. Надано теоретичний аналіз використання моделей латентної дифузії, їх порівняння із іншими наявними методами генерації. Також було зазначено важливі аспекти та рекомендації щодо оптимізації параметрів для досягнення найвищої якості та стабільності результатів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Rombach, R. (2022). High-Resolution Image Synthesis with Latent Diffusion Models. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2112.10752.pdf>
2. Ronneberger, O. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/1505.04597.pdf>
3. Zhang, L. (2023) Adding Conditional Control to Text-to-Image Diffusion Models. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/pdf/2302.05543.pdf>.
4. Yang, T. (2023) Pixel-Aware Stable Diffusion for Realistic Image Super-resolution and Personalized Stylization. [Електронний ресурс]. – Режим доступу: <https://arxiv.org/abs/2308.14469>.
5. Text-Guided 3D Synthesis with Latent Diffusion Models. [Електронний ресурс]. – Режим доступу: <https://er.ucu.edu.ua/bitstream/handle/1/3945/Kovalenko%2C%20Danylo%20-%20Text-Guided%203D%20Synthesis%20with%20Latent%20Diffusion%20Models.pdf?sequence=1&isAllowed=y>.

УДК 004.415.3:681.6

М.Г. Бердник¹, А.С. Куракіна¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ВЕБ-ДОДАТОК ОПТИМІЗАЦІЇ НАБОРУ ПРОДУКТІВ ДЛЯ ЗБАЛАНСОВАНОГО ХАРЧУВАННЯ

Анотація. У роботі побудовано математичну модель, метод, а також розроблено веб-додаток для оптимізації набору продуктів для збалансованого харчування за персональними характеристиками користувача та обраним типом дієти.

Ключові слова: дієтичне харчування, макронутрієнти, веб-додаток, Django, Python, оптимізація, продуктовий кошик, сервер, SPA, Angular.

Вступ. Актуальність дослідження полягає у необхідності надання зручного інструменту планування продуктів харчування для збалансованої дієти, контролю бюджету на їх закупівлю, а також оптимізація та пришвидшення цих процесів.

Вибір правильного раціону харчування залежить від цілей та стану організму. Так багато захворювань, наприклад, цукровий діабет, артеріальна гіпертензія, хвороби серця, для свого лікування та профілактики потребують дотримання відповідних дієт. Таким чином розуміння рекомендацій та обмежень дієти є важливим аспектом здорового харчування. Дослідження цих особливостей, їх оформлення для зручного перегляду та маніпуляцій, а також надання швидких рекомендацій щодо закупівель дозволить підвищити зручність та ефективності контролю за здоров'ям.

У реаліях нестабільної економічної ситуації обмеження бюджету може стати наріжним каменем організації збалансованого харчування, адже постає питання ефективного вибору продуктів, так щоб з найменшими тратами,

отримувати найбільшу кількість енергії та корисних речовин у потрібному співвідношенні.

Вивчення та аналіз предметної галузі дозволили зробити висновок про те, що хоча існує велика кількість додатків для розрахунку калорій та підбору здорового раціону харчування, більшість з них не враховує реалії українського ринку та можливості поєднувати планування продуктового кошику з контролем його вартості.

Так як підготовка списку закупівель відповідний потрібній дієті і плану бюджету може виявитися комплексним завданням, автоматизація цього процесу дозволить економити час, сформувати корисні харчові звички та полегшить контроль за добовим вживанням калорій та нутрієнтів.

Постановка задачі. Для реалізації поставленої мети було поставлено та вирішено наступні завдання: розробка математичної моделі, методів оптимізації та веб-додатка, що буде виконувати реалізацію описаних методів та за введеними даними пропонувати оптимальний набір продуктів для закупівель відповідно обраної дієти та особливостей користувача.

Програмний додаток має бути представлений у вигляді клієнт-серверного веб-додатку. Він має підтримувати адаптацію до різних типів пристроїв та операційних систем. Веб-додаток має працювати у всіх сучасних браузерах та мати зрозумілий, зручний інтерфейс. Виконання обчислень має виконуватися на боці серверу з введенням та виведенням результатів на боці клієнта.

Програмний додаток має пропонувати наступний функціонал:

- заповнення вхідних даних для обчислення: обмеження бюджету, період для якого треба виконати обчислення;
- введення даних користувача: статі, віку, ваги, зросту, рівню фізичної активності;
- розрахунок добової рекомендованої норми калорій за введеними даними;
- вибір бажаної дієти та додаткових обмежень щодо споживання продуктів;
- налаштування списку продуктів, на основі яких будуть виконуватися розрахунки;
- виконання моделювання продуктового кошику, що відповідає обраному типу дієти та параметрах;
- вивід результатів моделювання у вигляді списку продуктів та статистичної інформації щодо співвідношення макронутрієнтів та енергетичної цінності продуктового кошику;
- експорт та завантаження результатів моделювання;
- адміністрування програмного додатку: управління списками продуктів, обмежень та дієт, управління списками користувачів.

Основний зміст роботи. Перед формування математичної моделі генерації списку продуктів для закупівель потрібно провести обчислення та підготовку вхідних даних, а саме:

- розрахунок добової норми калорій людини;
- визначення балансу макронутрієнтів відносно обраної дієти;
- підготовка списку продуктів та обмежень.

Розрахунок добової норми калорій користувача проводиться за наступними значеннями: статтю, віком, вагою, зростом та рівнем фізичної активності людини. Для розрахунку рівню основного обміну скористаємося формулою Міффіліна–Сан-Жеора. Для чоловіків розрахунок буде мати наступний вигляд, представлений у формулі (1).

$$\text{BMR} = 10 \cdot W + 6.25 \cdot H - 5 \cdot Y + 5, \quad (1)$$

де BMR – рівень основного метаболізму, ккал;

W – маса людини, кг;

H – зріст людини, см;

Y – вік людини, рік.

Для жінок формула Міффіліна–Сан-Жеора має різницю у значенні останньої константи, як показано у формулі (2).

$$\text{BMR} = 10 \cdot W + 6.25 \cdot H - 5 \cdot Y - 161. \quad (2)$$

Для отримання добової норми калорій потрібно помножити рівень основного метаболізму на індекс фізичної активності, значення якого береться згідно рекомендацій Всесвітньої організації охорони здоров'я.

Для отримання добової норми калорій потрібно помножити рівень основного обміну на індекс фізичної активності.

Визначення мінімального та максимального значення білків, жирів та вуглеводів береться згідно рекомендацій для окремої обраної дієти.

Останнім етапом підготовки є обробка списків продуктів та підготовка їх для обчислень: визначення вартості, калорійності та вмісту макронутрієнтів на одиницю продукту. Список продуктів фільтрується за обмеженнями: виключаються продукти або їх групи, що заборонені у обраному типі дієти. Для кожного продукту прив'язуються обмеження щодо його мінімальної та максимальної дозволеної кількості.

Для побудови математичної моделі було вивчено ситуацію, коли людина вирішила дотримуватися певної дієти та має намір зібрати продуктової кошик на певний період часу t . При цьому існує обмеження на максимальну вартість кошику, що дорівнює s_{\max} .

Кількість продуктів у кошику позначимо через x_i , $i = 0, 1, \dots, n - 1$, кожен продукт має власну ціну за одиницю p_i . Загальну вартість всіх продуктів позначимо через s .

Метою моделювання є мінімізація цільової функції (3), що визначає суму всіх продуктів продуктового кошику.

$$\min_x \sum_{i=0}^{n-1} p_i x_i = s, s > 0. \quad (3)$$

Для повноцінного функціонування організму ця людина має отримувати з їжі за добу певну кількість енергії E , що була розрахована у попередньому розділі. Таким чином додаються наступні обмеження у формулі (4).

$$\left\{ \begin{array}{l} \sum_{i=0}^{n-1} e_i x_i = W \\ s \leq s_{max} \end{array} \right., \quad (4)$$

де e_i – кількість енергії на одиницю продукту, ккал;
 $W = t \cdot E = \text{const}$ – потреби в енергії за певний термін;
 E – затрати енергії на добу;
 t – період часу, діб;
 s – загальна вартість продуктів;
 s_{max} – максимальна допустима вартість продуктів.

Варто передбачити ситуацію, коли програма може обрати лише один або кілька найбільш дешевих та калорійних продуктів у якості результату. Таким чином додається обмеження на мінімальну та максимальну кількість вживання продукту у певний період. Обмеження представлено у формулі (5)

$$x_{i \min} \leq x_i \leq x_{i \max}, \quad (5)$$

де x_i – кількість продукту;
 $x_{i \max}$ – мінімальна кількість продукту за певний період;
 $x_{i \min}$ – мінімальна кількість продукту за певний період.

Кожна дієта має власне рекомендоване співвідношення макронутрієнтів, необхідний для забезпечення повноцінної роботи організму. Таким чином останніми обмеженнями моделі є обмеження на мінімальну та максимальну кількість білків, жирів та вуглеводів, що мають бути отримані у результаті розрахунків. Представимо ці обмеження у формулі (6).

$$\left\{ \begin{array}{l} pr_{min} \leq \sum_{i=0}^{n-1} x_i pr_i \leq pr_{max} \\ c_{min} \leq \sum_{i=0}^{n-1} x_i c_i \leq c_{max} \\ f_{min} \leq \sum_{i=0}^{n-1} x_i f_i \leq f_{max} \end{array} \right. \quad (6)$$

де pr_i – кількість білків на одиницю продукту;
 pr_{max} – максимальна кількість білків за заданий період;
 pr_{min} – мінімальна кількість білків за заданий період;
 c_i – кількість вуглеводів на одиницю продукту;
 c_{max} – максимальна кількість вуглеводів за заданий період;
 c_{min} – мінімальна кількість вуглеводів за заданий період;
 f_i – кількість жирів на одиницю продукту;
 f_{max} – максимальна кількість жирів за заданий період;
 f_{min} – мінімальна кількість жирів за заданий період.

Вирішення завдання полягає у знаходженні списку продуктів з мінімізацією їх загальної вартості $s \rightarrow \min$ та отриманні максимальної потрібної кількості енергії з урахуванням обмежень, що задаються у формулах (4), (5) та (6).

Для реалізації описаної математичної моделі було взято бібліотеку Python SciPy, яка розроблена для роботи з науковими алгоритмами та обчисленнями, що включає методи оптимізації, інтерполяції, лінійної алгебри, обробку зображень.

У якості метода оптимізації було обрано метод послідовного квадратичного програмування, що виконує мінімізацію цільової функції за заданими обмеженнями рівності та нерівності та індивідуальними обмеженнями аргументів.

Для реалізації серверної частини додатку було використано Django веб-фреймворк, архітектура якого дозволяє ефективно розділити шари представлення, обробки та збереження даних за рахунок реалізації шаблону Model – Template – View.

Частина Model описує шар доступу до даних: перевірка вхідних значень, обробка та взаємодія з базою даних.

Частина Template описує шар відображення даних на сторінці, або передачі до клієнта.

Частина View відповідає за бізнес-логіку додатку. Тут контролюється доступи до моделей, виконуються обчислення і формуються у відповідний шаблон. Цей шар зв'язує між собою Model та Template.

Для організації взаємодії між API сервера та клієнтом було використано Django Rest Framework, що представляє інструменти для серіалізації та

десеріалізації даних: для конвертації даних з об'єктів Python у JSON або інший формат для відправки через мережу.

Для взаємодії з базою даних Django має вбудований ORM, що дозволяє представляти таблиці у вигляді програмних класів та виконувати операції на ними через методи, а не прямі SQL-запити, що підвищує ефективність та безпечність роботи.

Для збереження даних користувачів, продуктів та розрахунків було обрано СУБД PostgreSQL. Дана СУБД підтримує принципи ACID (Atomicity, Consistency, Isolation, Durability) роботи з даними, що робить її стабільним та цілісним рішенням для систем, де важлива цілісність інформації.

Додаток включає авторизації, реєстрації, адміністрування та перевірки вхідних даних, виконання розрахунків та перегляду їх результатів.

Було розроблено та описано API запити серверу, а також механізм взаємодії між клієнтом та сервером.

Інтерфейс додатку налаштований на зручність користування з пристроїв з різним розміром екрану та операційною системою, з можливістю налаштування візуальної теми сторінок.

За введеними вхідними даними програмний додаток виконує моделювання та оптимізацію продуктового кошику та повертає список продуктів, енергетична цінність яких задовольняє потребам користувача, а баланс білків, жирів та вуглеводів знаходиться у рекомендованих межах.

Для перевірки доцільності впровадження програмної системи було проведено аналіз та порівняння результатів оптимізації з прикладами раціону відповідних дієт.

Наукова новизна полягає у розроблених моделі та методу оптимізації процесу вибору продуктів задля підвищення ефективності організації здорового харчування.

Висновки. За результатами дослідження ефективності роботи додатку було зроблено наступні висновки:

- додаток є ефективним інструментом для розрахунку продуктового кошику, що має енергетичну цінність необхідного для задоволення щоденних потреб користувача згідно його персональних характеристик та рівня фізичної активності та рекомендований обраною дієтою баланс макронутрієнтів;

- коректна конфігурація обмежень дозволяє моделювати продуктовий кошик, що задовольняє вимоги здорового балансу харчування та має мінімальну можливу вартість для заданого набору продуктів;

- так як додаток фокусується на балансі саме макронутрієнтів, тобто білків, жирів та вуглеводів, він потребує окремого налаштування для додавання в раціон більшої кількості овочів та фруктів. Для підвищення ефективності обчислень рекомендовано включити у процес розрахунку баланс також мікронутрієнтів, рекомендовані добові норми вітамінів та харчових волокон;

– так як метою є максимальне зменшення вартості продуктового кошика, результати розрахунків дозволяють ефективно планувати закупівлі при обмеженні бюджету, але пропонують менше різноманіття продуктів.

УДК 372.881.111.1

В.В. Заболотнікова¹, Н.М. Нечай¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ENHANCING PROFESSIONAL WRITING SKILLS: TEACHING STUDENTS TO CREATE EFFECTIVE PROMPTS FOR AI

Abstract. Розглянуто проблему використання генеративних мовних моделей на основі штучного інтелекту в процесі підготовки спеціалістів для різних сфер економіки. Представлено огляд деяких елементів навчання написанню ефективних завдань для моделей на основі ШІ.

Keywords: *AI tools, generative language model, prompt, professional writing.*

Introduction. In today's digital age, proficiency in professional writing extends beyond conventional forms of communication. Artificial intelligence (AI) tools have become increasingly useful in various spheres of life, industry and economy. Therefore, the ability to craft effective prompts is an essential skill for professionals in various fields. Teaching students to articulate their thoughts in a way that resonates with AI systems enhances their technological literacy and equips them with a valuable tool for the workplace.

Formulation of the problem. AI-based generative language models along with image generators and chatbots have become very popular with students and created a real problem for universities all over the world. Traditional essay assignments, presentation texts and other writing tasks can be easily produced with the help of AI-powered tools. Lots of researchers admit that “tools commonly used to detect plagiarism in the academic field are vulnerable to attacks by these AI-based tools.” [1] On the other hand, according to the data provided by Oxford University Press [4] many countries work on including courses teaching AI skills at the secondary or even primary educational levels. This paper suggests helping students perceive AI tools as assistants and sources of help for their creative process instead of illegal plagiarizing.

The main content of the work. Following the best practices of the world's leading educational institutions (like [2] and [3] to name just a few) it would be wise to incorporate the skills of creating effective prompts for AI generative models into the foreign language learning curriculum for future professionals in the spheres of IT, engineering, economics and others.

In professional settings, individuals often interact with AI for tasks ranging from drafting reports to conducting research. As AI systems, powered by natural language

processing algorithms, rely on well-structured and clear prompts to deliver accurate and relevant results, the skill of formulating effective prompts becomes crucial for efficient collaboration between humans and AI. Thus, A.J. Wallbank emphasises the need to train students “on the art of (a) constructing effective prompts, and (b) critically navigating and evaluating outputs” [5].

