

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА ТЕОРІЯ КОДУВАННЯ

УДК 004.92

Є. К. Завальнюк, О. Н. Романюк, В. В. Войтко, О. В. Романюк, А. В. Снігур

РОЗРОБКА МОДИФІКОВАНОЇ МОДЕЛІ ШЛІКА ДЛЯ
ВИЗНАЧЕННЯ СПЕКУЛЯРНОЇ СКЛАДОВОЇ КОЛЬОРУ

Вінницький національний технічний університет, Вінниця

Анотація. Проаналізовано найпоширеніші дистрибутивні функції відбивної здатності (ДФВЗ), до числа яких відноситься функція Шліка. Ця функція порівняно з ДФВЗ Блінна та Фонга має меншу степінь, що дає можливість її використання у засобах динамічної комп'ютерної графіки. Функція Шліка для обчислення використовує коефіцієнт спекулярності поверхні та косинус кута між вектором нормалі до поверхні і серединним вектором. Недоліком функції є нереалістичне відтворення відблиску у зоні затухання.

З метою реалістичного відтворення як епіцентру відблиску, так і його блумінгу запропоновано нову модифікацію функції Шліка з використанням поправочних коефіцієнтів і степенів. Розроблено програму підбору степенів і коефіцієнтів. Наведено граф-схему програми з детальним описом. Засобами Mathcad 2001 Professional здійснено порівняльний аналіз ДФВЗ Шліка, Блінна та модифікованої функції. Доведено перевагу модифікованої функції порівняно з ДФВЗ Шліка у швидкості падіння в зоні затухання.

Встановлено менше значення відносно похибки апроксимації модифікованої ДФВЗ Шліка на рівні епіцентру відблиску. Побудовано графіки залежностей максимальних відносних й абсолютних похибок апроксимації модифікованої ДФВЗ Шліка порівняно з ДФВЗ Блінна залежно від коефіцієнта спекулярності поверхні. Знайдено аргумент, при якому модифікована ДФВЗ досягає мулевого рівня. Доведено монотонне спадання запропонованої ДФВЗ на проміжку аргументів $[0, \pi/2]$. Для запропонованої ДФВЗ, ДФВЗ Шліка та Блінна розраховано інтервали обчислення спекулярної складової кольору. Граничні значення інтервалів використано при знаходженні відношень розмірів плям відблисків для модифікованої ДФВЗ і ДФВЗ Блінна, Шліка та Блінна. Відношення розмірів плям для ДФВЗ проілюстровані тривимірними графіками. Встановлено, що зона блумінгу для модифікованої ДФВЗ обчислюється на меншому проміжку, ніж для ДФВЗ Шліка. Модифікована ДФВЗ може бути використана в системах високопродуктивного рендерингу та забезпечує підвищену реалістичність формування відблисків порівняно з ДФВЗ Шліка.

Ключові слова ДФВЗ Шліка, ДФВЗ Фонга, ДФВЗ Блінна-Фонга, затухання відблиску, епіцентр відблиску.

Abstract. The most common bidirectional reflectance distribution functions (BRDF), among which the Schlick function is included, were analyzed. Compared to Blinn and Phong's BRDF, this function has a smaller degree, which makes it possible to use it in dynamic computer graphics tools. The Schlick function uses the specular coefficient of the surface and the cosine of the angle between the vector normal to the surface and the median vector for calculation. The disadvantage of the function is the unrealistic reproduction of glare in the attenuation zone. In order to realistically reproduce both the glare epicenter and its blooming, a new modification of the Schlick function using correction coefficients and exponents is proposed. A program for selecting powers and coefficients has been developed. A diagram of the program with a detailed description is given. Mathcad 2001 Professional tools were used to perform a comparative analysis of the Schlick, Blinn BRDFs and modified function. The advantage of the modified function compared to Schlick's BRDF in the fall speed in the attenuation zone is proved. A smaller value of the relative error of approximation of the modified Schlick BRDF at the level of the glare epicenter was proven. Graphs of the dependences of the maximum relative and absolute approximation errors of the modified Schlick's BRDF compared to Blinn's BRDF depending on the specular coefficient of the surface are plotted. An argument was found in which the modified BRDF reaches the sludge level. The monotonic decrease of the proposed BRDF on the interval of arguments $[0, \pi/2]$ is proven. The intervals for calculating the specular color component were calculated for the proposed BRDF, Schlick and Blinn BRDFs. The limit values of the intervals were used when finding the ratio of the size of the glare spots for the modified BRDF and Blinn BRDF, Schlick and Blinn BRDFs. Spot size ratios for BRDFs are illustrated by three-dimensional graphs. It was proven that the blooming zone for the modified BRDF is calculated on a shorter interval than for Schlick's BRDF. The modified BRDF can be used in high-performance rendering systems and provides increased realism in the formation of reflections compared to Schlick's BRDF.

Key words: Schlick BRDF, Phong BRDF, Blinn-Phong BRDF, glow attenuation, glow epicenter.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-4-12>.

Вступ

При формуванні тривимірних графічних сцен особливу увагу приділяють реалістичності відтворення складових об'єктів. При цьому необхідно точно відтворити як форму і конструкторські особливості, так і візуальні характеристики поверхонь. Важливим для реалістичності є відтворення відблисків, за які відповідає спекулярна складова кольору. При відтворенні відблисків використовуються моделі відбиття на основі двонаправлених функцій. Використання моделей Блінна і Фонга потребує значних обчислювальних витрат, що позначається на швидкості формування графічних сцен. Тому актуальною задачею є розробка нової дистрибутивної функції, яка має меншу обчислювальну складність і з достатньою точністю відтворює як епіцентр відблиску, так і його блумінг.