Before starting to practice writing for AI, it would be reasonable to educate students about the capabilities and limitations of AI systems. A solid understanding of these systems' operations is foundational for crafting prompts that yield meaningful results.

Clarity and precision should be the first point to emphasize as being important for the language of prompt writing. Ambiguous or vague prompts can lead to inaccurate AI responses. Students should learn to articulate their queries or instructions concisely. Then the awareness of the context should be focused on. Consideration of the context in which the prompts will be interpreted ensures that they are tailored to specific tasks or domains, resulting in more accurate and relevant AI-generated content.

Finally, ethical considerations of prompt writing should be discussed in class. Students should be aware of the potential biases that can be introduced through poorly constructed prompts and strive to create fair and unbiased interactions with AI systems. Good examples of the differences between biased and unbiased prompting can be found on the website of the Center for Faculty Excellence of Montana State University [3]. This leads us to the practical considerations of the process of writing AI prompts.

When teaching students how to write effective prompts for AI, it's essential to focus on specific language elements that enhance clarity, precision, and context. Here are some specific language elements to incorporate into the process of instruction:

Use of key terms. Students need to be able to include relevant keywords in their prompts, ensuring that AI systems grasp the main intent of the query. Reading exercises aimed at identifying key words and word combinations should lay the basis for this ability. Students should know the specific terminology of their professional sphere to tailor their prompts to the context of the task.

Sentence structure. Some structures can be more effective than others for producing the necessary result. For example, using interrogative words like "who," "what," "where," "when," "why," and "how" can help frame clear and specific queries. The use of question tags or qualifiers can be used to refine the scope of the prompt. It is suggested that students practice various sentence structures in their queries and analyse the results thus developing their writing, reading and critical thinking skills simultaneously.

Avoiding ambiguity. Ambiguous terms or vague instructions may lead to misinterpretation by AI systems. Therefore, it can be helpful to encourage pair work or work in small groups to allow students to check each other's writing for clarity and ease of understanding. Then they can use the results of their work to track how slight changes in wording can impact the precision of the AI's response.

Contextual Clues. Students need to learn to provide context in their prompts. This includes mentioning relevant information about the task, project, or field to guide

the AI system in delivering more accurate results. Specifying the desired format or structure for the output will help to obtain a better result as well.

Tone and style. Maintaining a respectful and professional tone when interacting with AI systems should be a focus of attention as well. The appropriate tone and style for prompt writing depends on the professional context, students should be able to adjust their language to suit formal or informal settings.

In order to maintain clarity in AI instructions it is advisable to break down complex sentences into clear and sequential steps. This helps in creating prompts that are easy for AI systems to follow. Action verbs and straightforward language should be used to convey instructions effectively. To make the prompts more precise, especially for tasks that involve data analysis or numerical outputs quantifiable metrics should be used. Students need to practice specifying quantities, percentages, or ranges to their prompts to see how this can enhance the precision of AI-generated results.

Finally, the importance of iterative revision and editing cannot be overemphasised. This includes checking for grammatical errors, ensuring clarity, and refining language for better AI comprehension.

The above-mentioned are some specific language elements and teaching practices that can be incorporated into the process of training professionals in various fields to not only develop a nuanced understanding of prompt writing for AI but also help students acquire practical skills that are directly applicable in their future professional settings.

Conclusion. It is possible and necessary to teach students how to use generative language models and other AI-based tools effectively and productively to help them use these tools to enhance their future professional activities and become competitive in the international job market.

REFERENCES:

1. Y. Xiao, S. Chatterjee and E. Gehringer, "A New Era of Plagiarism the Danger of Cheating Using AI," 2022 20th International Conference on Information Technology Based Higher Education and Training (ITHET), Antalya, Turkey, 2022, pp. 1-6, doi: 10.1109/ITHET56107.2022.10031827.
2. Effective Prompts for AI: The Essentials MIT Sloan Teaching & Learning Technologies – Massachusetts Institute of Technology [Электронный ресурс] – Режим доступа до ресурсу: <https://mitsloanedtech.mit.edu/ai/basics/effective-prompts/>
3. How To Write Effective Prompts for Generative AI Tools - Center for Faculty Excellence | Montana State University [Электронный ресурс] – Режим доступа до ресурсу: <https://www.montana.edu/facultyexcellence/teaching-advising/genai/prompts.html>
4. AI in education: where we are and what happens next - Oxford University Press (oup.com) [Электронный ресурс] – Режим доступа до ресурсу: <https://corp.oup.com/feature/ai-in-education-where-we-are-and-what-happens-next/?fbclid=IwAR2Nh1nIUGhsO0QJ-ZfteFWSZewGLy8m6MFt9F7wEtOaFBwHIW8bD4r07u4>
5. Adrian J. Wallbank. ChatGPT prompt engineering as academic skill | THE Campus Learn, Share, Connect (timeshighereducation.com) [Электронный ресурс] – Режим доступа до ресурсу: <https://www.timeshighereducation.com/campus/prompt-engineering-academic-skill-model-effective-chatgpt-interactions>

РОЗДІЛ 5

ІНФОРМАЦІЙНА БЕЗПЕКА

УДК 004.056.5(045)

Д.С. Тимофєєв¹, А.С. Совенко¹

¹Харківський національний університет радіоелектроніки, Харків, Україна

ОЦІНКА РИЗИКІВ В ПРОЦЕСІ УПРАВЛІННЯ РИЗИКАМИ КІБЕРБЕЗПЕКИ ПОВ'ЯЗАНИМИ З ЛЮДСЬКИМ ФАКТОРОМ

Анотація. Описано процес управління ризиками інформаційної та кібербезпеки. Проаналізовано ризики пов'язані з людським фактором. Виконано порівняльне дослідження застосування окремих методів управління, аналізу та оцінки ризиків.

Ключові слова: *Ризик, загроза, кібербезпека, обізнаність персоналу, зловмисник, інформаційно-комунікаційна система, факторний аналіз, поведінковий аналіз.*

Вступ. Дослідження IBM X-Force [1] показало, що час виконання атак скоротився на 94% за останні кілька років. Атаки на реалізацію яких раніше були потрібні місяці, тепер досягають успіху впродовж днів або годин. Оскільки зловмисники рухаються швидше, організації повинні застосовувати проактивний підхід до кібербезпеки, орієнтований на загрози. Другий рік поспіль головним переносником шкідливого програмного забезпечення став фішинг, виявлений у 41% випадків. Подібна статистика спостерігається стосовно інших типів загроз пов'язаних з людським фактором. Важливим аспектом протидії кіберзагрозам пов'язаним з персоналом є застосування зрілого процесу управління інформаційними ризиками з акцентом на ефективне використання відповідних методів аналізу та оцінки ризиків.

Постановка задачі:

- проаналізувати актуальні ризик-орієнтовані підходи до управління інформаційною та кібербезпекою;
- визначити перспективні методи аналізу та оцінки ризиків пов'язаних з людським фактором.

Основний зміст роботи: Впровадження процесу управління ризиками є ключовим аспектом ефективного управління. На поточний час в практиці управління інформаційною та кібербезпекою широко застосовуються понад 100 законодавчих, нормативних та договірних рамок (фреймворків), які включають в себе або потребують застосування більше тисячі різноманітних контролів або механізмів забезпечення безпеки.

Найбільш широко відомими та такими що практично використовуються в поточний час та розвиватимуться в найближчій перспективі є наступні:

– група стандартів ISO серій 31 та 27 зокрема у найбільш широкому контексті процес описано у стандарті ISO 31000, однак стосовно до процесу управління ризиками інформаційної безпеки в рамках впровадження СУІБ використовується стандарт ISO 27005 [2], який детально описує складові процесу та надає практичні рекомендації, щодо реалізації його етапів, включно з прикладами оцінки ризиків;

– фреймворк NIST Risk Management Framework [3], є одним з базових документів серії управління кібербезпекою, що дозволяє поєднати принципи управління ризиками на підприємстві Integrating Cybersecurity and Enterprise Risk Management (ERM) (NISTIR 8286) та підходи до управління NIST Cybersecurity Framework (CSF). Фреймворк посиляється на набір стандартів і рекомендацій NIST для підтримки впровадження програм управління ризиками для виконання вимог Федерального закону США про модернізацію інформаційної безпеки (FISMA), включаючи вибір, впровадження, оцінку та постійний моніторинг. Широко використовується як державним так і приватним бізнесом, зокрема останнім часом і в Україні з урахуванням адаптації національної нормативно-правової бази до світових практик;

– ENISA's Risk Management/Risk Assessment (RM/RA) Framework [4], фреймворк який активно доповнюється останнім часом значною кількістю документів та практичних шаблонів придатних до адаптивного застосування бізнес організаціями та державними установами країн ЄС від малого та середнього до транснаціонального масштабу.

Безумовно не менш відомими є COSO ERM Framework, CIS Critical Security Controls; COBIT 2019 (Control Objectives for Information Technology) – IT-стандарт і фреймворк з кібербезпеки, розроблений ISACA; BSI Standard 200-3: Risk Analysis based on IT Grundschrift; OCTAVE FORTE Model for Enterprise Risk Management; FAIR; HITRUST CSF – комплексний фреймворк з організації безпеки, створений для забезпечення конфіденційності на основі різних ризиків та стандартів, зокрема HIPAA, GDPR, PCI-DSS; ряд національних комплексних документів Австралії, Нової Зеландії, Саудівської Аравії, Франції та інших використання яких обмежується рядом факторів характерних для певних груп цих документів, таких як секторальна специфіка застосування, національна спрямованість, та інш.

Характерною ознакою більшості вищеперерахованих фреймворків є відсутність вимог до застосування певного єдиного методу аналізу чи оцінки ризиків, як правило вибір конкретного методу чи групи методів покладається на виконавців процесу, або осіб що приймають рішення. Окрім того аналіз та оцінка проводяться на різних рівнях деталізації та у декілька ітерацій.

Антропогенні, або обумовлені діями суб'єкта, загрози являють собою найбільш широку групу загроз у відомих таксономіях та класифікаціях. Традиційно вони поділяються на внутрішні та зовнішні по відношенню до об'єкту захисту, на навмисні (таргетовані і нетаргетовані) та ненавмисні, які спричинені халатністю суб'єкта або браком його кваліфікації. Розглядаючи ризик інформаційної або кібербезпеки як ймовірність впливу певної загрози або

групи загроз через вразливості на відповідний актив, слід відзначити наявність людського фактору, як джерела загроз, так і пов'язаними з суб'єктивною природою вразливостями.

Сучасні методи побудови систем управління інформаційною безпекою та систем кіберзахисту потребують нових підходів до забезпечення безпеки та сталого функціонування в умовах динамічного оновлення спектру та методів реалізації кіберзагроз, розробці та комплексних організаційно-технічних рішень. Обов'язковою є наявність узгоджених контролей спрямованих не тільки на програмно-технічний захист інформаційно-комунікаційних систем та технологій, а й на запобігання впливу методів соціальної інженерії, підвищення рівня культури інформаційної безпеки та кібергігієни організації. Системний підхід до загальної проблеми забезпечення безпеки вимагає створення та постійної підтримки моделі системи комплексної оцінки рівня культури інформаційної безпеки персоналу як важливого компонента загальної організації інформаційної безпеки і стосується окремих підприємств, установ та організацій.

Достатньо ефективним є комбінування в процесі аналізу та оцінки ризиків пов'язаних з людським фактором застосування кількісного методу факторного аналізу ризиків FAIR (Factor Analysis of Information Risk)[5] - методології, яка класифікує і систематизує пов'язані з ризиками фактори і їх взаємний вплив, з урахуванням ймовірності для частоти і величини збиткових подій з методами управління людськими ризиками Human Risk Management (HRM) заснованими на поведінковому аналізі [6,7]. Процес оцінки ризиків методом FAIR здійснюється на наступних рівнях:

- Оцінка ризиків на рівні організації: визначає діапазон інсайдерських загроз, з якими стикається організація, та визначає їх пріоритетність з точки зору їх ймовірності та впливу.

- Оцінка ризиків на рівні групи: вимагає оцінки, які групи працівників мають найбільший доступ до ключових активів і, отже, найбільшу можливість реалізовувати загрози, на рівні організації. Цей рівень може виявити групи, які були залучені в області роботи або дійсно мали доступ до активу в першу чергу.

- Рольова (індивідуальна) оцінка ризику: це факультативний рівень, який не завжди буде необхідним для кожної організації. На цьому рівні може бути виявлено ролі з підвищеним ризиком, які вимагають детальної перевірки. Процес є ресурсоемним і вимагає залучення персоналу, який добре знає ролі і характеристики рівнів наданого доступу.

Поведінковий аналіз, в свою чергу, базується на оцінці множини поведінкових факторів суб'єкта та дозволяє оцінити ступінь зрілості поведінки усвідомлення кібербезпеки, для окремого суб'єкта, груп співробітників та організації в цілому. Прикладом є використання моделі знань–ставлення–поведінки (КАВ), яка лежить в основі НАIS-Q [8], яка оцінює фактори три групи факторів: індивідуальні, організаційні та фактори впливу. На основі моделі КАВ передбачається, що в міру того, як знання працівника про безпечну поведінку в сфері інформаційної безпеки на робочому місці збільшуються, їх ставлення

покращується, що призводить до покращення поведінки в інформаційній безпеці.

Висновки: на поточний момент неможливо визначити єдиний підхід до управління, аналізу або оцінки ризиків інформаційної та кібербезпеки. Динамічність та багатофакторність процесу забезпечення кібербезпеки з урахуванням людського фактора вимагає поєднання комплексу методів та підходів заснованих, як на статистично-ймовірнісному так і на поведінковому аналізі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Повний звіт The IBM Security X-Force Threat Intelligence Index 2023 (Електрон. Ресурс) / Спосіб доступу: URL: <https://www.ibm.com/reports/threat-intelligence#>
2. ISO/IEC 27005:2022(en), Information security, cybersecurity, and privacy protection — Guidance on managing information security risks.
3. National Institute of Standards and Technology Special Publication 800-37, Revision 2, 183 pages (December 2018) – (Електрон. Ресурс) / Спосіб доступу: URL: <https://doi.org/10.6028/NIST.SP.800-37r2>
4. ENISA (2021) Interoperable EU Risk Management Framework. (Електрон. Ресурс) / Спосіб доступу: URL: <https://www.enisa.europa.eu/publications/interoperable-eu-risk-management-framework>
5. Freund J., Jones J. Measuring and managing information risk. A FAIR approach: Jack Freund, Jack Jones. – Oxford: Butterworth of Elsevier, 2017. - 391 с.
6. Wang, K.; Guo, X.; Yang, D. Research on the Effectiveness of Cyber Security Awareness in ICS Risk Assessment Frameworks. Electronics 2022, 11, 1659. <https://doi.org/10.3390/electronics11101659>
7. Human risk management: The definitive guide. – 2023. (Електрон. Ресурс) / Спосіб доступу: URL: <https://www.cybsafe.com/value/human-risk-management/>
8. Kathryn Parsons, Dragana Calic, Malcolm Pattinson, Marcus Butavicius, Agata McCormac, Tara Zwaans, The Human Aspects of Information Security Questionnaire (HAIS-Q): Two further validation studies, Computers & Security, Volume 66, 2017, Pages 40-51, ISSN 0167-4048, (Електрон. Ресурс) / Спосіб доступу: URL: <https://doi.org/10.1016/j.cose.2017.01.004>.

КЛАСИФІКАТОРИ МЕРЕЖЕВИХ АТАК

Анотація. Наведено загальна характеристика та порівняння класифікаторів мережесих атак. Результати порівняння можуть бути застосовані для вибору класифікаторів при розробці методів та засобів забезпечення кібербезпеки, а також вирішення інших задач з кібербезпеки.

Ключові слова: кібербезпека, класифікація, мережесві атаки, засоби захисту, таксономія, виявлення атак, аналіз ризиків.

Вступ. Насьогодні спостерігається швидке зростання мережесих атак як за кількістю та масштабами, так і за ступенем небезпеки. Атаки стають все більш витонченими і мають здатність поширюватися з великою швидкістю. Крім того, на тлі розвитку нових інформаційних технологій, з'являються нові типи атак. Тому важливо вдосконалювати методи та засоби, які необхідні для виявлення, класифікації, захисту та протидії різним типам атак. Окремо можна виділити задачу аналізу загроз та ризиків, які можуть виникати внаслідок реалізації мережесих атак.

Класифікація мережесих атак є неодмінною умовою для вироблення чіткого розуміння атак. Існує достатньо велика кількість класифікаторів мережесих атак, які схожі між собою, проте базуються на різних таксономіях атак. Всі вони мають свої переваги та недоліки, які треба враховувати при виборі відповідної класифікації мережесих атак.

Основний зміст роботи. Класифікація мережесих атак - це процес групування мережесих атак у певні підгрупи. Класифікація мережесих атак необхідна для можливості чіткого розуміння механізмів реалізації атак, аналізу можливих наслідків, а також визначення схожих типів атак у майбутньому.

Класифікація мережесих атак повинна відповідати певним вимогам [1,2]:

- **Прийнятність.** Таксономія повинна бути розроблена таким чином, щоб вона стала загальноприйнятою та ґрунтуватися на попередніх роботах, які є загальноновизнаними.
- **Зрозумілість.** Класифікація повинна бути легкою для розуміння тими, хто працює у сфері комп'ютерних мереж, безпеки або суміжних галузях.
- **Повнота.** Для того, щоб класифікація була повною, всі мережесві атаки повинні бути включені в цю класифікацію і мати певну категорію.
- **Взаємовиключення.** Ця вимога відносить кожену загрозу до одного класу.
- **Повторюваність.** Класифікація повинна бути повторюваною.
- **Однозначність.** Групування має бути чітко визначене таким чином, щоб не виникало жодних сумнівів щодо того, до якої категорії слід віднести мережесву атаку.

- Корисність. Корисна класифікація може бути використана в сфері комп'ютерних мереж, кібербезпеки або в інших суміжних сферах.

Одна з перших класифікацій мережних атак була запропонована Пітером Меллом (Peter Mell) [3]. По суті, це спрощена класифікація, яка відображає найбільш типові атаки:

- Віддалене проникнення (remote penetration). Атаки, які дають змогу реалізувати віддалене керування комп'ютером через мережу.

- Локальне проникнення (local penetration). Атаки, що призводять до отримання несанкціонованого доступу до вузлів, на яких вони ініційовані.

- Віддалена відмова в обслуговуванні (remote denial of service). Атаки, що дають можливість порушити функціонування системи або перенавантажити комп'ютер через мережу (зокрема, через Інтернет).

- Локальна відмова в обслуговуванні (local denial of service). Атаки, що дають змогу порушити функціонування системи або перенавантажити комп'ютер, на якому їх ініційовано.

- Сканування мережі (network scanning). Аналіз топології мережі та активних сервісів, доступних для атаки.

- Використання сканерів уразливостей (vulnerability scanning). Сканери вразливостей призначені для пошуку вразливостей на локальному або віддаленому комп'ютері.

- Злам паролів (password cracking). Для цього використовують програмні засоби, що підбирають паролі користувачів.

- Пасивне прослуховування мережі (sniffing). Пасивна атака, спрямована на розкриття конфіденційних даних, зокрема ідентифікаторів і паролів доступу.

Перші чотири класи атак розрізняють переважно за кінцевим результатом (або метою реалізації), а решта — за способом їх здійснення.

Така класифікація не дає змоги визначати елементи мережі, схильні до впливу тієї чи іншої атаки, а також наслідки, до яких може призвести успішна реалізація атак.

Говард [4] пропонує таксономію інцидентів, яка класифікує атаки за подіями, тобто атакою, спрямованою на певну ціль, що має призвести до зміни стану. Подія включає в себе дію та ціль. Цей підхід дозволяє описати всі кроки, які охоплює атака, і те, як вона розвивається. Атака складається з п'яти частин, які зловмисник виконує для досягнення несанкціонованого результату. Цими кроками є: інструменти, вразливість, дія, ціль і несанкціонований результат.

- інструмент - це механізм, який використовується для здійснення атаки;

- вразливість - це тип експлойту, який використовується для здійснення атаки;

- дія - це метод, використаний зловмисником для здійснення атаки;

- ціль - це намір, який зловмисник намагається реалізувати;

- несанкціонований результат - це зміна стану, спричинена атакою.

Такий підхід забезпечує достатньо інформативний опис атак, але недостатньо деталізований.

Гансман і Хант [5] запропонували таксономію з чотирма унікальними вимірами, які забезпечують цілісну класифікацію, що охоплює мережеві та комп'ютерні атаки. Їх таксономія забезпечує узгодженість у формулюваннях опису атак. Перший вимір - вектор атаки - використовується для класифікації атаки. Другий вимір класифікує ціль атаки. Третій вимір складається з класифікаційного номера вразливості або критеріїв з таксономії Говарда. Четвертий вимір визначає корисне навантаження або наслідки, до яких призводить атака. У межах кожного виміру надаються різні рівні інформації для надання деталей атаки. Ця таксономія не дозволяє виконувати класифікацію змішаних атак, а також не визначає вразливості.

Іншу таксономію під назвою VERDICT (Validation Exposure Randomness Deallocation

Improper Conditions Taxonomy) запропонував Daniel Lough [6]. Замість деревоподібної таксономії тут використовуються чотири характеристики атак:

- неправильна перевірка;
- неправильна вразливість;
- неправильна випадковість;
- неправильний розподіл.

В цій таксономії відсутня класифікація за типом атаки та її важко застосовувати для нових атак.

Більш повна таксономія кібератак має назву AVOIDIT (Attack Vector, Operational Impact, Defense, Information Impact, and Target) [7]. В цій таксономії використовується п'ять основних класифікаторів для характеристики атаки:

- за вектором атаки;
- за метою атаки;
- за оперативним впливом;
- за інформаційним впливом;
- за захистом.

Ця п'ята категорія, класифікація за захистом, може використовуватися для пом'якшення або усунення атаки. Ця таксономія ефективно класифікує змішані атаки та має прикладний характер.

В класичних джерелах, мережеві атаки класифікують за наступними ознаками [8]:

- характер впливу (пасивні; активні);
- мета впливу (порушення конфіденційності інформації або ресурсів системи; порушення цілісності інформації; порушення працездатності (доступності);
- умова початку здійснення впливу (віддалений вплив; напад за запитом від об'єкта; напад за настанням очікуваної події на об'єкті, що атакується; безумовний напад);

- наявність зворотного зв'язку з об'єктом, що атакується (зі зворотним зв'язком; без зворотного зв'язку);
- розташування суб'єкта атаки щодо об'єкта, що атакується (внутрішньо-сегментні; міжсегментні);
- рівень еталонної моделі ISO/OSI, за яким здійснюється вплив.

Така класифікація дуже узагальнено описує мету атаки та можливі наслідки.

Одною з альтернативних класифікацій мережних атак є класифікації на основі послідовних запитань [9]. Сутність цієї класифікації побудована на послідовних запитаннях: "Хто", "Де", "Як" і "Що". До одного типу атак можна віднести атаки, які мають однаковий тип зловмисників (Хто), однакові місця, де були розпочаті атаки (Де), використання схожих інструментів для атаки (Як), а також ступінь і тип впливу атаки (Що).

До переліку зловмисників відносяться:

- Joke – здійснює мережеву атаку в першу чергу для навчання та/або самоствердження.
- White-hat hackers – здійснюють мережеву атаку з метою з'ясування вразливостей мережі, яку атакують, і повідомляють про це мережевому адміністратору.
- Black-hat hackers – здійснюють мережеву атаку, використовуючи певні вразливості мережі та пошкоджуючи або викрадаючи інформацію з атакованої мережі.
- Little sisters – організації або компанії, які здійснюють атаки на мережі конкурентів з метою отримання фінансової вигоди.
- Big brothers – уряди або організації, пов'язані з урядом.

Місця, де були ініційовані атаки, поділяють:

- на основі хоста, коли атака запускається з комп'ютера або будь-якого пристрою, який має мережеве підключення;
- на основі мережі, коли атака може бути запущена з декількох пристроїв, з'єднаних між собою.

За масштабом атаки визначають:

- Об'єктні - об'єктом атаки є окремий об'єкт у реальному житті, який має підключення до мережі.
- Хост-орієнтовані - ціль атаки знаходиться на комп'ютерному терміналі.
- Локальні сегментні - ціль атаки знаходиться в сегменті мережі, який має багато хостів, з'єднаних між собою.
- Сегментно-орієнтовані - цей тип намагається атакувати ядро глобальної мережі.
- На основі бездротової мережі - ціль атаки знаходиться в мобільній мережі.

Інструменти для атаки визначаються:

- Платформою: Software, Hardware, Embedded hardware, Mobile.

- Каналами, які використовуються: Legacy network equipment ports, Undefined network equipment ports, Virtualization channel, User-to-network channel, Network-to-network channel.

При визначенні типу атаки аналізують параметри, які можуть мати аномальні значення: активність системи, обсяг трафіку, запити.

Відстежуючи процес мережевої атаки від початку до кінця, цей підхід дозволяє проаналізувати ланцюг здійснення атаки, але теж є достатньо узагальненим.

Висновки. Із врахуванням сучасного рівня атак, застосування на практиці класифікатора Пітера Меллома є недоцільним. При аналізі загроз та ризиків доцільно використовувати класифікації на основі послідовних запитань, AVOIDIT та Гансмана. Крім цього, враховуючи наявність в AVOIDIT розділу із рекомендаціями щодо захисту, цей класифікатор можна використовувати персоналу із експлуатації інформаційно-комунікаційних систем. Для розробки засобів захисту, за думкою авторів, більш підходить класифікатор Говарда та той, що описаний в класичних джерелах.

Слід зазначити, що поява нових атак призводить до зниження ефективності застосування наявних класифікацій, тому їх використання без внесення змін не є можливим. Саме тому перспективними є бази зі структурованим описом мережних атак, які постійно оновлюються.

ПЕРЕЛІК ПОСИЛАНЬ

1. Lindqvist, U., & Jonsson, E. (1997, May). How to systematically classify computer security intrusions. In Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on (pp. 154-163). IEEE.
2. Bisbey, R., & Hollingworth, D. (1978). Protection analysis: Final report. ISI/SR-78-13, Information Sciences Inst, 3.
3. Грайворонський М. В., Новіков О. М. Безпека інформаційно-комунікаційних систем. — К.: Видавнича група BHV, 2009. 608 с.: іл.
4. Howard, John D. and Longstaff, Thomas A. “A Common Language for Computer Security Incidents,” Technical report, Sandia National Laboratories, 1998.
5. Hansman, S., Hunt R., “A taxonomy of network and computer attacks”. Computer and Security (2005).
6. Lough, Daniel. “A Taxonomy of Computer Attacks with Applications to Wireless Networks,” PhD thesis, Virginia Polytechnic Institute and State University, 2001.
7. Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., & Wu, Q. (2009). AVOIDIT: A cyber attack taxonomy. In Proc. of 9th Annual Symposium On Information Assurance-ASIA (Vol. 14).
8. Бурячок В. Л. Технології забезпечення безпеки мережевої інфраструктури: підручник. – К.: КУБГ, 2019. 218 с.
9. Md Mehedi Hassan Onik, Nasr Al-Zaben, Hung Phan Hoo and Chul-Soo Kim, “A Novel Approach for Network Attack Classification Based on Sequential Questions”, Annals of Emerging Technologies in Computing (AETiC), pp. 1-14, Vol. 2, No. 2, 1st April 2018.

МЕТОДИ МАШИННОГО НАВЧАННЯ В СИСТЕМАХ ВИЯВЛЕННЯ ВТОРГНЕНЬ

Анотація. Розглянуто використання методів машинного навчання в системах виявлення вторгнень інформаційних систем. Представлено тенденції в використанні глибокого навчання для підвищення якості сучасних систем виявлення вторгнень.

Ключові слова: система виявлення вторгнень, машинне навчання, аномалії трафіку, глибоке навчання, нейронна мережа, machine learning, intrusion detection IDS, ANN, CNN, anomaly detection, cybersecurity.

Вступ. Захист цифрових даних від кібератак є важливою проблемою, оскільки суспільство стає цифровим. Ці дані можуть включати операційні дані, персональну інформацію або стратегічну інформацію. Вони генеруються, зберігаються і передаються як інформаційних системах організацій, підприємств, бізнесу а також і на персональних пристроях. Зокрема, кіберзахист набув особливої актуальності зі зростанням дистанційної роботи.

Постановка задачі. Для загальної класифікації методів машинного навчання в системах виявлення вторгнень необхідно виконати наступні кроки:

- розглянути, які типи систем виявлення вторгнень існують;
- визначити які методи використовують для ідентифікації вторгнень;
- розглянути загальні тенденції використання методів виявлення вторгнень;
- скласти загальну класифікацію методів виявлення вторгнень;
- зробити висновки з використання сучасних методів виявлення аномального стану в системах виявлення вторгнень.

Основний зміст роботи. Сучасні інформаційні системи є привабливою мішенню для кіберзлочинців через брак технічних знань користувачів, незахищеність пристроїв Інтернету речей (IoT), неадекватні конфігурації, погане впровадження засобів контролю та високу цінність цифрових активів. Постійно виявляються нові вразливості та нові загрози. Щодня нові кібератаки (атаки нульового дня) використовують ці слабкі місця, завдаючи величезної потенційної шкоди.

Системи виявлення вторгнень (IDS) які базуються на пошуку аномального стану є перспективним підходом до виявлення невідомих кібератак.. Ці системи працюють, встановлюючи базову лінію нормальної поведінки, і вони безперервно проводять моніторинг мережі або проміжних та кінцевих пристроїв, щоб позначити підозрілі відхилення. Через мінливу поведінку користувачів і кібератак найскладнішим аспектом є підтримка моделей в актуальному стані [2].

Сучасні інформаційні системи передбачають багато взаємодій між користувачем і пристроями. Крім того, гетерогенність цих систем ускладнює створення систем виявлення вторгнень. Поведінка користувачів може змінюватися з часом, а моделі пристроїв, системи зв'язку, протоколи можуть значно відрізнятись. Тому необхідно мати системи захисту, які можуть адаптуватися до середовища і виявляти як старі, так і нові кібератаки, не вимагаючи оновлення моделі.

IDS системи можуть мати різні конфігурації, архітектури та стратегії виявлення кібератак. IDS можна класифікувати відповідно до їхніх стратегій виявлення, стратегій розміщення та стратегій перевірки. Крім того, деякі показники оцінки необхідні для підтвердження продуктивності, точності та доцільності впровадження систем.

Те, як IDS намагається виявити кібератаку, залежить від обраної стратегії. Нижче описані ті, що широко використовуються в літературі [1,3]:

- На основі сигнатур: також називається IDS на основі правил або зловживань. Ці IDS мають базу даних, в якій зберігаються шаблони і сигнатури відомих атак. Якщо спостережувана поведінка відповідає одному з цих правил, відбувається виявлення. Цей тип IDS потребує постійного оновлення.

- На основі аномалій: використовує моделі, статистику або правила для прогнозування аномальної поведінки. Моделі або правила базуються на евристичних методах або методах штучного інтелекту. Ці методи класифікують активність між нормальною та аномальною відповідно до певного порогового значення.

- На основі специфікацій: схожий на підхід з виявленням аномалій. Однак він базується на специфікаціях, розроблених вручну, і фіксує легітимну поведінку системи. Він виявляє нерегулярну поведінку за межами визначених правил. Ці специфікації можуть бути створені на основі політик безпеки.

- Гібридна IDS: поєднує системи виявлення вторгнень (IDS) на основі сигнатур, аномалій і специфікацій для підвищення продуктивності.

У відповідності до особливостей інфраструктури, загальних вимог та вимог з кібербезпеки стратегії впровадження системи виявлення вторгнень (IDS) поділяють на наступні категорії: централізовані, розподілені та гібридні.