Аналіз літературних джерел

За оптичні властивості поверхні відповідає двопробенева дистрибутивна функція відбивної здатності ДФВЗ (BRDF – Bidirectional Reflectance Distributive Function). Методи, які використовуються для моделювання освітлення, оперують із розсіяним і відбитим світлом. Відбите світло у своєму складі має дві компоненти: дифузну та спекулярну [1]. Розсіяне (фонове) світло – це світло, яке відбивається від навколишнього середовища, сусідніх об'єктів [2].

При наявності в сцені точкового джерела світла, інтенсивність дифузійного відбиття пропорційна косинусу кута між нормаллю до поверхні й напрямком на джерело світла. У цьому випадку для обчислення інтенсивності дифузійного відбиття застосовують закон косинусів Ламберта [3].

У більшості випадків дистрибутивна функція визначається через скалярний добуток відповідних нормованих векторів. Значно рідше оперують зі значенням кута між векторами, оскільки це передбачає виконання обернених тригонометричних функцій.

Нехай I – інтенсивність джерела світла, \vec{L} , \vec{V} – відповідно вектори до джерела світла та спостерігача, \vec{N} – вектор нормалі до поверхні (рисунок 1).

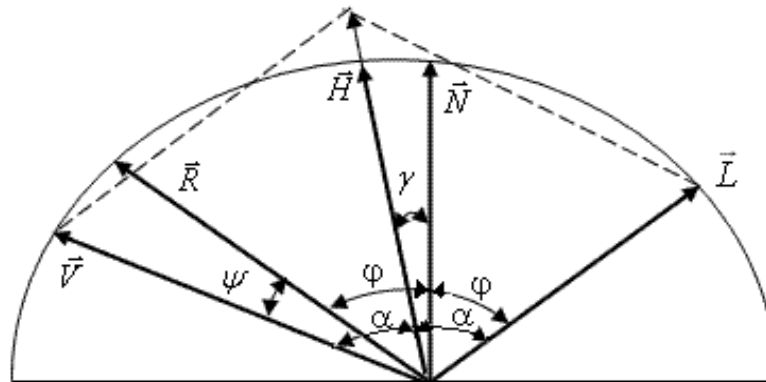


Рисунок 1 – Вектори нормалей до точки поверхні

У комп'ютерній графіці найбільш часто використовують модель визначення спекулярної складової кольору Б. Фонга [3], що має вигляд

$$I_s = I \cdot \cos^n \alpha. \quad (1)$$

У ДФВЗ Фонга $\cos \alpha = \vec{V} \cdot \vec{R}$, де $\vec{R} = 2 \cdot (\vec{L} \cdot \vec{N}) \cdot \vec{N} - \vec{L}$. Вектор \vec{R} називають вектором дзеркального відбиття. У дистрибутивній функції Блінна, яка історично з'явилася пізніше ДФВЗ Фонга, замість $\cos \alpha$ використовують $\cos \gamma = \vec{N} \cdot \vec{H}$, де $\vec{H} = \vec{L} + \vec{V} / |\vec{L} + \vec{V}|$ [4]. Вектор \vec{H} називають серединним, оскільки його напрям співпадає з бісектрисою кута між векторами \vec{L} , \vec{V} .

У комп'ютерній графіці іноді для формування відблисків використовують ДФВЗ Гаусса [3], яка має такий вигляд $e^{-\frac{n \cdot (\angle(\vec{H}, \vec{L}))^2}{2}}$. Така дистрибутивна функція є достатньо точною, реалістично відтворює як зону епіцентру відблиску, так і його затухання (блюмінгу), однак отримала широке поширення тільки для аналізу зображень, оскільки передбачає використання обернених функцій для обчислення кута між векторами нормалей

У сучасних пакетах тривимірної графіки широкого поширення отримала дистрибутивна функція Шліка [1], що обчислюється [5]

$$\cos \gamma / (n - n \cos \gamma + \cos \gamma). \quad (2)$$

ДФВЗ Шліка має значно меншу обчислювальну складність порівняно з функціями Фонга та Блінна. Аналіз функції показав, що за областю епіцентру відблиску спостерігається суттєве розходження з результатами, отриманими згідно з моделлю освітлення Блінна. Суттєвим недоліком ДФВЗ Шліка [3] є те, що дистрибутивна функція в зоні формування блюмінга падає до нульового рівня надзвичайно повільно, що обумовлює неприродне освітлення графічного об'єкта та додаткові обчислення за рахунок збільшення інтервалу зміни аргументу.

Низька степінь функції Шліка обумовлює можливість її використання в засобах динамічної комп'ютерної графіки, але невисока точність відтворення відблисків є перешкодою використання для високореалістичного відтворення графічних сцен.

Тому актуальною задачею є модифікація ДФВЗ Шліка з метою більш реалістичного відтворення відблисків.

Мета статті

Мета статті – розробка модифікованої ДФВЗ Шліка, яка з достатньою точністю відтворює спекулярну складову кольору.

Розробка модифікованої ДФВЗ Шліка

Розглянемо модифікацію ДФВЗ Шліка. Знайдемо нову дистрибутивну функцію відбивної здатності поверхні у вигляді

$$\frac{a_1 \cos(x)^{b_1}}{a_2 (a_3 n^{b_3} - a_4 (n * \cos(x))^{b_4} + a_5 \cos(x)^{b_5})^{b_2}}. \quad (3)$$

Знайдемо невідомі змінні b_n (степені) і a_n (множники).