Для оцінки ефективності моделі IDS використовують наступні метрики: точність, True Positive Rate (TPR), коефіцієнт хибнопозитивних спрацьовувань (FPR), коефіцієнт хибнонегативних спрацьовувань (FNR), True Negative Rate (TNR) та F-міра точності.

Аномалії - це приклади даних, які не відповідають очікуваній поведінці. Згідно з [4], аномалії або викиди - це "суттєві відхилення від норми". Аномалії можуть з'являтися в даних з кількох причин, таких як зловмисна діяльність або помилка, наприклад, шахрайство з кредитними картками, хакерські вторгнення та системні збої. Існує три типи аномалій [6]:

- точкова аномалія - виникає, коли лише один екземпляр даних вважається аномалією від решти даних;

- контекстуальні аномалії - виникають, коли дані є аномальними в певному контексті, але не в іншому контексті;
- колективні аномалії - виникають, коли пов'язаний екземпляр даних є аномальним для набору даних.

Виявлення аномалій можна розділити на три категорії відповідно до наявності мічених даних. До цих трьох категорій відносяться[5]:

- контрольоване виявлення аномалій: у нас є набір даних, позначених як нормальні та аномальні. На основі цих даних будується модель, яка прогнозує, чи є новий екземпляр нормальним або аномалією;
- напівконтрольоване виявлення аномалій: до генерації моделі включаються лише екземпляри нормальних класів. Нові зразки, які не можуть бути класифіковані як нормальні, є аномалією;
- неконтрольоване виявлення аномалій: для побудови моделі класифікації не потрібні навчальні мічені дані.

У літературі описано декілька таксономій методів виявлення аномалій [5,8-11]. На основі попередніх таксономій (ми об'єднуємо їх) пропонуємо до використання наступну загальну таксономію з наступними категоріями:

- на основі класифікації - методи, які створюють модель (класифікатор) з набору маркованих екземплярів даних (навчання), а потім класифікують тестовий екземпляр, використовуючи навчену модель (тест);
- комбіновані учні - використовується через зважені комбінації декількох методів, як правило, класифікаторів;
- на основі кластеризації - методи, які групують об'єкти в кластери таким чином, щоб вони були більш схожими, ніж об'єкти в інших кластерах.
- на основі теорії інформації - методи, які аналізують інформацію набору даних за допомогою інформаційно-теоретичних вимірювань, таких як складність, ентропія та відносна ентропія.
- на основі правила найближчого сусіда - методи, які припускають, що типові приклади даних трапляються в щільних районах, а аномалії - далеко від найближчих сусідів;
- на основі м'яких обчислень - методи, для яких часто не існує точних рішень, такі як генетичні алгоритми, штучні нейронні мережі, нечіткі множини, грубі множини, алгоритми мурашиних колоній та штучні імунні системи;
- спектральні - методи, які відображають високорозмірні дані в низькорозмірне представлення.
- статистичні - методи, засновані на припущенні, що нормальні екземпляри даних трапляються у високоімовірних областях стохастичної моделі, а аномалії - у низькоімовірних областях стохастичної моделі.

Для класифікації моделей машинного навчання в [7] Гупта та ін пропонують використовувати таксономію, яка включає класичні методи машинного навчання з навчанням під наглядом та ансамблеве навчання в минулих тенденціях. По-друге, сучасні тенденції включають

напівконтрольоване та неконтрольоване навчання. Нарешті, майбутні тренди включають глибоке навчання (DL) та навчання з підкріпленням.

Виходячи з еволюції використання цих методів за останні кілька років для отримання найкращих результатів ансамблеве навчання є найефективнішим і найпоширенішим методом або стратегією [1]. Серед методів ансамблевого навчання особливо виділяється випадковий ліс. З іншого боку, DL - це ще один набір методів, який виділяється. Найпоширенішими алгоритмами в цій категорії є згортова нейронна мережа (CNN) та автокодер. Інші категорії - це керовані методи машинного навчання на основі дерева рішень (DT), нейронні мережі (ANN) та метод k-найближчих сусідів.

Висновки. За останні роки найкращі результати з ефективності демонстрували методи ансамблевого навчання та DL. Крім того, за останні кілька років з'явилося багато різноманітних методів для покращення результатів виявлення аномалій. При впровадженні цих методів у реальному середовищі необхідно враховувати, що їх потрібно перенавчати на даних з конкретного середовища. Крім того, маркування набору даних не є можливим для контрольованих і напівконтрольованих методів. Тому необхідно шукати стратегії для впровадження цих систем за допомогою методів, які адаптуються до середовища та його змін.

ПЕРЕЛІК ПОСИЛАНЬ

1. Araya J. I. I., Rifà-Pous H. Anomaly-based cyberattacks detection for smart homes: A systematic literature review //Internet of Things. – 2023. – С. 100792.
2. R. Singh, H. Kumar, R.K. Singla, R.R. Ketti, Internet attacks and intrusion detection system: A review of the literature, Online Inform. Rev. 41 (2) (2017) 171–184, <http://dx.doi.org/10.1108/OIR-12-2015-0394>.
3. M.F. Elrawy, A.I. Awad, H.F.A. Hamed, Intrusion detection systems for IoT-based smart environments: A survey, J. Cloud Comput. 7 (1) (2018) 21, <http://dx.doi.org/10.1186/s13677-018-0123-6>.
4. K.G. Mehrotra, C.K. Mohan, H. Huang, Anomaly Detection Principles and Algorithms, Vol. 1, Springer, 2017.
5. V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 15:1–15:58, <http://dx.doi.org/10.1145/1541880.1541882>.
6. A.B. Nassif, M.A. Talib, Q. Nasir, F.M. Dakalbab, Machine learning for anomaly detection: A systematic review, Ieee Access 9 (2021) 78658–78700, Publisher: IEEE.
7. R. Gupta, S. Tanwar, S. Tyagi, N. Kumar, Machine learning models for secure data analytics: A taxonomy and threat model, Comput. Commun. 153 (2020) 406–440, Publisher: Elsevier.
8. A. Patcha, J.-M. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, Comput. Netw. 51 (12) (2007) 3448–3470, <http://dx.doi.org/10.1016/j.comnet.2007.02.001>, URL <https://www.sciencedirect.com/science/article/pii/S138912860700062X>.
9. M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita, Network anomaly detection: Methods, systems and tools, IEEE Commun. Surv. Tutor. 16 (1) (2014) 303–336, <http://dx.doi.org/10.1109/SURV.2013.052213.00046>, Conference Name: IEEE Communications Surveys & Tutorials.
10. M. Ahmed, A. Naser Mahmood, J. Hu, A survey of network anomaly detection techniques, J. Netw. Comput. Appl. 60 (2016) 19–31, <http://dx.doi.org/10.1016/j.jnca.2015.11.016>, URL <https://www.sciencedirect.com/science/article/pii/S1084804515002891>.

УДК 004.655

Л.В. Кабак¹, Б.І. Мороз¹, В.О. Гарбуз¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

СУЧАСНА АКТУАЛЬНІСТЬ ВИКОРИСТАННЯ ШИФРУВАННЯ ІНФОРМАЦІЇ

Анотація: На сьогодні все більше джерел інформації детально висвітлюють ростучу важливість шифрування даних через зростаючі обсяги інформації в інтернеті. Вони розгортають сучасні фактори, що приводять до поширення шифрування даних як засобу захисту конфіденційності та приватності.

Ключові слова: шифрування даних, кібербезпека, квантові технології, хмарні сервіси, IoT, законодавство про конфіденційність.

Вступ: Зростання обсягів інформації, що циркулює в інтернеті, створює необхідність у посиленні захисту даних. Цей тренд викликає розширення застосування шифрування як ефективного засобу захисту конфіденційності та приватності.

Завдяки інтенсивному обміну даними між користувачами, компаніями та різними пристроями в мережі, величезний обсяг інформації потребує посилення захисту. Шифрування дозволяє забезпечити безпеку передачі, зберігання та обробки цих даних, знижуючи ризик їх незаконного доступу чи втрати.

Захист важливих особистих даних, фінансової інформації, а також конфіденційної корпоративної інформації у мережі стає дедалі більш важливим. Тому використання шифрування як засобу захисту стає не лише актуальним, але й обов'язковим для збереження конфіденційності та приватності в цифровому світі.

Аналіз останніх досліджень: Останні дослідження в кібербезпеці привертають увагу до сучасних методів шифрування, оскільки вони виявляються ефективними у запобіганні загрозам та зловживанням з боку кіберзлочинців. Ці методи надають широкий спектр захисту даних, що важливо у сучасному інтернет-просторі, де конфіденційність і приватність даних є надзвичайно важливими[5].

Основною причиною застосування сучасних методів шифрування є підвищення кібербезпеки. Ці методи здатні запобігти атакам, забезпечуючи безпеку та конфіденційність інформації під час передачі чи зберігання. Вони працюють на різних рівнях, від захисту окремих файлів до зашифрування даних на рівні самої мережі. Іншими словами, ці методи дозволяють уникати порушення безпеки та забезпечують інші переваги, що стають невід'ємною частиною сучасної кібербезпеки.



Рис. 1. Рейтинг програм які використовуються компаніями для захисту інформації.

Основні тенденції, що підтримують шифрування:

Зростання обсягів даних: Зростання обсягів інформації, що циркулює в інтернеті, створює необхідність у посиленні захисту даних. Цей тренд викликає розширення застосування шифрування як ефективного засобу захисту конфіденційності та приватності.

Завдяки інтенсивному обміну даними між користувачами, компаніями та різними пристроями в мережі, величезний обсяг інформації потребує посилення захисту. Шифрування дозволяє забезпечити безпеку передачі, зберігання та обробки цих даних, знижуючи ризик їх незаконного доступу чи втрати.

Захист важливих особистих даних, фінансової інформації, а також конфіденційної корпоративної інформації у мережі стає дедалі більш важливим. Тому використання шифрування як засобу захисту стає не лише актуальним, але й обов'язковим для збереження конфіденційності та приватності в цифровому світі.

Чутливість до кібербезпеки: В сучасному цифровому середовищі загострена чутливість до кібератак та загроз безпеці даних. Ця тенденція викликає потребу в застосуванні передових методів шифрування для запобігання несанкціонованому доступу до інформації.

Зловмисники, що діють у кіберпросторі, стали вдосконалювати свої техніки та засоби для отримання доступу до конфіденційної інформації. У такому контексті використання захисних методів шифрування стає критично важливим, щоб унеможливити або ускладнити доступ до даних навіть у випадку їх незаконного отримання.

Сучасні шифрувальні методики надійно захищають дані шляхом перетворення їх у незрозумілі форми для незаконних користувачів, забезпечуючи тим самим конфіденційність та цілісність інформації. Враховуючи постійну загрозу кібератак, використання сучасних методів шифрування є важливою стратегією для захисту даних та запобігання їх неправомірному використанню[2].

Розвиток квантових технологій: Завдяки швидкому розвитку квантових технологій, у нас виникає потреба в удосконаленні шифрувальних методів, щоб забезпечити надійний захист конфіденційної інформації навіть у контексті квантових обчислень[4].

Загроза, яку становлять квантові обчислення для сучасного шифрування, полягає у їх здатності швидко розшифрувати складні криптографічні ключі, які використовуються в традиційних методах шифрування. Ця здатність може стати серйозною загрозою для захисту даних, що зберігаються в даний час.

Розробники шифрувальних систем працюють над реалізацією квантово-стійких методів шифрування, які стоять на варті відповіді на цю загрозу. Ці методи використовують принципи квантової механіки для створення криптографічних ключів, які стійкі навіть у випадку застосування квантових обчислень. Такий підхід дозволяє забезпечити високий рівень захисту даних, навіть перед квантовими технологіями, що підтримуються на сучасному рівні розвитку науки і техніки.

Розширення хмарних сервісів та IoT: Поширення хмарних сервісів та Інтернету Речей (IoT) призводить до значного збільшення обсягів даних, які постійно передаються та зберігаються в онлайн-середовищі. Це спричиняє підвищений попит на швидкі та надійні методи шифрування, зокрема ті, що працюють в реальному часі.

Хмарні сервіси забезпечують доступ до великих обсягів даних через мережу Інтернету, що вимагає надійного захисту від несанкціонованого доступу. Захист таких обсягів даних в реальному часі від кіберзагроз та атак є критично важливим, оскільки ці дані можуть бути чутливими та потенційно вразливими.

Що стосується Інтернету Речей (IoT), пристрої цієї мережі постійно збирають, обробляють та відправляють дані. Застосування ефективних методів шифрування в реальному часі для захисту цих даних є ключовим аспектом для забезпечення приватності та цілісності інформації, яка передається між цими пристроями та зберігається у хмарних сервісах[3].

Отже, розширення використання хмарних сервісів та Інтернету Речей підкреслює необхідність розробки та впровадження надійних, ефективних та широкомасштабних методів шифрування, спроможних працювати в режимі реального часу.

Нормативи та вимоги щодо конфіденційності: Регуляції, такі як Загальний регламент про захист персональних даних (GDPR), створюють строгі вимоги та стандарти щодо захисту особистих даних. Ці законодавчі акти накладають на компанії велику відповідальність за збереження та обробку особистої інформації своїх користувачів та клієнтів.

Дотримання нормативів GDPR вимагає від компаній впроваджувати сильні методи захисту даних, включаючи шифрування. Це відбувається через забезпечення конфіденційності, цілісності та доступності особистих даних, що зберігаються та обробляються компанією[1].

Організації, які збирають та обробляють особисту інформацію, змушені застосовувати ефективні методи шифрування для захисту даних від несанкціонованого доступу, модифікації та втрати. Це може охоплювати шифрування даних у спокої, під час передачі, а також під час зберігання. Застосування сильних шифрувальних засобів стає необхідним для відповідності законодавству щодо захисту приватності та нормативних вимог, що ставить компанії перед завданням ефективної організації процесів збереження та захисту особистих даних своїх користувачів.

Висновок: Різноманітні тенденції, такі як зріст обсягів даних, загрози кібербезпеки, розвиток квантових технологій, розширення хмарних сервісів та IoT, разом зі строгими правилами конфіденційності, спільно підкреслюють необхідність міцних протоколів шифрування, підкреслюючи його ключову роль у захисті чутливої інформації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Регламент Європейського Парламенту і Ради (ЄС) 2016/679 від 27 квітня 2016 року про захист фізичних осіб у зв'язку з опрацюванням персональних даних: https://zakon.rada.gov.ua/laws/show/984_008-16#Text
2. КІБЕРБЕЗПЕКА В ЦИФРОВОМУ ОСВІТНЬОМУ СЕРЕДОВИЩІ ЗАКЛАДІВ ПРОФЕСІЙНОЇ ОСВІТИ: <https://binpo.com.ua/wp-content/uploads/2022/12/%D0%9A%D0%91%D0%95%D0%A0%D0%91%D0%95%D0%9F%D0%95%D0%9A%D0%90.pdf>
3. Хмарні обчислення та Інтернет речей (IoT): <https://ts2.space.uk/%D1%85%D0%BC%D0%B0%D1%80%D0%BD%D1%96-%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F-%D1%82%D0%B0-%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D1%80%D0%B5%D1%87%D0%B5%D0%B9-iot/#gsc.tab=0>
4. Квантова технологія: https://uk.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%BD%D1%82%D0%BE%D0%B2%D0%B0_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D1%96%D1%8F
5. КІБЕРБЕЗПЕКА В СУЧАСНОМУ СВІТІ: АКТУАЛЬНІ ВИКЛИКИ: http://dspace.onua.edu.ua/bitstream/handle/11300/14516/%D0%9E%D0%AE%D0%90_%D0%9A%D1%96%D0%B1%D0%B5%D1%80%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D0%BA%D0%B0_%D0%B2_%D1%81%D1%83%D1%87%D0%B0%D1%81%D0%BD%D0%BE%D0%BC%D1%83_%D1%81%D0%B2%D1%96%D1%82%D1%96_%D0%B0%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D1%96_%D0%B2%D0%B8%D0%BA%D0%BB%D0%B8%D0%BA%D0%B8_2020.pdf?sequence=1&isAllowed=y

ОСОБЛИВОСТІ МАШИННОГО НАВЧАННЯ ДЛЯ СИСТЕМ ВИЯВЛЕННЯ ВТОРГНЕНЬ В МЕРЕЖАХ ІНТЕРНЕТУ РЕЧЕЙ

Анотація. Розглянуто використання методів глибокого навчання для системах виявлення вторгнень в мережах інтернету речей. Представлена покрокова оптимізація моделі виявлення аномального трафіку для використання в кінцевих пристроях інтернету речей або IoT-шлюзах.

Ключові слова: система виявлення вторгнень, інтернет речей, машинне навчання, аномалії трафіку, глибоке навчання, нейронна мережа, IoT, machine learning, intrusion detection IDS, TinyML, CNN, anomaly detection, cybersecurity.

Вступ. Інтернет речей (IoT) розширює обчислювальні і мережеві можливості, які включають не тільки комп'ютери і мобільні телефони, а й широкий спектр інших пристроїв і датчиків по всьому світу. За оцінками Statista [1], до 2025 року кількість об'єктів Інтернету речей досягне 75 мільярдів. Безпека є однією з головних проблем, яка загрожує впровадженню різних пристроїв Інтернету речей. Через різноманітну і динамічну природу IoT захист даних з пристроїв IoT є складним завданням. Побудова ефективного і надійного підходу до безпеки зараз є однією з найважливіших речей.

Основний зміст роботи. Машинне навчання, особливо алгоритми глибокого навчання, є дорогими в обчислювальному плані і часто потребують потужного апаратного прискорювача такого як графічний процесор (GPU) для роботи. Навчання може тривати від кількох годин до кількох днів. Такі програми зазвичай працюють на великих комп'ютерах і в спеціальних центрах обробки даних. IoT пристрої про те мають дуже обмежені ресурси і їх не буде вистачати для розміщення моделі, не кажучи вже про навчання. Останнім часом з'явився декілька проектів, які спрямовані на зменшення моделей машинного навчання, щоб запускати їх на пристроях з обмеженими ресурсами.

Особисту увагу привертає TinyML [2], який дозволяє запускати алгоритми машинного навчання навіть на найменших мікроконтролерах. Це підмножина прикладного машинного навчання, яка підходить для великих моделей машинного навчання та глибокого навчання для крихітних вбудованих систем, що працюють на мікроконтролерах або інших процесорах з наднизьким енергоспоживанням. Зменшення розміру моделі часто супроводжується погіршенням якості, тому замість стиснення основна увага приділяється оптимізації моделі для цільового пристрою.

В централізованих IDS усі дані з пристроїв та проміжного обладнання надсилаються зазвичай у хмару, або великі сервери для накопичення та обробки. Такий підхід має свої недоліки. Далі наведено декілька ключових [2]:

- Конфіденційність і безпека даних. У традиційному машинному навчанні пристрої Інтернету речей надсилають свої дані в хмару для обробки. Це створює серйозні проблеми з безпекою та конфіденційністю.

- Енергоспоживання. Моделі машинного навчання споживають величезну кількість енергії. Дослідницька група з Массачусетського університету підрахувала у 2019 році, що сеанси глибокого навчання моделі машинного навчання можуть генерувати до 626 155 фунтів викидів CO₂, що приблизно дорівнює викидам вуглецю від п'яти автомобілів за весь час їхнього життя.

- Пропускна здатність і затримка мережі: Для підтримки сотень і тисяч пристроїв Інтернету речей потрібна нескінченна пропускна здатність, щоб безперервно передавати їхні дані в хмару для обробки. Іншим ключовим аспектом є затримка в мережі. У повільних мережах затримка вища. Це небажано для роботи користувачів у додатках, що працюють у режимі реального часу.

TinyML розрахований на менш потужні процесори, які споживають дуже мало енергії, як правило, в межах декількох міліват. Такі процесори часто базуються на Cortex-M, мають лише кілька сотень кілобайт оперативної пам'яті (RAM), кілька мегабайт флеш-пам'яті та тактову частоту в десятки мегагерц. TinyML-додатки забезпечують низьке енергоспоживання, низьку затримку в роботі моделі машинного навчання. Вони також гарантують збереження конфіденційності користувача, оскільки дані повністю обробляються на периферійному пристрої.

На першому кроці відбувається збір даних. Точність алгоритму машинного навчання значною мірою залежить від різноманітності набору даних, на яких навчалася модель. Не гарантується, що модель машинного навчання буде працювати на тестових даних, які статистично відрізняються від навчальних.

На наступному кроці створюється модель. Весь набір даних розбивається на три частини: навчальний, валідаційний і тестовий. Модель спочатку створюється на навчальному наборі, а потім проходить внутрішню оцінку на валідаційному наборі. Ефективність на валідаційному наборі використовується як зворотний зв'язок для налаштування архітектури моделі. Ідеально навчена модель повинна демонструвати однакові результати на навчальній та перевіірочній вибірках. Далі модель оцінюється на тестовому наборі, а потім формується модель для розгортання.

На останньому кроці отримана модель оптимізується та перетворюється для використання на пристроях інтернету речей. Модель потрібно оптимізувати для цільових пристроїв. При оптимізації моделі головна мета - перетворити вихідну модель на меншу і швидшу модель з мінімальними втратами якості. Оптимізація моделі може бути досягнута різними способами, такими як квантування, вагова обрізка та вагова кластеризація.

При квантуванні ваги моделі перетворюються з 32-бітних чисел з плаваючою комою на 16-білі числа з плаваючою комою або навіть 8-білі цілі числа, що призводить до зменшення розміру моделі. Цілочисельні моделі

працюють набагато швидше. У великій нейронній мережі значна кількість параметрів має дуже незначний вплив на загальну продуктивність. Для видалення незначних параметрів з мережі використовується вагове обрізання. Обрізання негативно впливає на ефективність моделі але допомагає її стиснути. При ваговій кластеризації кількість унікальних ваг у моделі зменшується шляхом групування їх у певну кількість різних кластерів. Після оптимізації моделі її перетворюють на нативний код цільової системи.

Алгоритми оптимізації які спрямовані на стиснення базової моделі машинного навчання реалізовані у фреймворку TensorFlow [3]. TensorFlow Lite - це мобільна бібліотека для оптимізації та розгортання великих моделей TensorFlow на мобільних пристроях під управлінням Android, iOS або вбудованих пристроях на базі Linux, таких як Raspberry Pi або деяких мікроконтролерів [4,5]. Існують також такі програмні платформи, як Edge Impulse або Neuton для створення високооптимізованих наскрізних TinyML-додатків з нуля без написання коду.

Висновки. Рішення на TinyML дозволяє використовувати моделі побудовані за допомогою глибокого навчання на малопотужних контролерах та вирішує проблеми конфіденційності і безпека даних так як вони залишаються на пристроях. Виглядає перспективним поєднати технології TinyML з розподіленими моделями, наприклад федеративним навчанням. В IDS використання TinyML також вирішить проблеми конфіденційності і безпека даних, а ще й зменшить час на виявлення аномального стану та вивільнить мережні ресурси.

ПЕРЕЛІК ПОСИЛАНЬ

1. Statista: Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions). <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>. Accessed 05 February 2022.
2. Rohan Banerjee, Hands-on TinyML. BPB Publications. 2023. -309 p.
3. Gian Marco Iodice, TinyML Cookbook, Second Edition: Combine machine learning with microcontrollers to solve real-world problems, Packt. 2023, isbn:9781837637362. -344p.
4. Pete Warden, Daniel Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers, O'Reilly Media. 2019. – 504p.
5. Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili, Machine Learning with PyTorch and Scikit-Learn. Packt. 2022. – 717p.

МЕТОДИКА ВИКОРИСТАННЯ OSINT ПІД ЧАС ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

Анотація. Описано процес впровадження методології розвідки на основі відкритих джерел як першочергового етапу тестування на проникнення.

Ключові слова: розвідка, тестування на проникнення, інформаційна безпека, кібербезпека, відкриті джерела, пентест, OSINT.

Вступ. Розвідка на основі відкритих джерел – це активно зростаюча галузь інформаційної безпеки, що наразі залучається у військовій справі, перевірці кадрів при працевлаштуванні, веденні бізнесу та перевірці контрагентів, а також у особистих цілях з метою отримання необхідної інформації для подальшого використання.

Технології та засоби розвідки на основі відкритих джерел невпинно розвиваються задля покращення рівня інформаційної безпеки та обізнаності громадян щодо інформаційної безпеки та правил поведження в глобальній мережі Інтернет.

Постановка задачі. Враховуючи кількість кібер-інцидентів, які щодня фіксуються кіберполіцією України, є сенс превентивно залучити OSINT-фахівця для перевірки ІКС як першочерговий етап тестування на проникнення вибраної організації та подальших дій щодо підвищення рівня обізнаності правил кібербезпеки та поведження в глобальній мережі Інтернет співробітників організації..

Основний зміст роботи. OSINT (Open-Source Intelligence) – це діяльність по отриманню розвідувальної інформації з відкритих джерел кіберпростору; розвідка на основі відкритих джерел.

Розвідка на основі відкритих джерел активно залучається у багатьох сферах, починаючи від малого бізнесу та завершуючи роботою спецслужб.

Вперше термін OSINT був використаний військовими та розвідувальними службами для позначення збору стратегічно важливої, але загальнодоступної інформації у питаннях національної безпеки. З появою Інтернету, соціальних медіа та цифрових сервісів з'явилися широкі можливості для збору інформації на основі відкритих джерел.

У кібербезпеці OSINT може використовуватись як для захисту, так і для нападу. Так, ці техніки можуть допомогти у виявленні витоків даних. Наприклад: часто трапляється, що хакери після зламу систем публікують частину вкрадених даних у темній мережі. Якщо вчасно зреагувати на виток інформації, вона може запобігти подальшому розповсюдженню і зберегти недоторканність чутливих даних.

Більш того, відповідно до більшості прийнятих стандартів, OSINT є першою ланкою в чи не найпопулярнішому виді тестувань з кібербезпеки – тестах на проникнення.

Водночас, цю ж техніку використовують для того, аби убезпечити дані конкретних осіб та працівників державних установ і приватних компанії. Виявивши слабкі точки, фахівці можуть покращити безпеку захисту персональних даних.

Кожна задача, яку слід вирішити OSINT-спеціалісту, є унікальною і передбачає окремий підхід. Тобто, розвідка на основі відкритих джерел не має готового алгоритму. Серед способів отримання інформації можна виділити наступні:

- моніторинг соціальних мереж. Даний метод передбачає моніторинг платформ соціальних мереж на предмет активності, пов'язаної з конкретною особою. Аналіз соціальних мереж може допомогти виявити потенційні загрози безпеки або підозрілу поведінку;

- аналіз форумів і дошок оголошень. Цей спосіб включає в себе аналіз активності на форумах і дошках оголошень, пов'язаних з певною особою, включаючи повідомлення та коментарі. Це може допомогти виявити потенційні уразливості або підозрілу поведінку, а також обізнаність або причетність користувача до певної теми або спільноти;

- пошукова система і аналіз темних веб-сторінок. Метод передбачає аналіз результатів пошуку на просторах темної мережі на предмет інформації, пов'язаної з конкретною особою, включаючи будь-які витоку паролів або особистої інформації. Це може допомогти виявити потенційні загрози безпеки або вразливості;

- аналіз метаданих: Сенс даного методу – проаналізувати метадані, пов'язаних з конкретною особою, таких як: дата і час дії, IP-адреси та інформація про пристрій тощо. Це може допомогти виявити потенційні загрози або підозрілу поведінку, а також відстежити активність користувача на різних платформах;

- поведінковий аналіз. Методика передбачає аналіз шаблонів поведінки, пов'язаних з конкретною особою, таких як: частота і час активності, використання мови та взаємодію з іншими користувачами. Це може допомогти виявити потенційні загрози або підозрілу поведінку, а також проаналізувати коло спілкування конкретної особи, її звички та зв'язки з іншими особами тощо.

Тестування на проникнення – це процес перевірки (або тестування) безпеки системи, який включає в себе засоби пошуку та використання вразливостей системи з метою їхнього подальшого уникнення.

Тестування на проникнення є одним із основних методів, що залучається для ідентифікації вразливих областей системи.

При тестуванні на проникнення проводиться оцінка безпеки системи, персоналу або організації із використанням етичних методів проникнення.

Тестуванням на проникнення займаються спеціалісти, які входять в групу «білих капелюхів».

Зазвичай, метою тестування на проникнення є ідентифікація максимальної кількості вразливостей системи задля усунення «чорних дір» та перевірки пильності та обізнаності базових правил кібербезпеки співробітників організації.

Тож, головно метою тестування на проникнення можна назвати пошук усіх можливих вразливостей ІКС, які можуть призвести до небажаних наслідків, а саме:

- порушення конфіденційності, цілісності та доступності інформації;
- некоректна робота системи;
- відмова від обслуговування системи;
- економічні ризики та фінансові втрати.

Тестування на проникнення однаково стосується як віртуального рівня безпеки ІКС, так і фізичного (технічне обладнання ІКС).

Для реалізації тестування на проникнення необхідний комплексний підхід, що складається із засобів та методик, які проводяться кваліфікованими спеціалістами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Щодо розвідки з відкритих джерел у мережі загального користування інтернет / Сумський С.А. // Міжнародна науково-практична конференція «Кримінальна розвідка методологія законодавства, зарубіжний досвід» Одеса, ОДУВС, 29 квітня 2016 р.
2. Shodan — Computer Search Engine | OSINT Framework #2. URL: <https://cenabibrahimov.medium.com/shodan-computer-search-engine-osint-framework-2-ed5d9ab0980b>
3. Про розвідувальні органи. Закон України від 22.03.2001 № 2331-III // Відомості Верховної Ради України. — 2001. — № 19. — URL: <https://zakon.rada.gov.ua/laws/show/2331-14>
4. Спецслужби (ГУР, СЗР, СБУ) // Український мілітарний портал. — 2009 — URL: <https://mil.in.ua/uk/specsluzhby=gur-szr-sbu/>

УДК 004.056.5(045)

Д.С. Тимофєєв¹, А.С. Совенко¹

¹Харківський національний університет радіоелектроніки, Харків, Україна

МІНІМІЗАЦІЯ РИЗИКІВ ПОВ'ЯЗАНИХ З ПЕРСОНАЛОМ В ПРОЦЕСІ ВПРОВАДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ ВІДПОВІДНО ДО ВИМОГ СТАНДАРТУ ISO/IEC 27001:2022

Анотація. Описано класифікації порушників інформаційної та кібербезпеки. Проаналізовано сценарії загроз пов'язані з людським фактором. Наведено заходи контролю персоналу на основі ISO 27001:2022.

Ключові слова: Ризик, внутрішні загрози, класифікація порушників, обізнаність персоналу, зловмисник, сценарії загроз.

Вступ. Людський фактор часто найважче контролювати та передбачити, коли йдеться про захист даних. В багатьох випадках достатньо лише в одного мотивованого зловмисника для реалізації ризику з катастрофічними наслідками, наприклад, постійно існує потенційна небезпека від зловмисних інсайдерів і

незадоволених співробітників, які хочуть завдати шкоди репутації компанії або викрасти дані, покидаючи організацію.

Постановка задачі:

- проаналізувати внутрішні антропогенні загрози;
- визначити методи протидії загрозам пов'язаним з людським фактором в процесі впровадження СУІБ.

Основний зміст роботи: Стрімкий перехід на використання цифрових технологій в усіх сферах діяльності та приватного життя людини, промислових системах та адміністративних установах, інших об'єктах критичної інфраструктури призводить до симетричного зростання рівня загроз та ризиків пов'язаних з інформаційною безпекою, які необхідно адекватно оцінювати, обробляти відповідними методами та бажано завчасно упереджувати або швидко та ефективно мінімізувати наслідки. Значна кількість таких ризиків тим чи іншим чином пов'язана з загрозами або вразливостями антропогенної природи. За різними статистичними даними, антропогенний фактор впливу присутній в причинах від 60 до 90 відсотків інцидентів інформаційної та кібербезпеки.