Оскільки при $x=0$, то $\cos(x)$ дорівнює 1, тому у запропонованій ДФВЗ Шліка $n - n \cos(x) = 0$. Звідси $b_4 = b_3, a_4 = a_3$. З урахуванням останнього, отримуємо такий вигляд запропонованої ДФВЗ

$$\frac{a_1 \cos(x)^{b_1}}{a_2 (a_3 n^{b_3} - a_3 (n * \cos(x))^{b_3} + a_5 \cos(x)^{b_5})^{b_2}}. \quad (4)$$

Позначимо запроповану ДФВЗ як F_{MSH} .

Розроблено програму для підбору коефіцієнтів a_n й степенів b_n , граф-схему якої наведено на рисунку 2.

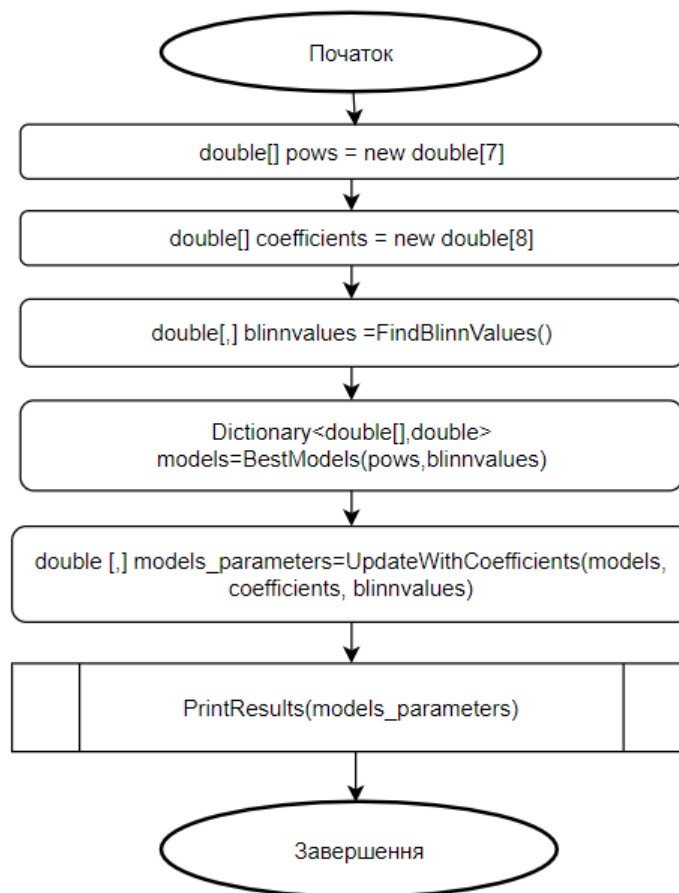


Рисунок 2 – Блок-схема комп'ютерного перебору параметрів F_{MSH}

Для спрощення визначення невідомих в F_{MSH} розділимо процедури підбору коефіцієнтів і степенів. Підберемо степені $b_1 - b_5$. На даному етапі значення коефіцієнтів $a_1 - a_5$ приймемо рівними одиниці.

Враховуючи швидке зростання обчислювальної складності перебору та необхідність низької обчислювальної складності функції, значення масиву можливих степенів $rows$ виберемо з інтервалу $[-3,3]$.

Масив можливих коефіцієнтів $coefficients$ містить значення: 0.25, 0.5, 1, 1.25, 1.5, 2, 2.5, 3.

Далі у подвійному циклі для кутів $[0,90]$ і коефіцієнтів спекулярності поверхні n $[1,1000]$ (всередині методу `FindBlinnValues`) викликається функція обчислення значень ДФВЗ Блінна. Отримані значення формують двовимірний масив `blinnvalues`.

У чотиридимірному циклі всередині функції `BestModels` (відповідно до 4 невідомих степенів) для n і кутів заповнюється двовимірний масив значень F_{MSH} з випадковими степенями. Обчислюється середнє значення абсолютних відхилень масиву від `blinnvalues`. У циклах формується список п'яти наборів степенів з найменшим середнім абсолютним відхиленням. Список записується у словник `models` (ключ словника – набір степенів, значення – відхилення).

У чотиридимірному (відповідно до 4 невідомих коефіцієнтів) циклі функції `UpdateWithCoefficients` випадкові набори коефіцієнтів поєднуються зі знайденими п'ятьма наборами степенів. Для кожного набору степенів визначається набір коефіцієнтів, що забезпечує найменше середнє абсолютне відхилення від `blinnvalues`.

Функція повертає двовимірний масив, де рядки представляють інформацію сукупності параметрів F_{MSH} : середнє абсолютне відхилення від ДФВЗ Блінна, п'ять значень степенів, п'ять значень коефіцієнтів. Здійснюється передача результату функції масиву `models_parameters`.

Функція `PrintResults` здійснює виведення інформації про отримані моделі (набори шуканих параметрів).

Проаналізуємо отримані результати (таблиця 1). Найвищу точність має модель 1. Однак модель вимагає 5 піднесень до вищих степенів. Модель 0 є другою за точністю. Для неї необхідно 3 піднесення до вищих степенів. Модель 2 є компромісною за точністю та потребує лише одне піднесення. Тому, враховуючи точність й простоту моделей, обрано модель 2.

Таблиця 1 – Отримані параметри для F_{MSH}

№	Степені					Коефіцієнти					Середнє абсолютне відхилення від ДФВЗ Блінна
	2	2	1	1	2	2	1.25	1	1	1.25	
0	2	2	1	1	2	2	1.25	1	1	1.25	0.0063
1	3	2	1	1	3	2	1.25	1	1	1.25	0.0062
2	1	2	1	1	1	2	1.25	1	1	1.25	0.0066
3	2	2	1	1	3	2	1.25	1	1	1.25	0.0066
4	1	2	1	1	2	2	1.25	1	1	1.25	0.0070

Для розрахунку спекулярної складової найбільш оптимальним є такий вид F_{MSH}

$$\frac{2\cos x}{\left(1 + \frac{1}{4}\right) \left(n - n * \cos x + \left(1 + \frac{1}{4}\right) \cos x\right)^2} \cdot \quad (5)$$

Проаналізуємо запропоновану ДФВЗ. Позначимо ДФВЗ Шліка як F_{SH} , ДФВЗ Блінна як F_B .