З метою формалізації оцінки ризиків антропогенної природи зазвичай використовують модель порушника (зловмисника). Модель порушника представляє собою опис можливих дій порушника, який складається на основі аналізу типу потенційного зловмисника, рівня його повноважень, знань, мотивації, теоретичних та практичних можливостей.

Можна виділити три категорії зловмисників залежно від їхнього місцезнаходження та знань про організацію-мішень [1]:

- внутрішні в організації: також відомі як інсайдери, вони мають високий рівень знань про мережу, системи, безпеку, політику та процедури організації-мішені, та її процедури. Згідно з [2], існує два вектори, що сприяють інсайдерським загрозам. Перший – коли співробітники організації, які мають зловмисні наміри (наприклад, розголошувати та/або продавати неpubлічну інформацію) та другий – коли їх наміри незловмисні (наприклад, ненавмисна помилка або недбалість). Більшість втрат пов'язана з останнім вектором загроз.

- зовнішні по відношенню до організації: У порівнянні з внутрішніми загрозами, такі зловмисники повинні витратити багато часу перед атакою на збір інформації про ціль через обмеженість попередніх знань.

- змішані групи: складаються з внутрішніх і зовнішніх зловмисників.

Слід також відзначити, що досить часто представники групи зовнішніх порушників використовують методи соціальної інженерії для реалізації вектору атаки через експлуатацію співробітників організації, які в свою чергу можуть реалізовувати як свідомі порушення, так і діяти без розуміння, що є інструментом реалізації загроз.

В роботі [3] виконано порівняльний аналіз відомих класифікацій порушників в результаті узагальнено виділено наступні ролі:

- кримінальні – хакерство за гроші;

- ідеологічні, соціальні (хактивісти) – відстоюють соціальну ідеологію/ідею;
- ідеологічні, насильницькі (терористи) – прагнуть просунути політичну позицію за допомогою насильства;
- національно-державні – хакери, що працюють на або від імені національних груп або урядів окремих держав;
- інсайдери(працівники), що діють навмисно – діючі працівники, які зловживають довіреною доступом зі зловмисною метою;
- інсайдери(працівники), що діють ненавмисно – необізнаний або недбалый працівник;
- анархісти – хакерство заради розваги та/або слави.
- корпоративні – промислове/корпоративне шпигунство.

Визначимо базові незалежні сценарії загроз які можливо деталізувати в процесі аналізу ризиків пов'язаних з персоналом, з урахуванням типових сценаріїв [4].

1. Витік: Порушник отримує конфіденційну інформацію з систем установи. Цей сценарій включає витіки інформації, що ідентифікує особу (персональні дані), а також широкомасштабний витік службової інформації, комерційної таємниці або іншої конфіденційної інформації.

2. Шахрайство: Порушник змінює або підробляє інформацію в системах установи, щоб установа виплатила гроші або перевела інші активи за вказівкою супротивника. Цей сценарій зосереджується на шахрайських транзакціях внаслідок кібератаки, і виключає шахрайство, здійснене некібернетичними методами.

3. Нецільове використання: Порушник модифікує або підробляє програмне забезпечення чи конфігураційні дані в системах установи, щоб зловмисник міг спрямувати їх використання (зазвичай для перепродажу потужностей, як у випадку з фермами ботнетів або майнінгом криптовалют). Цей сценарій зосереджується на узурпації ресурсів, що зазвичай відбувається дуже приховано.

4. Руйнування: Порушник модифікує або знищує інституційні активи, щоб перешкодити установі виконувати свої основні функції. Цей сценарій включає заперечення, зрив або підрив бізнес-операцій супротивником.

5. «Дружній вогонь»: Порушник обманом змушує керівників бізнес-напрямів або персонал з кіберзахисту до дій, що підривають операційну діяльність. Цей сценарій зосереджується на модифікації або фальсифікації бізнес-даних або даних конфігурації, а також на модифікації або порушенні бізнес-функцій.

6. Атака з висхідного потоку: Порушник компрометує постачальника або партнера, щоб збільшити вразливість організації до атак. Цей сценарій включає атаки на установи-партнери, а також на установи, що входять до ланцюга постачання установи.

7. Нанесення шкоди репутації: Противник порушує роботу установи або фабрикує інформацію, яку установа подає своїм клієнтам та партнерам, завдаючи шкоди її репутації. Цей сценарій тісно пов'язаний з тими, що передбачають порушення або заперечення функцій місії, але також включає

модифікацію несуттєвої, або зовнішньо несуттєвої, але видимої ззовні інформації або послуг у спосіб, що підриває довіру до інституції.

8. Атака "сходінками" (через ланцюг постачання): Порушник компрометує системи установи з метою атакувати організації, що знаходяться умовно «нижче за течією» (наприклад, клієнтів, клієнтів клієнтів). Як і попередній цей сценарій відноситься до тих, що пов'язані з порушенням функцій місії. Однак він також пов'язаний зі сценаріями, що передбачають отримання конфіденційної інформації або шахрайські транзакції.

9. Вимагання: Порушник модифікує або виводить з ладу активи з метою отримання фінансової вигоди (наприклад програми-вимагачі, розподілені атаки на відмову в обслуговуванні). Цей сценарій тісно пов'язаний з тими, що передбачають модифікацію з метою шахрайства, а також порушення або відмову у виконанні бізнес-функцій.

Впровадження системи управління інформаційною безпекою (СУІБ) безумовно не гарантує побудови надійної системи кіберзахисту, але надає можливість зацікавленим сторонам організувати контрольовані процеси управління для захисту інформаційних активів організації з постійним вдосконаленням. В частині протидії ризикам пов'язаним з діяльністю персоналу процесний підхід є одним з найбільш ефективних. Важливою групою вимог до СУІБ організації є впровадження виділених в окремий розділ заходів контролю персоналу на основі ISO 27001 [5]. Розглянемо їх детально.

Контроль за відбором кандидатів. Перевірка біографічних даних усіх кандидатів на роботу в організації здійснюється до вступу в організацію та на постійній основі з урахуванням чинного законодавства, нормативних актів та етики та бути пропорційною бізнес-вимогам, класифікації інформації, до якої надається доступ, та передбачуваним ризикам.

Умови працевлаштування. У трудових договорах повинні бути визначені обов'язки персоналу та відповідальність персоналу та організації за інформаційну безпеку.

Поінформованість про інформаційну безпеку, освіта та навчання з питань інформаційної безпеки. Персонал організації та відповідні зацікавлені сторони повинні отримувати відповідну обізнаність, освіту та навчання з питань інформаційної безпеки та регулярно оновлення політики інформаційної безпеки організації, політики та процедур, що стосуються конкретних тем, відповідно до їхніх посадових обов'язків.

Контроль дисциплінарного процесу. Дисциплінарний процес повинен бути формалізований та доведений до відома для вжиття дій проти персоналу та інших відповідних зацікавлених сторін, які вчинили порушення політики інформаційної безпеки.

Обов'язки після звільнення або зміни місця роботи. Обов'язки та відповідальність за інформаційну безпеку, які залишаються дійсними після звільнення або зміни місця роботи, повинні бути визначені, забезпечені виконанням та доведені до відома відповідного персоналу та інших зацікавлених сторін.

Угоди про конфіденційність або нерозголошення. Угоди про конфіденційність або нерозголошення, що відображають потреби організації в захисті інформації, повинні бути визначені, задокументовані, регулярно переглядаються та підписуються персоналом та іншими відповідними зацікавленими сторонами.

Керування дистанційною роботою. Заходи безпеки повинні бути впроваджені, коли персонал працює віддалено, щоб захистити інформацію, до якої є доступ, яка обробляється або зберігається за межами приміщень організації.

Звітування про події інформаційної безпеки. Організація повинна забезпечити механізм, за допомогою якого персонал може повідомляти про події інформаційної безпеки, що спостерігаються або підозрюються, через відповідні канали.

Окрім визначених контролей ризику пов'язані з антропогенними загрозами мінімізуються шляхом впровадження суворого контролю доступу та політик, які обмежують ризик потенційних інцидентів і збитки. Необхідно забезпечити дотримання принципу найменших привілеїв. Також вкрай важливим є застосування надійних методів автентифікації та шифрування, таких як багатофакторна автентифікація (MFA) або інфраструктура відкритих ключів (PKI), для запобігання несанкціонованому доступу або витоку даних. Крім того, мають бути встановлені чіткі та узгоджені політики та процедури для керування правами доступу, як-от скасування або зміна доступу, коли користувачі залишають організацію, змінюють ролі або порушують правила.

Базовими заходами запобігання внутрішнім загрозам є виявлення та моніторинг користувачів, які становлять найбільший ризик для організації. Зокрема, які мають доступ до інформації що підлягає захисту, ті що нещодавно змінили ролі чи обов'язки, користувачі, які демонстрували ознаки незадоволення, або з особистими фінансовими проблемами.

Для забезпечення моніторингу можливо запровадити системи аналітики поведінки користувачів (UBA) або аналітики поведінки користувачів і об'єктів (UEBA), з їх допомогою можливо аналізувати аномальні чи підозрілі дії користувачів, наприклад доступ або передача даних у нестандартний час, шаблони поведінки чи місцезнаходження.

Обов'язкове проведення аудитів і переглядів систем, даних і мережі, для виявлення вразливостей, якими можуть скористатися інсайдери. Необхідно проводити регулярні випадкові перевірки дій користувачів, журналів доступу, передачі даних та інцидентів безпеки, для виявлення аномалій або порушень. Політика безпеки, процедури та засоби контролю, мають регулярно переглядатись щоб переконатись, що вони оновлені, ефективні та відповідають бізнес-цілям і вимогам відповідності.

Важливо мати чітко визначений і перевірений план реагування на інциденти, який визначає ролі та обов'язки групи безпеки, кроки та процедури стримування та пом'якшення інциденту, канали зв'язку та ескалації для інформування та залучення відповідних сторін, а також інструменти і ресурси для підтримки процесу реагування та відновлення. Забезпечується проведення ретельного та об'єктивного розслідування інцидентів, щоб визначити

першопричину та наслідки, а також оновити та вдосконалити заходи безпеки та засоби контролю на основі висновків і рекомендацій розслідування.

Не слід також забувати про постійну співпрацю з внутрішніми та зовнішніми зацікавленими сторонами та учасниками процесу забезпечення безпеки в організації.

Висновки: внутрішні загрози зазвичай важче виявити або упередити, ніж зовнішні і їм складно протидіяти виключно програмно-технічними рішеннями безпеки. Якщо зловмисник використовує авторизований вхід, наявні механізми безпеки можуть не визначити аномальну поведінку. Крім того, зловмисникам легше уникнути виявлення, якщо вони знайомі із заходами безпеки організації. Важливо зазначити, що широке застосування як засіб протидії подібним класам загроз знаходять системи управління ризиками засновані на поведінковому аналізі співробітників.

ПЕРЕЛІК ПОСИЛАНЬ

1. L. Ablon, M.C. Libicki, and A.A. Golay. Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar. Rand Corporation, 2014.
2. M. Albanese, S. Jajodia, A. Pugliese, and V. Subrahmanian, "Scalable Analysis of Attack Scenarios," in V. Atluri, C. Diaz (Eds.), European Symposium on Research in Computer Security – ESORICS 2011, Springer, pp. 416–433, 2011.
3. Falk, Courtney. Using an Ontology to Classify Cyber Threat Actors., 2018.
4. Cyber Threat Modeling: Survey, Assessment, and Representative Framework, 2018, Authors: Deborah J. Bodeau, Catherine D. McCollum, David B. Fox
5. ISO/IEC 27001:2022(en), Information security, cybersecurity, and privacy protection. Information security management systems. Requirements.

УДК 004.056.5(045)

І.С. Біжко¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ОСОБЛИВОСТІ ВПРОВАДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ НА СІЛЬСЬКОГОСПОДАРСЬКИХ ПІДПРИЄМСТВАХ

Анотація. Робота передбачає аналіз існуючих загроз та ризиків для інформаційної безпеки на сільськогосподарських підприємствах, вивчення специфічних аспектів господарювання в аграрному секторі та розробку стратегій, що враховують особливості та потреби сільськогосподарських підприємств.

Ключові слова: Система управління інформаційною безпекою (СУІБ), сільське господарство, сільськогосподарське підприємство.

Вступ. У сучасному інформаційному суспільстві сільське господарство відіграє ключову роль у забезпеченні продовольчої безпеки та економічного розвитку. Однак зростаюча залежність від інформаційних технологій також вносить свої виклики та загрози для сільськогосподарських підприємств. Сучасні

технології пропонують низку можливостей для оптимізації процесів, але одночасно створюють нові ризики, пов'язані з безпекою інформації.

У цьому контексті, важливо розглядати системи управління інформаційною безпекою як стратегічний інструмент для сільськогосподарських підприємств. Впровадження таких систем стає необхідністю для забезпечення цілісності, конфіденційності та доступності цінної інформації, яка є критичною для ефективного ведення господарської діяльності.

У даному дослідженні розглядаються особливості впровадження систем управління інформаційною безпекою на сільськогосподарських підприємствах. Аналіз викликів та можливостей, що стоять перед цими підприємствами, дозволяє визначити ключові аспекти, які варто враховувати при розробці та впровадженні ефективних стратегій управління інформаційною безпекою. Такий підхід допомагає не лише захищати важливі ресурси та інформацію, але і стимулює сталість та розвиток у сільському господарстві в умовах сучасних технологічних викликів.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- провести аналіз інформації щодо існуючих вразливостей та ризиків для інформаційної безпеки на сільськогосподарських підприємствах;
- дослідити специфічні аспекти господарювання в аграрному секторі та визначити особливості, які впливають на інформаційну безпеку.

Основний зміст роботи. СУІБ - це скорочення від "Система Управління Інформаційною Безпекою". Ця система включає в себе комплекс заходів та технічних засобів, спрямованих на забезпечення конфіденційності, цілісності та доступності інформації в організації чи підприємстві. Метою СУІБ є ефективний контроль і захист інформації від небажаних втручань, кіберзагроз, витоків даних та інших потенційних ризиків для інформаційної безпеки. Схема СУІБ виглядає наступним чином (рис. 1)

Система управління інформаційною безпекою в контексті сільського господарства є ключовим інструментом для забезпечення захисту інформаційних ресурсів та оптимізації безпеки даних. Розглянемо деякі особливості, що обґрунтовують використання СУІБ у сільському господарстві:

- захист фінансових даних:

Сільськогосподарські підприємства обробляють значні обсяги фінансових даних, включаючи бюджетування, облік витрат та доходів. СУІБ допомагає встановити ефективні заходи безпеки для захисту цих фінансових інформаційних потоків. Сільськогосподарські підприємства стають об'єктом кібератак через цінні дані та залежність від технологій. Захист від кіберзагроз є необхідністю.

- оптимізація виробничих процесів:

Використання СУІБ дозволяє захищати дані, пов'язані з агротехнічними дослідженнями та виробничими процесами. Це забезпечує конфіденційність інновацій та відкриттів у галузі сільського господарства. Сільське господарство

все більше використовує технології, щоб вдосконалити виробничі процеси. Захист технічних даних, таких як результати агрономічних досліджень чи дані про виробничі процеси, стає критичним для конкурентоспроможності.

- зберігання та аналіз даних:

СУІБ забезпечує безпеку для баз даних, які містять інформацію про врожай, ринкові тенденції та клієнтську інформацію. Сільськогосподарські підприємства зберігають значну кількість особистих та фінансових даних своїх співробітників та клієнтів. Захист цих даних є важливим аспектом інформаційної безпеки.

- аналітичні інструменти

СУІБ допомагають в здійсненні ефективного аналізу даних для прийняття управлінських рішень.

- захист технічних систем:

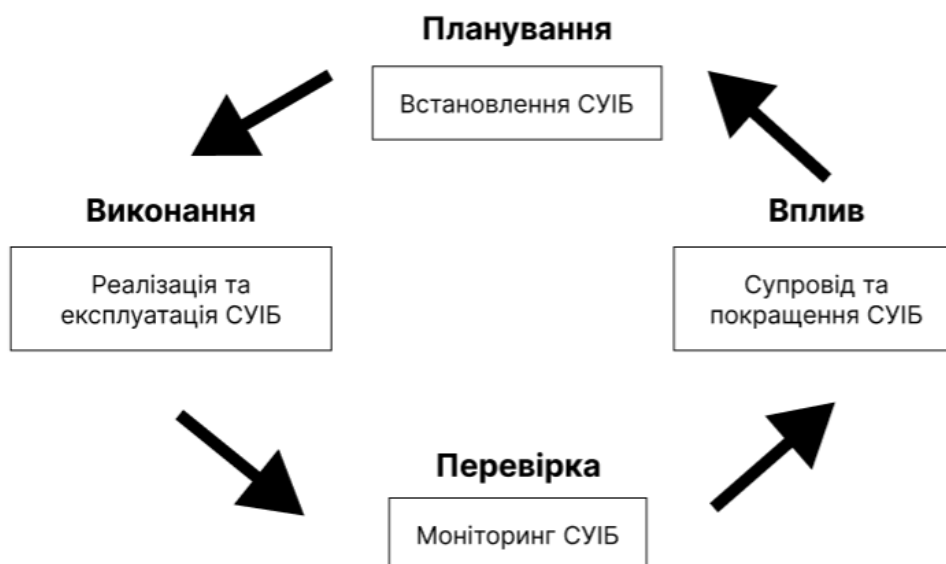


Рис. 1. Схема системи управління інформаційною безпекою

В сучасному сільському господарстві використовуються технічні засоби, такі як дрони та сенсори. СУІБ гарантує безпеку цих технічних систем, запобігаючи несанкціонованому доступу та кіберзагрозам.

Залучення інформаційної безпеки на сільськогосподарських підприємствах визначається рядом факторів, від яких залежить ефективне управління та збереження цінних даних. У такому випадку, з плином часу, питання створення СУІБ на об'єктах аграрної промисловості постійно поставатиме. У ході дослідження було виявлено, що сільське господарство, як і будь-яка інша галузь, стикається зі зростаючими викликами і загрозами в інформаційному просторі. Аналіз існуючих ризиків та загроз показав, що

конфіденційність фінансових та технічних даних є ключовим аспектом, на який слід звертати увагу при впровадженні систем управління інформаційною безпекою.

Особливості господарювання в аграрному секторі вимагають унікальних стратегій та рішень для забезпечення безпеки інформаційних ресурсів. Враховуючи це, були розроблені та визначені стратегії, специфічні для сільськогосподарських умов, з урахуванням їх потреб та особливостей.

Сільськогосподарські підприємства в Україні стикаються із рядом вразливостей та ризиків у сфері інформаційної безпеки, які можуть вплинути на їхню діяльність та безпеку даних. Однією з найважливіших проблем є відсутність належного рівня кіберзахисту. Сільськогосподарські підприємства, в порівнянні з іншими галузями, можуть не приділяти достатньо уваги захисту від кібератак, що робить їх особливо вразливими.

Низький рівень освіти та свідомості персоналу також є серйозною проблемою. Відсутність належної освіти з питань кібербезпеки може призводити до соціального інженерінгу та несанкціонованого доступу до систем. Крім того, недостатній рівень захисту аграрних даних є проблемою, оскільки збільшення використання інформаційних технологій може створювати ризики для конфіденційності та цілісності цих даних.

Використання технологій Інтернету речей (IoT) в аграрному секторі також може призводити до вразливостей через неналежно захищені пристрої та мережі. Обмін даними в агросфері може включати ризики, пов'язані з конфіденційністю та захистом цих даних.

Фізична безпека є важливою аспектом, особливо через велику територію сільськогосподарських підприємств, що створює виклики для забезпечення адекватного рівня безпеки обладнання та об'єктів. Крім того, залежність від інформаційних технологій та систем може збільшувати ризик від інцидентів та витоку даних.

Важливо враховувати інші аспекти, такі як ризики обміну даними в агросфері та відсутність планів кризової готовності. Застосування належних заходів щодо кіберзахисту, впровадження ефективних політик та процедур безпеки, регулярне навчання персоналу, а також використання сучасних технологій безпеки можуть допомогти зменшити ці ризики та вразливості.

Висновки. В результаті дослідження було проведено аналіз вразливостей та ризиків, які притаманні сільськогосподарським підприємствам. Також дослідили особливості які можуть впливати на інформаційну безпеку.

ПЕРЕЛІК ПОСИЛАНЬ

1. ISO/IEC 27000 - Інформаційна технологія. Безпека інформації. Моделі систем управління безпекою. Загальні та словникові положення.
2. ISO/IEC 27001 - Інформаційна технологія. Безпека інформації. Системи управління інформаційною безпекою. Вимоги.
3. ISO/IEC 27002 - Інформаційна технологія. Безпека інформації. Практичні рекомендації з управління інформаційною безпекою.

4. Коваленко, О. (2020). "Технології захисту інформації в агропромисловості." Київ: Видавництво "Агроінфо".

УДК 004.056.4

О.В. Кручинін¹, В.О. Святошенко¹

¹Харківський національний університет радіоелектроніки, Харків, Україна

ХАРАКТЕРИСТИКА АТАК НА СИСТЕМИ ВИЯВЛЕННЯ ВТОРГНЕНЬ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Анотація. Наведена загальна характеристика атак на системи виявлення вторгнень на основі машинного навчання. Особливу увагу приділено умовам та цілям реалізації цих атак.

Ключові слова: машинне навчання; системи виявлення вторгнень; ворожі атаки; глибоке навчання; мережева безпека; кібербезпека.

Вступ. Виявлення зловмисних подій є важливим питанням у сфері кібербезпеки. Оскільки ручна перевірка неможлива, коли щодня відбуваються велика кількість подій, фахівцям з кібербезпеки допомагають системи виявлення вторгнень (IDS - Intrusion Detection System), які аналізують дані з різних джерел і при виконанні певних умов генерують оповіщення про небезпеку. Сучасні системи IDS все частіше будуються на базі систем машинного навчання (ML - Machine Learning). Але самі системи ML мають вразливості, які дають можливість здійснювати атаки на самі IDS. Захист від цих атак є складним завданням через широку різноманітність можливих механізмів і векторів атак. Тому розуміння цих механізмів і векторів є важливою задачею.

Основний зміст роботи. IDS – це система виявлення вторгнень, яка представляє собою програмне забезпечення, призначене для забезпечення комп'ютерної безпеки, яке спрямоване на виявлення різноманітних порушень безпеки, починаючи від зовнішніх спроб злому до інсайдерських вторгнень. Основними завданнями IDS є моніторинг хостів і мережі, оцінка активності комп'ютерної системи, видача попереджень та реагування на аномальну поведінку. Важливим обмеженням багатьох IDS є фільтрація та зменшення кількості хибних тривог. Враховуючи, що з розвитком інформаційних технологій спостерігається поява все нових атак, перспективним є створення IDS на основі ML. Зокрема, багато IDS поліпшують ефективність, використовуючи нейронні мережі (NN - Neural Network) для глибокого навчання. Системи IDS на основі глибоких нейронних мереж (DNN - Deep Neural Network) створюються для покращення процесу навчання, обробки великих обсягів даних і передбачення майбутніх атак.

Однак, системи на основі ML чутливі до різноманітних атак, спрямованих на неправильну класифікацію. До таких атак відносяться змагальні атаки, які використовують змагальне машинне навчання (AML - Adversarial Machine

Learning). У змагальному машинному навчанні зловмисник намагається заплутати модель машинного навчання і змусити її прийняти неправильне рішення. Противник досягає цього шляхом модифікації вхідних даних, які подаються на модель машинного навчання або на етапі навчання (атака отруєння), або на етапі виведення (атака ухилення) [1].

Для реалізації атаки отруєнням необхідні глибоке розуміння інформаційних систем та принципів побудови засобів захисту. У цій атаці зловмисник намагається забруднити навчальні дані, вносячи в них точно сплановані зразки, що в кінцевому підсумку ставить під загрозу процес навчання. Крім того, атаки отруєння можна класифікувати як:

- атаки отруєння, що породжують помилки, в яких зловмисник намагається спричинити відмову в обслуговуванні, спричиняючи якомога більше помилок класифікації;
- атаки отруєння, які створюють конкретні помилки. Тобто, метою зловмисника є створення певних помилок класифікації.

Атака ухилення реалізується шляхом введення шкідливих зразків, які не впливають на навчальні дані. Таким чином, мета полягає в тому, щоб помилково класифікувати зразки шкідливого програмного забезпечення як доброякісні під час роботи моделі. Крім того, атаки ухилення можна класифікувати як:

- атаки ухилення на основі загальних помилок, в яких зловмисник зацікавлений в обмані класифікації, незалежно від того, що класифікатор прогнозує як вихідний клас;
- атаки ухилення з конкретною помилкою, коли зловмисник намагається обдурити класифікацію, але неправильно класифікує зразки даних як певний клас.

Цілі, тактика та результати атак на IDS на основі ML залежить від рівня доступу зловмисника до поточних параметрів моделі [2]:

- Атаки White-Box: за умови що модель машинного навчання має відкритий вихідний код та до неї є вільний доступ. Зловмисник має глибоке розуміння архітектури мережі та параметрів, які були отримані в результаті навчання.
- Атаки Black-Box: зловмисник не має жодних попередніх знань про модель і аналізує вразливість моделі, використовуючи інформацію з налаштувань або попередні вхідні дані.
- Атаки Gray-Box: це проміжний варіант між Black-Box та White-Box в ході якого зловмисник проходить процес ітеративного навчання, який використовує процеси виведення для отримання додаткового розуміння моделі. Таким чином, він може мати часткові знання про модель.

Реалізація атаки Black-Box вважається слабкою та не матиме такого ж впливу, як атака White-Box [3]. Слід зазначити, що коли знання обмежені, як, наприклад, у сценаріях Gray-Box та Black-Box, можуть бути здійснені атаки на конфіденційність, щоб дізнатися більше про цільовий класифікатор ML [4].

Під контролем зломисника в загальному випадку можуть бути наступні елементи системи ML [2]:

- Навчальні дані: набір даних, що використовується для навчання моделей ML. Доступ до навчальних даних буває двох видів: доступ на читання, який може дозволити зломиснику відтворити фазу навчання і отримати детектор, подібний до того, що використовується в IDS; або доступ на запис, який може бути використаний для здійснення атак отруєння шляхом введення нових даних або модифікації наявних зразків. Отримання доступу до навчальних даних залежить від джерела отримання IDS: детектор може бути розроблений і управлятися повністю власними силами, або ж придбаний у стороннього постачальника. У першому випадку зломисник теоретично може отримати навчальні дані, ідентифікувавши пристрій, на якому зберігається набір даних, а потім перевірити або викрасти фактичні дані [1].

- Набір ознак: він обумовлює особливості моделей ML, які використовуються. Атака простору ознак змінює їх. Тобто, розмір набору ознак моделі ML може бути використаний для зловживань як поверхня атаки. В роботі [5] зазначається, що великі набори ознак мають більше характеристик, що дає зломиснику більше можливостей для маніпулювання ними. Як наслідок, великі набори ознак можуть бути збурені швидше, ніж менші набори ознак, які мають менше модифікованих ознак і потребують більше збурень.

- Модель виявлення: детально описує навчену модель ML, що входить до складу IDS і використовується для виявлення атак. Знання про модель можуть бути нульовими, частковими або повними. Доступ до моделі вимагає високих адміністративних прав. Тому вірогідність того, що інфікований хост надасть зломиснику доступ до IDS, який містить модель виявлення, є малою.

- Oracle: в атаках типу "оракул" зломисник, який отримав доступ до моделі за допомогою оракул-прогнозу, викрадає копію віддалено розгорнутої моделі машинного навчання. Це дозволяє зломиснику продублювати функціональність моделі, тобто "вкрасти модель".

Глибинні маніпуляції: це маніпуляції зломисника, які можуть змінити обсяг трафіку або одну чи більше ознак у просторі ознак, який атакується. В результаті, цього стає можливим реалізація атаки Black-Box, але вона не матиме такого ж впливу, як атака White-Box.

При аналізі атак слід враховувати обмеження з якими стикається зломисник при реалізації атаки на ML [6]:

- відсутність уніфікованих атак для різних моделей ML;
- розмір збурення повинен бути достатньо великим, щоб мати ефект та достатньо малим, щоб не бути виявленим;
- наявність стійкості до збурень в реальних системах.

За спрямованістю атаки на ML можна розділити на дві групи:

- Цілеспрямована атака. Така атака спрямована на один вхідний зразок або групу зразків.

- Нецільові атаки. Модель ML в цій атаці дає збій випадковим чином. Крім того, нецільові атаки легше здійснити, ніж цілеспрямовані, оскільки вони пропонують більше можливостей і простору для дій зловмисника.

Висновки. У кібербезпеці використанню алгоритмів ML приділяється багато уваги, особливо в IDS. Велика кількість розробок спрямована на підвищення швидкості, точності, достовірності та інших важливих метрик IDS на основі ML. Але, зловмисні атаки можуть мати значний вплив на алгоритми ML. Таким чином, IDS на основі ML є вразливими до зловмисних атак.

Атаки на IDS на основі ML, впливають на точність класифікації IDS, що значно зменшує ефективність засобів кібербезпеки. У такій ситуації IDS буде ненадійною в захисті, що становить серйозну загрозу для наших мереж. Таким чином, є перспективним розвиток методів та засобів захисту IDS на основі ML від змагальних атак.

ПЕРЕЛІК ПОСИЛАНЬ

1. Chakraborty, A.; Alam, M.; Dey, V.; Chattopadhyay, A.; Mukhopadhyay, D. Adversarial Attacks and Defences: A Survey. arXiv 2018, arXiv:1810.00069. Available online: <http://arxiv.org/abs/1810.00069>. (дата звернення: 25.11.2023).
2. Apruzzese, G.; Andreolini, M.; Ferretti, L.; Marchetti, M.; Colajanni, M. Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems. Digit. Threat. Res. Pract. 2021, 3, 1–19.
3. Liu, G.; Khalil, I.; Khreishah, A. ZK-GanDef: A GAN Based Zero Knowledge Adversarial Training Defense for Neural Networks. In Proceedings of the 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Portland, OR, USA, 24–27 June 2019; pp. 64–75.
4. Ibitoye, O.; Abou-Khamis, R.; Matrawy, A.; Shafiq, M.O. The Threat of Adversarial Attacks on Machine Learning in Network Security—A Survey. arXiv 2019, arXiv:1911.02621. Available online: <http://arxiv.org/abs/1911.02621> (дата звернення: 25.11.2023).
5. Sarker, I.H.; Abushark, Y.B.; Alsolami, F.; Khan, A.I. Intrudtree: A machine learning based cyber security intrusion detection model. Symmetry 2020, 12, 754.
6. Zhang, J.; Li, C. Adversarial examples: Opportunities and challenges. IEEE Trans. Neural Netw. Learn. Syst. 2019, 31, 2578–2593

ВІДОМОСТІ ПРО АВТОРІВ

Алексєєв Михайло Олександрович – д.т.н., професор, завідувач кафедри програмного забезпечення комп'ютерних систем, НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Алексєєв Олексій Михайлович – к.т.н., доцент, доцент кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Алдона Дерен – професор, Вроцлавський технологічний університет, Польща.

Басараб Данило Русланович – магістр групи 122м-22-4 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Берднік Михайло Геннадійович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Біжко Іван Сергійович – магістр групи 125м-22-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ведерников Денис Сергійович – магістр групи 121м-22-3 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Великих Олексій Віталійович – магістр групи КІ-22м-2 кафедри електроник обчислювальних машин Дніпровського національного університету ім.О.Гончара, м. Дніпро, Україна.