На рисунку 3 зображено графіки F_{MSH} , F_{SH} , F_B . З наведених графіків видно, що запропонована F_{MSH} забезпечує краще відтворення блюмінгу відблиску порівняно з F_{SH} , що дає можливість усунути її суттєвий недолік – неприродне відтворення зони затухання відблиску та надлишкові обчислення.

Порівняємо максимальні відносні похибки F_{MSH} і F_{SH} (δ_{msh} , δ_{sh} відповідно) при апроксимації F_B у точці, де значення $F_B = 0.3$. Цей рівень відповідає за епіцентр відблиску.

Знайдемо абсцису цієї точки (x_{ep}) для еталонної функції F_B , яка дорівнює :

$$a \cos\left(\exp\left(\frac{-1.2}{n}\right)\right). \quad (6)$$

При n з проміжку $[2, 1000]$ δ_{msh} суттєво менше δ_{sh} (не більше 16% проти не більше 52%) (рисунки 3, 4).

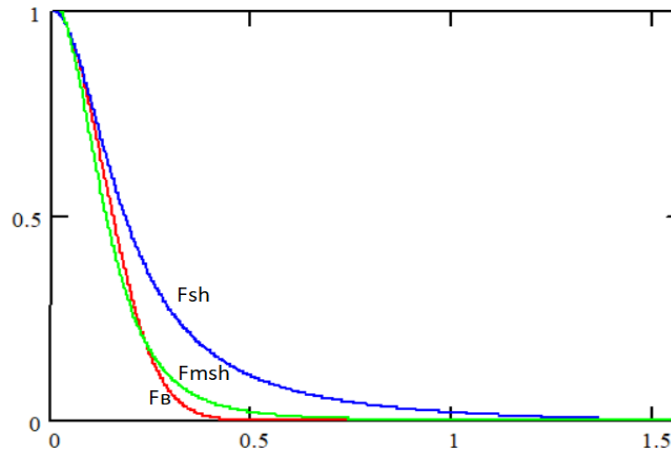


Рисунок 3 – Графіки F_{MSh} , F_{Sh} , F_B ($n=60$)

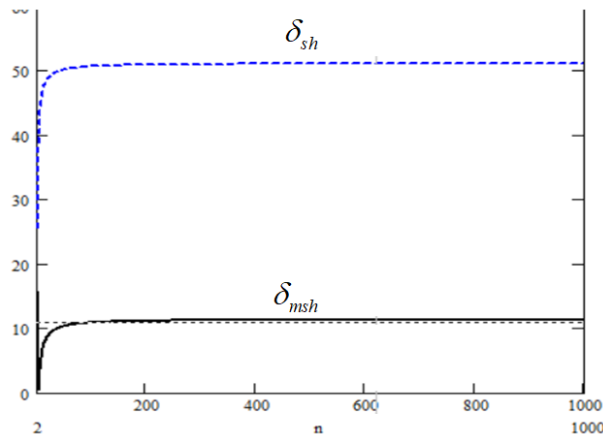


Рисунок 4 – δ_{msh} та δ_{sh} у точці x_{ep}

Формула для обчислення модуля абсолютної похибки при апроксимації F_B має вигляд

$$|F_{apr} - \cos(x)^n|, \tag{7}$$

де F_{apr} – апроксимація F_B .

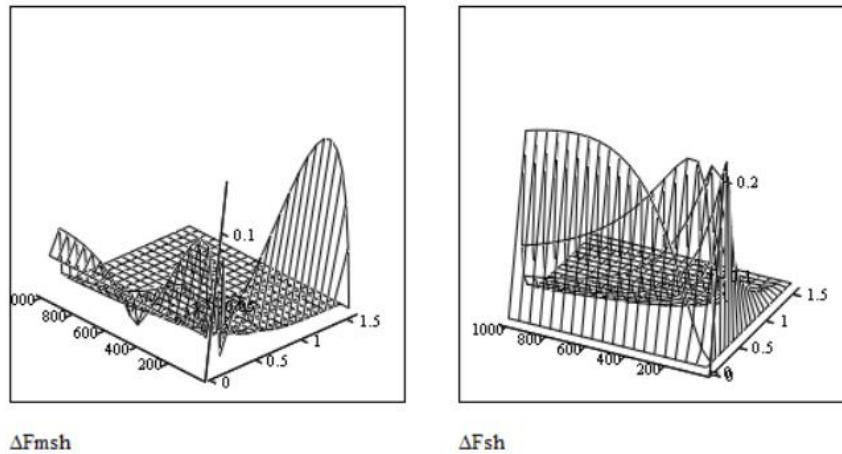
Побудуємо тривимірні графіки (рисунки 5) абсолютних похибок F_{MSh} , F_{Sh} від F_B . ($\Delta_{F_{MSh}}$, $\Delta_{F_{Sh}}$ відповідно) залежно від n і кута.

З графіків встановлюємо, що найбільша $\Delta_{F_{MSh}}$ менша за найбільшу $\Delta_{F_{Sh}}$ приблизно удвічі.

Знайдемо значення аргументів, за яких F_{MSh} досягає нульового значення. Для цього прирівнюємо F_{MSh} до нуля. Знаходимо, що F_{MSh} досягає нуля при $x=1.570796$, що наближено дорівнює $\frac{\pi}{2}$.

Доведемо, що функція F_{MSh} монотонно спадає. Для цього знайдемо похідну, яка дорівнює:

$$\frac{-1.6 * \sin(x)}{(n - n * \cos(x) + 1.25 \cos(x))^2} - \frac{3.2 * \cos(x)}{(n - n * \cos(x) + 1.25 * \cos(x))^3} (n \sin(x) - 1.25 \sin(x)) \tag{8}$$

Рисунок 5 – Графіки $\Delta_{F_{MSH}}$, $\Delta_{F_{SH}}$ відносно n та кута

На рисунку 6 наведено графік зміни похідної від n та кута x .

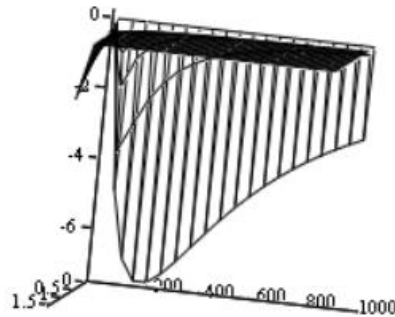


Рисунок 6 – Графік похідної функції

Функція монотонно спадає, оскільки значення її похідної від'ємне.

Обчислимо співвідношення розмірів відблисків F_{MSH} і F_{SH} відносно F_B . Введемо граничне значення 2^{-q} , яке визначає аргумент, за межами якого розрахунок ДФВЗ припиняється.

Спершу запишемо нерівність для F_{SH} , яка визначає інтервал розрахунку аргументу для визначення спекулярної складової кольору.

$$0 \leq \frac{\cos x}{n - n \cos(x) + \cos(x)} \leq 2^{-q}. \quad (9)$$

У результаті розв'язання нерівності отримуємо:

$$0 \leq x \leq a \cos\left(\frac{n}{2^q + n - 1}\right). \quad (10)$$

Аналогічно запишемо нерівність для визначення інтервалу зміни значення для F_{MSH} :

$$0 \leq \frac{2 \cos x}{1.25(n - n \cos(x) + 1.25 \cos(x))^2} \leq 2^{-q}. \quad (11)$$

Знаходимо значення кута x :

$$0 \leq x \leq a \cos \left(\frac{4 \left(16 * 2^q + 20n^2 - 25n - 4 * 2^{q+0.5} (8 + 20 * 2^{-q} n^2 - 25 * 2^{-q} n)^{\frac{1}{2}} \right)}{5(16n^2 - 40n + 25)} \right) \quad (12)$$

Для F_B нерівність має такий вигляд

$$0 \leq \cos(x)^n \leq 2^{-q}. \quad (13)$$

Результатом розв'язку є

$$0 \leq x \leq a \cos \left(2^{\frac{-q}{n}} \right). \quad (14)$$

Обчислимо \mathfrak{R} . Для F_{SH} це значення розраховується так

$$\frac{a \cos \left(\frac{n}{2^q + n - 1} \right)}{a \cos \left(2^{\frac{-q}{n}} \right)} \quad (15)$$

Знайдемо \mathfrak{R} для F_{MSH}

$$\frac{a \cos \left(\frac{4 \left(16 * 2^q + 20n^2 - 25n - 4 * 2^{q+0.5} (8 + 20 * 2^{-q} n^2 - 25 * 2^{-q} n)^{\frac{1}{2}} \right)}{5(16n^2 - 40n + 25)} \right)}{a \cos \left(2^{\frac{-q}{n}} \right)} \quad (16)$$

На рисунку 7 наведено графіки відношень \mathfrak{R} для F_{MSH} та F_B (А), F_{SH} і F_B (В).

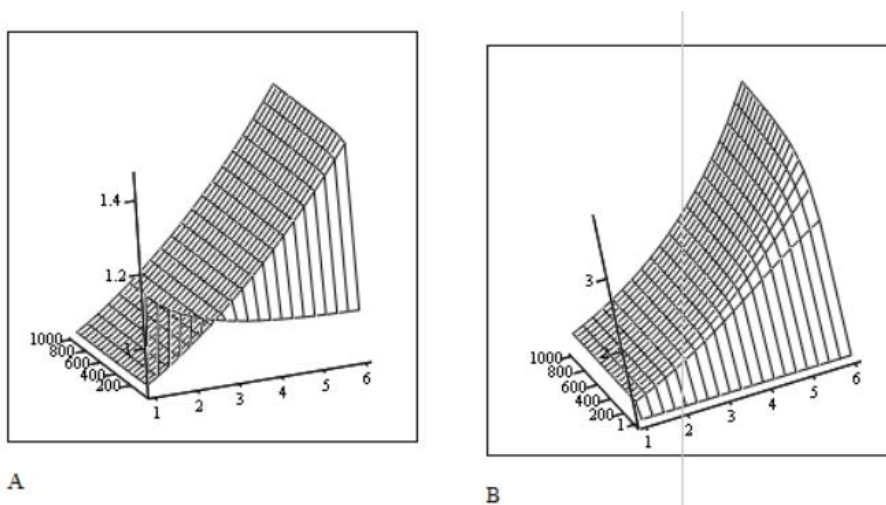


Рисунок 7 – Графіки відношень \mathfrak{R} для F_{MSH} і F_B (А), F_{SH} і F_B (В)

Отже, для розробленої функції блюмінг відповідає інтервалу, що зменшує обсяги обчислень.

Розраховані граничні значення аргументів ДФВЗ використаємо при побудові графіків максимальних відносних похибок (δ) F_{MSH} , F_{SH} залежно від n .