Візнюк Артем Валентинович (Artem Vizniuk) – магістр групи 121м-22-3 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Гай Олеся Володимирівна – магістр групи 126м-22з-1 кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Галушко Олександр Геннадійович – магістр групи 123м-22-1 кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Гнатушенко Володимир Володимирович – д.т.н., професор, завідувач кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Голінько Олександр Васильович (O.V. Holinko) – аспірант групи 122А-21-2 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Голобояр Євген Сергійович – магістр групи 122М-22-4 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Гресь Наталія Леонідівна – к.т.н., доцент, доцент кафедри міжнародних відносин і аудиту НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Гуліна Ірина Григорівна – к.п.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Деменков Сергій Олександрович – магістр групи 121М-22з-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Дмитрієва Ольга Анатоліївна – д.т.н., професор, Дослідницький центр моделюючих технологій (SimTech) університету Штутгарта, Інститут моделювання водно-екологічних систем університету Штутгарта, Німеччина. (Olga Dmytriyeva, d. of sc., prof., ¹Research center of simulation technologies (SimTech), University of Stuttgart, ²Institute for modelling hydraulic and environmental systems, University of Stuttgart)

Дяченко Григорій Григорійович (Grygorii Diachenko) – к.т.н., доцент, доцент кафедри електропривода НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Заболотнікова Валентина Василівна – старший викладач кафедри іноземних мов НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Зберовський Олександр Владиславович – д.т.н., професор, завідувач кафедри екології та охорони навколишнього середовища Дніпровського державного технічного університету, м. Кам'янське, Україна.

Іванченко Олег Васильович – д.т.н., доцент, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Йолкін Єгор Вадимович – магістр групи 122м-22-2 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кабак Леонід Віталійович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Каштан Віта Юріївна – к.т.н., доцент, доцент кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Коваленко Олександр Сергійович – магістр групи 121м-22-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кожевников Антон Вечеславович – к.т.н., доцент, доцент кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Коробко Ольга Валеріївна – асистент кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кострицька Світлана Іванівна – професор, завідувач кафедри іноземних мов НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кручинін Олександр Володимирович – магістр групи АМСЗІзм-22 кафедри інфокомунікаційної інженерії ім. В.В. Поповського, Харківський національний університет радіоелектроніки, Харків, Україна.

Куваєв Володимир Михайлович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Куракіна Анастасія Станіславівна – магістр групи 121м-22-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Лактіонов Іван Сергійович (Ivan Laktionov) – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ларикова Марія Володимірівна – магістр групи 122м-18-3 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Лисицький Олег Костянтинович – магістр групи 121м-22-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мартиненко Андрій Анатолійович – старший викладач кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мацюк Сергій Михайлович – к.т.н., доцент, доцент кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мещеряков Леонид Іванович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Молокова Світлана Валеріївна – магістр групи 126м-22-1 кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мороз Борис Іванович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мороз Дмитро Максимович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Нечай Наталія Михайлівна – старший викладач кафедри іноземних мов НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ніколаєнко Артем Віталійович – магістр групи 122м-22-3 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Одновол Микола Миколайович – доцент кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Петрушенко Артем В'ячеславович – магістр групи 121м-22-3 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Піньковська Анастасія Константинівна – магістр групи 126м-22-1 кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Положевець Юрій Іванович – магістр групи 122м-22-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Притула Микола Ігорович – аспірант групи 124м-22 Інститут прикладної математики і механіки НАН України, Слов'янськ, Україна.

Приходченко Сергій Дмитрович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Політов Артем Ігорович – магістр групи КС-23м кафедри електроник обчислювальних машин Дніпровського національного університету ім. О.Гончара, м. Дніпро, Україна.

Потемпа Михал – магістр-інженер, керівник фірми «GeoBit», Хшанув, Польща.

Різниченко Ігор Дмитрович – магістр групи КІ-22м-2 кафедри електроник обчислювальних машин Дніпровського національного університету ім.О.Гончара, м. Дніпро, Україна.

Родна Катерина Станиславовна – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Садиченко Данило Валерійович – магістр групи 122м-22-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Святошенко Володимир Олександрович – магістр групи АМСЗІзм-22 кафедри інфокомунікаційної інженерії ім. В.В. Поповського, Харківський національний університет радіоелектроніки, Харків, Україна.

Семенов Сергій Геннадійович – д.т.н., професор, Університет Комісії Національної освіти в Кракові, Польща (Sergey Semenjv, d. of sc., prof., University Education Commission , Krakow, Poland)

Сенченко Олексій Сергійович – канд. фіз-мат. наук, доцент, Інститут прикладної математики і механіки НАН України, Слов’янськ, Україна.

Сироткина Олена Ігорівна – к.т.н, доцент, School of Computer Science University of Windsor, Windsor, Ontario, Kanada.

Сконечный Ян – професор, Вроцлавський технологічний університет, Польща.

Слесарєв Володимир Вікторович – д.т.н., професор, професор кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Совенко Антон Сергійович – магістр групи АМСЗІзм-22 кафедри інфокомунікаційної інженерії ім. В.В. Поповського, Харківський національний університет радіоелектроніки, Харків, Україна.

Соколова Наталя Олегівна – к.т.н, доцент, доцент кафедри інформаційних технологій та комп’ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Спірінцев В’ячеслав Васильович – к.т.н., доцент, доцент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Спірінцева Ольга Володимирівна – к.т.н., доцент, доцент кафедри електроник обчислювальних машин Дніпровського національного університету ім. О.Гончара, м. Дніпро, Україна.

Сущевський Дмитро Валерійович – к.т.н, доцент, доцент кафедри інформаційних технологій та комп’ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Тимофєєв Дмитро Сергійович – магістр групи АМСЗІзм-22 кафедри інфокомунікаційної інженерії ім. В.В. Поповського, Харківський національний університет радіоелектроніки, Харків, Україна.

Тріско Антон Євгенович – магістр групи 1221м-22-2 кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Турчина Інесса Геннадіївна – магістр групи 126м-22з-1 кафедри інформаційних технологій та комп’ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Удовик Ірина Михайлівна – к.т.н., декан факультету інформаційних технологій, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Удовик Олександр Васильєвич – студент третього (освітньо-наукового) рівня вищої освіти за спеціальністю 141 «Електроенергетика, електротехніка та електромеханіка» кафедри електротехніка та електромеханіка НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Уланова Наталія Петрівна – к.т.н., доцент, доцент кафедри вищої математики НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ус Світлана Альбертовна – канд. фіз.-мат. наук, професор, кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Хазова Оксана Валеріївна – старший викладач кафедри іноземних мов НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Хара Герман Леонідович – магістр групи 126м-22-1 кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Харь Алена Тарасовна – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Хархула Ольга Юріївна – магістр групи 124м-22-1 кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Цалюк Максим Семенович – магістр групи 122м-22-4 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Швачич Геннадій Григорович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Шевченко Ілля Віталійович – магістр групи 122м-22-2 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Шевцова Ольга Сергіївна – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Щербина Павло Олександрович – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ширін Артем Леонідович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Шпорта Анна Григорівна – к.ф.-м.н., доцент, доцент кафедри вищої математики НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Шрамко Даріна Ігорівна – магістр групи 125м-22-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Юдина Анастасія Олегівна – магістр групи 124м-22-1 кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

ЗМІСТ

Розділ 1 МІЖНАРОДНЕ СПІВРОБІТНИЦТВО У СФЕРІ ОСВІТИ, НАУКИ І ВИРОБНИЦТВА.....	3
1. Мороз Б.І., Сироткина О.І., Кострицька С.І., Ларикова М.В. Selenium vs. playwright: exploring end-to-end testing in multi-threaded environment... 3	3
2. Kozhevnykov A., Dmytriyeva O. Creating using the mathcad system of laboratory experimentation on the subject "Numerical methods in computer science"..... 9	9
3. Приходченко С.Д., Семенов С.Г., Родна К.С., Лисицький О.К. Вивчення методів гейміфікації та розробка інноваційних гейміфікаційних підходів..... 15	15
4. Ширін А. Л., Потемпа М., Гуліна І.Г, Петрушенко А. В., Методи управління станом фреймворку Flutter..... 19	19
5. Кабак Л.В., Мороз Б.І., Сконеchnый Я., Сіда І. А. The modern popularity of non-relational databases 26	26
6. Holinko O.V. Information support for the process of diagnosing the condition of methane analyzers..... 29	29
7. Приходченко С.Д., Алдона Дерен, Мартиненко А.А., Батальський М.А. JDBC database interaction method..... 32	32
Розділ 2 ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ.....	36
8. Алексеев М.О., Алексеев О.М., Голобояр Є.С. Архітектурне рішення в мобільних додатках під платформу iOS через інтеграцію SwiftUI..... 36	36
9. Куваєв В.М., Мещеряков Л.І., Харь А.Т., Політов А.І. Організація людино-машинного інтерфейсу підтримки технічного забезпечення систем керування технологічним процесом..... 40	40
10. Каштан В.Ю., Гнатушенко В.В., Коробко О.В. , Гай О.В. Інформаційна технологія моніторингу вулканічної активності в Ла-Пальмі..... 45	45
11. Кабак Л.В., Мороз Д.М., Буток В.О. Дослідження сучасних технологій управління особистими фінансами за допомогою аналітичних інструментів 48	48
12. Спірінцев В.В., Басараб Д.Р. Керування функціями відеодзвінка за допомогою розпізнавання жестів в режимі реального часу..... 52	52
13. Каштан В.Ю., Сущевський Д.В., Молокова С.В. Інформаційна технологія дослідження просторово-часових змін берегової лінії тузловських лиманів..... 57	57
14. Ус С.А., Юдина А.О. Дослідження методів прогнозування для аналізу розвитку ветеринарних послуг..... 61	61
15. Кабак Л.В., Радостев А.О. Особливості сучасних баз даних та можливості їх використання в інформаційних системах різного типу..... 64	64
16. Спірінцева О.В., Великих О.В. Розробка та дослідження особистого освітнього блогу на платформі React Native..... 69	69
17. Хара Г.Л., Соколова Н.О. Інформаційна система тестування веб-додатків..... 71	71

18. Каштан В.Ю., Коробко О.В., Турчина І.Г. Комп'ютерна технологія моніторингу стану лісового покриву природного заповідника "древлянський" за даними sentinel-2.....	74
19. Спірінцев В.В., Родна К.С., Положевець Ю.І. Розробка та дослідження месенджера на базі технології блокчейн.....	78
20. Піньковська А.К., Соколова Н.О. Розробка інформаційної технології конфігурування компонентів автоматизованої системи виробництва.....	82
Розділ 3 ПРОГРАМНІ ЗАСОБИ УПРАВЛІННЯ, ЗБОРУ, ОБРОБКИ І ПЕРЕДАЧІ ІНФОРМАЦІЇ.....	85
21. Удовик І.М., Мещеряков Л.І., Швачич Г.Г., Деменков С.О. Реалізація автоматичної генерації коду для ORM-системи ActiveJDBC з використанням технології Anotation Processing.....	85
22. Спірінцев В.В., Хазова О.В., Садиченко Д.В. Дослідження та порівняння ефективності балансувальників навантаження	92
23. Куваєв В.М., Мещеряков Л.І., Харь А.Т., Політов А.І. Особливості реалізації людино-машинного інтерфейсу при підвищених вимогах до реакції оператора на позаштатні ситуації.....	97
24. Спірінцев В.В., Цалюк М.С. Забезпечення ефективного менеджменту стану додатку: роль State Management в розвитку мобільних додатків Flutter	101
25. Приходченко С.Д., Гресь Н.Л., Шпорта А.Г., Тріско А.Є. Дослідження способів та принципів взаємодії з Artificial Reality на прикладі ігрового додатку	106
26. Спірінцева О.В., Різниченко І.Д. Розробка веб-додатку для керування технологічними процесами з використанням специфікації OPC UA.....	110
27. Бердник М.Г., Іванченко О.В., Коваленко О.С. Програмне забезпечення розрахунку раціону харчування для хворих на цукровий діабет другого типу	112
28. Галушко О.Г., Соколова Н.О. Розробка серверного кластеру на основі програмного забезпечення з відкритим кодом.....	117
29. Спірінцев В.В., Щербина П.О., Шевченко І.В. Забезпечення ефективної інтеграції: роль RESTful API в розвитку мобільних додатків для OpenCart	120
Розділ 4 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ ДЛЯ ВИРІШЕННЯ ЗАВДАНЬ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ.....	125
30. Слесарєв В.В., Сенченко О.С., Притула М.І. Моделювання лінгвістичного представлення детермінованих графів з використанням програмних засобів.....	125
31. Artem Vizniuk, Ivan Laktionov, Grygorii Diachenko Study of the effectiveness of ml algorithms in predicting the probability of crop disease occurrence.....	132
32. Ус С.А., Хархула О.Ю. Дослідження та ранжування альтернатив пластикової упаковки.....	136
33. Приходченко С.Д., Шевцова О.С., Левдик І.А. Дослідження недоліків робочого процесу та робочої продуктивності з використанням поточних засобів ведення технічних інструкцій.....	140

34.	Мещеряков Л.І., Одновол М.М., Уланова Н.П., Ведерников Д.С. Розробка та дослідження ефективності впровадження програмного забезпечення реалізації модифікованого методу відновлення формальних граматик.....	145
35.	Приходченко С.Д., Удовик О.В., Мацюк С.М., Йолкін Є.В. Дослідження та оптимізація бізнес-процесів з використанням хмарних сховищ.....	152
36.	Спірінцев В.В., Ніколаєнко А.В. Застосування моделей латентної дифузії в контексті генеративного мистецтва.....	156
37.	Бердник М.Г., Куракіна А.С. Веб-додаток оптимізації набору продуктів для збалансованого харчування.....	164
38.	Заболотнікова В.В., Нечай Н.М. Enhancing professional writing skills: teaching students to create effective prompts for ai.....	170
	Розділ 5 ІНФОРМАЦІЙНА БЕЗПЕКА.....	173
39.	Тимофєєв Д.С., Совенко А.С. Оцінка ризиків в процесі управління ризиками кібербезпеки пов'язаними з людським фактором.....	173
40.	Кручінін О.В., Тимофєєв Д.С. Класифікатори мережових атак.....	177
41.	Святошенко В.О., Кручінін О.В. Методи машинного навчання в системах виявлення вторгнень.....	182
42.	Кабак Л.В., Мороз Б.І., Гарбуз В.О. Сучасна актуальність використання шифрування інформації.....	186
43.	Святошенко В.О., Совенко О.С. Особливості машинного навчання для систем виявлення вторгнень в мережах інтернету речей.....	190
44.	Кручінін О.В., Шрамко Д.І. Методика використання OSINT під час тестування на проникнення.....	193
45.	Тимофєєв Д.С., Совенко А.С. Мінімізація ризиків пов'язаних з персоналом в процесі впровадження системи управління інформаційною безпекою відповідно до вимог стандарту ISO/IEC 27001:2022.....	195
46.	Біжко І.С. Особливості впровадження системи управління інформаційною безпекою на сільськогосподарських підприємствах.....	200
47.	Кручінін О.В., Святошенко В.О. Характеристика атак на системи виявлення вторгнень на основі машинного навчання.....	204
	ВІДОМОСТІ ПРО АВТОРІВ.....	208

Наукове видання

XVIII Міжнародна конференція

**ПРОБЛЕМИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У
СФЕРІ ОСВІТИ, НАУКИ ТА ПРОМИСЛОВОСТІ**

Збірник наукових праць
Національний технічний університет «Дніпровська політехніка»,
№ 8
(Українською мовою)

Відповідальний за випуск Л.І. Мещеряков

Видано в редакції авторів публікацій.

Підп. до друку 25.12.2023. Формат 30×42/4.
Папір офсет. Ризографія. Ум. друк. арк. 10,7.
Обл.-вид. арк. 13,6. Тираж 100 пр. Заказ № 847

Підготовлено до друку та видруковано
НТУ «Дніпровська політехніка»,
Свідоцтво про внесення до Державного реєстру ДК №1842 від 11.06.2004 р.