Вираз для визначення δ апроксимації ДФВЗ відносно F_B має такий вигляд

$$\left| \frac{100(F_{apr} - \cos(x)^n)}{\cos(x)^n} \right|. \quad (17)$$

Для обчислення максимального δ апроксимації ДФВЗ по кожному n з врахуванням граничних значень у Mathcad 2001 було розроблено програму, граф-схема програми наведена на рисунку 8.

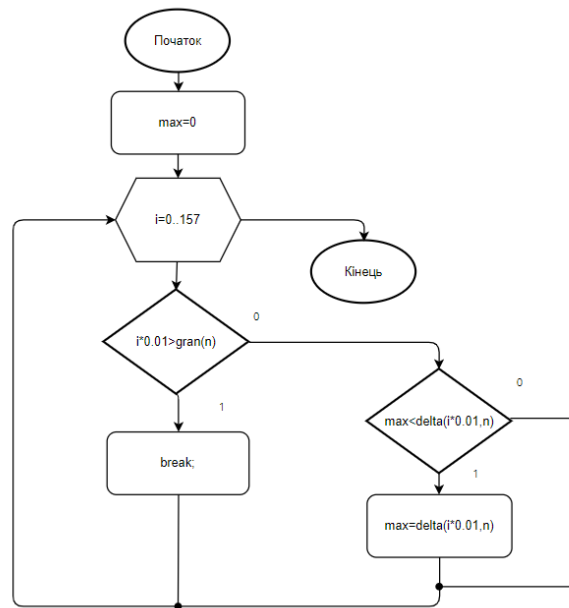


Рисунок 8 – Граф-схема програми обранку максимальної δ для визначеного n

У циклі множник $i*0.01$ відповідає значенню кута в радіанах та порівнюється із обчисленим граничним значенням розрахунку для ДФВЗ $gran(n)$. Якщо кут більший, здійснюється вихід з циклу оператором `break`. Інакше тимчасове максимальне значення δ (змінна `max`) порівнюється з обчисленим δ для нового кута (результат функції `delta`). Якщо значення `delta` більше, змінна `max` оновлюється.

На рисунку 9 зображено побудовані графіки максимальних δ F_{MSH} , F_{SH} відносно n . Значення для кожного n обчислені через розроблену програму.

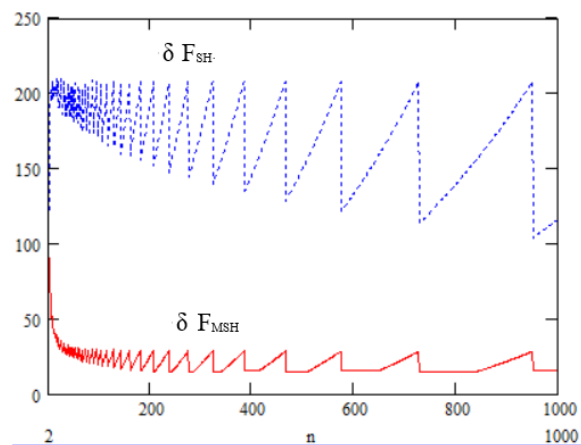


Рисунок 9 – Залежність δ F_{MSH} і δ F_{SH} від n

З побудовано графіка встановлюємо, що максимальні δF_{MSH} є меншими, ніж максимальні δF_{SH} у 1,5 – 7 разів для послідовності n.

Висновок

Запропоновано нову модифікацію ДФВЗ Шліка, здійснену шляхом комп'ютерного перебору. Нова модифікація зберігає обчислювальні переваги ДФВЗ Шліка та підвищує реалістичність відтворення відблисків у зоні блонінгу. Показано переваги модифікованої ДФВЗ над ДФВЗ Шліка за параметрами: максимальна відносна похибка, максимальна абсолютна похибка, відносна похибка на рівні епіцентру відблиску, розмір плями відблиску.

Список літератури

- [1] О. Н. Романюк, та А. В. Чорний, *Високопродуктивні методи та засоби зафарбовування тривимірних графічних об'єктів*. Вінниця, Україна: УНІВЕСУМ-Вінниця, 2006.
- [2] О. Н. Романюк, *Комп'ютерна графіка. Навчальний посібник*. Вінниця: УНІВЕСУМ-Вінниця, 2001.
- [3] О. Н. Романюк, "Класифікація дистрибутивних функцій відбивної здатності поверхні", *Наукові праці Донецького національного технічного університету. Сер.: Інформатика, кібернетика та обчислювальна техніка*, вип. 9. с. 145-151. 2008.
- [4] O. Romanuyk, and A. Chernij, "Methods for Specular Color Component Accelerate Calculation", *IEEE Workshop on Intelligent Data Acquisition and Advanced Computing systems: Technology and Applications*. Sofia, pp. 615-619. 2005.
- [5] О. Н. Романюк, "Альтернативна реалізація дистрибутивної двопроменевої функції для моделей освітлення Бліна та Фонга", *Наукові праці Донецького національного технічного університету. Серія «Обчислювальна техніка та автоматизація»*, випуск 106, с. 151-156. 2006.

Стаття надійшла: 22.09.2022.

References

- [1] O. N. Romanyuk, ta A. V. Chorny, *Vysokoproduktyvni metody ta zasoby zafarbovuvannya tryvymirnykh hrafichnykh obyektiv*. Vinnytsya, Ukrayina: UNIVESUM-Vinnytsya, 2006 [in Ukrainian].
- [2] O. N. Romanyuk, *Kompyuterna hrafika. Navchalnyu posibnyk*. Vinnytsya: UNIVESUM-Vinnytsya, 2001 [in Ukrainian].
- [3] O. N. Romanyuk, "Klasyfikatsiia dystributyvnykh funktsii vidbyvnoi zdatnosti poverkhni", *Naukovi pratsi Donetskoho natsionalnoho tekhnichnoho universytetu. Ser.: Informatyka, kibernetyka ta obchysliuvalna tekhnika*, vyp. 9. s. 145-151. 2008 [in Ukrainian].
- [4] O. Romanuyk, and A. Chernij, "Methods for Specular Color Component Accelerate Calculation", *IEEE Workshop on Intelligent Data Acquisition and Advanced Computing systems: Technology and Applications*. Sofia, pp. 615-619. 2005.
- [5] O. N Romanyuk, "Alternatyvna realizatsiia dystributyvnoi dvopromenevoi funktsii dlia modelei osvittennia Blina ta Fonha", *Naukovi pratsi Donetskoho natsionalnoho tekhnichnoho universytetu. Seriia «Obchysliuvalna tekhnika ta avtomatyzatsiia»*, vypusk 106, s. 151-156. 2006 [in Ukrainian].

Відомості про авторів

Завальнюк Євген Костянтинович – студент групи ІПІ-21м, кафедра програмного забезпечення.

Романюк Олександр Никифорович – доктор технічних наук, професор, завідувач кафедри програмного забезпечення.

Войтко Вікторія Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення.

Романюк Оксана Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення.

Снігур Анатолій Васильович – кандидат технічних наук, доцент кафедри обчислювальної техніки.

Ye. K. Zavalniuk, O. N. Romanyuk, V. V. Voitko, O. V. Romanyuk, A. V. Snigur

THE DEVELOPMENT OF THE MODIFIED SCHLICK MODEL FOR THE SPECULAR COLOR COMPONENT CALCULATION

Vinnytsia National Technical University, Vinnytsia

УДК 004.912

А. О. Концевой, О. В. Бісікало

МОДЕЛІ ГЛИБОКОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ КЛАСИФІКАЦІЇ ТЕКСТОВОЇ ІНФОРМАЦІЇ

Вінницький національний технічний університет, Вінниця

Анотація. Аналіз тексту в цілому є новою галуззю вивчення. Такі галузі, як маркетинг, управління продуктами, наукові дослідження та управління, вже використовують процес аналізу та вилучення інформації з текстових даних. У попередньому дописі ми обговорили технологію класифікації тексту, одну з найважливіших частин аналізу тексту. Класифікація тексту або категоризація тексту - це діяльність по позначенню текстів природною мовою відповідними категоріями із заздалегідь визначеного набору. Якщо говорити не просто, класифікація тексту - це процес вилучення загальних тегів із неструктурованого тексту. Ці загальні теги походять із набору заздалегідь визначених категорій. Класифікація вмісту та продуктів за категоріями допомагає користувачам легко шукати веб-сайт чи програму та переходити до них. Класифікація тексту, також відома як категоризація тексту, є класичною проблемою в обробці природної мови (NLP), метою якої є призначення міток або тегів для текстових одиниць, таких як речення, запити, абзаци та документи. Вона має широкий спектр застосувань, включаючи відповіді на запитання, виявлення спаму, аналіз настроїв, категоризацію новин, класифікацію намірів користувача, модерування вмісту тощо. Текстові дані можуть надходити з різних джерел, включаючи веб-дані, електронні листи, чати, соціальні мережі, квитки, страхові виплати, відгуки користувачів, а також запитання та відповіді від служби підтримки клієнтів. Текст є надзвичайно багатим джерелом інформації. Але витягувати корисні дані з тексту зазвичай складно та займає багато часу через неструктурований характер природно-мовної інформації. Моделі, засновані на глибокому навчанні, перевершили класичні підходи на основі машинного навчання в різних завданнях класифікації текстів, включаючи аналіз настроїв, категоризацію новин, відповіді на запитання та умовивід природної мови. У цій статті проводиться огляд найбільш поширених моделей класифікації текстів на основі глибокого навчання, розроблених за останні роки, проаналізовано їхній технічний внесок, схожість та сильні сторони.

Ключові слова: класифікація тексту, аналіз настроїв, відповіді на запитання, категоризація новин, глибоке навчання, висновок з природної мови, класифікація тем.

Abstract. Text analysis as a whole is a new field of study. Fields such as marketing, product management, research, and management already use the process of analysing and extracting information from textual data. In the previous post, we discussed text classification technology, one of the most important parts of text analysis. Text classification or text categorisation is the activity of labelling texts in natural language with appropriate categories from a predetermined set. To put it bluntly, text classification is the process of extracting generic tags from unstructured text. These generic tags come from a set of predefined categories. Categorising content and products helps users easily find and navigate to a website or app. Text classification, also known as text categorisation, is a classic problem in natural language processing (NLP) that aims to assign labels or tags to text units such as sentences, queries, paragraphs, and documents. It has a wide range of applications, including question answering, spam detection, sentiment analysis, news categorisation, user intent classification, content moderation, and more. Text data can come from a variety of sources, including web data, emails, chats, social media, tickets, insurance claims, user feedback, and customer service questions and answers. The text is an extremely rich source of information. But extracting useful data from text is usually difficult and time-consuming due to the unstructured nature of natural language information. Deep learning based models have surpassed classical machine learning based approaches in various text classification tasks, including sentiment analysis, news categorisation, question answering, and natural language inference. In this paper, we provide a comprehensive review of most widespread deep learning based models for text classification developed in recent years, and discuss their technical contributions, similarities, and strengths.

Key words: text classification, sentiment analysis, question answering, news categorisation, deep learning, natural language inference, topic classification.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-13-20>.

Вступ

Класифікація тексту, також відома як категоризація тексту, є класичною проблемою в обробці природної мови (NLP), метою якої є призначення міток або тегів для текстових одиниць, таких як речення, запити, абзаци та документи. Вона має широкий спектр застосувань, включаючи відповіді на запитання, виявлення спаму, аналіз настроїв, категоризацію новин, класифікацію намірів користувача, модерування вмісту тощо. Текстові дані можуть надходити з різних джерел, включаючи веб-дані, електронні листи, чати, соціальні мережі, квитки, страхові виплати, відгуки користувачів, а також запитання та відповіді від служби підтримки клієнтів. Текст є надзвичайно багатим джерелом інформації. Але витягувати корисні дані з тексту зазвичай складно та займає багато часу через неструктурований характер природно-мовної інформації [1].

Актуальність

Соціальні мережі – це феномен сьогодення. Переваги використання соціальних мереж полягають у вільному і швидкому зв'язку з друзями зазвичай у вигляді зручних об'єктів, таких як пости, картинки, відео і тексти. Ще одна особливість – широкі можливості у створення власних мереж: друзів, колег, членів сім'ї.

Агрегування інформації із загальнодоступних профілів дуже корисно для актуальних цілей, наприклад, таких як побудова стратегії маркетингу і виявлення груп осіб, пов'язаних із забороненими організаціями. Аналіз соціальних даних стрімко набирає популярність у всьому світі завдяки появі в 1990-х роках онлайн-сервісів соціальних мереж. З цим пов'язаний феномен соціалізації персональних даних, зокрема стали публічно доступними факти біографії, листування, щоденники, фото-, відео-, аудіоматеріали, нотатки про подорожі тощо.

Отже, соціальні мережі є унікальним джерелом даних про особисте життя та інтереси реальних людей. Це новітнє явище відкриває безпрецедентні можливості для вирішення дослідних і бізнес-задач (багато з яких до цього неможливо було вирішувати ефективно через брак даних), а також створення допоміжних сервісів і додатків для користувачів соціальних мереж. Крім того, таким станом речей обумовлюється підвищений інтерес до збору і аналізу соціальних даних з боку компаній і дослідницьких центрів.

Мета

Мета дослідження полягає в обґрунтуванні суттєвих особливостей вибору оптимальної моделі машинного навчання для розв'язання задачі класифікації текстової інформації

Задачі

1. Проаналізувати актуальні моделі машинного навчання для вирішення задач класифікації текстової інформації.
2. Оцінити їх переваги та недоліки.
3. Визначити базові метрики так критерії для вибору моделі для вирішення конкретних задач.

Розв'язання задач

Для зручності аналізу розглянемо відомі моделі, що згруповані в декілька типів на основі їх архітектури:

1. Мережі прямої подачі – розглядають текст як мішок слів.
2. Моделі на основі RNN (повторювані нейронні мережі) – розглядають текст як послідовність слів і призначені для захоплення залежностей у текстових структурах.
3. Моделі на основі CNN (згорткові нейронні мережі) – навчаються розпізнавати шаблони в тексті, такі як ключові фрази тощо.
4. Капсульні мережі вирішують проблему втрати інформації, від якої виникають операції об'єднання згорткових нейронних мереж, і нещодавно застосовувалися для текстової класифікації.
5. Мережі з розширеною пам'яттю поєднують нейронні мережі з формою зовнішньої пам'яті, яку моделі можуть читати і записувати.
6. Графові нейронні мережі призначені для захоплення внутрішніх графічних структур природної мови, наприклад дерева синтаксичного та семантичного розбору.
7. Сіамські нейронні мережі призначені для відповідності тексту, окремий випадок текстової класифікації.
8. Гібридні моделі поєднують повторювані нейронні мережі і згорткові нейронні мережі, щоб охопити локальні та глобальні особливості речень та документів [2].

Мережі прямої подачі є одними з найпростіших моделей глибокого навчання для представлення ті обробки тексту, тим не менш, вони досягли високої точності на багатьох тестах. Ці моделі розглядають текст як мішок слів. Для кожного слова вони вивчають векторне представлення, використовуючи модель вбудовування, таку як word2vec або Glove, беруть векторну суму або середнє значення вкладень як представлення тексту, передають його через одне або кілька шарів відомих як багатозарові перцептрони (MLP), а потім виконують класифікацію подання кінцевого шару за допомогою класифікатора, такого як логістична регресія, найвний байес або SVM. Прикладом цих моделей є глибока усереднена мережа (DAN), архітектура якої показана на рисунку 1.

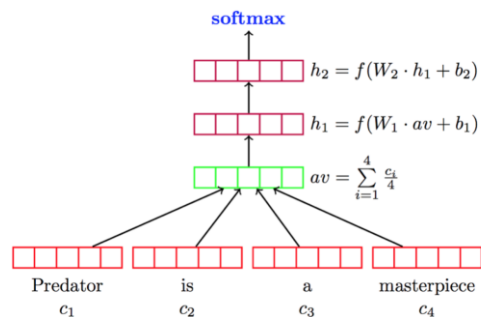


Рисунок 1 – Архітектура глибокої усередненої мережі (DAN)

Незважаючи на свою простоту, глибока усереднена мережа перевершує інші більш складні моделі, які призначені для явного вивчення композиційності текстів. Наприклад, глибока усереднена мережа перевершує синтаксичні моделі на наборах даних з високою синтаксичною дисперсією. Doc2vec викори-

стовує неконтрольований алгоритм для вивчення представлень ознак фіксованої довжини фрагментів текстів змінної довжини, таких як речення, абзаци та документи.

Як показано на рисунку 2, архітектура doc2vec подібна до моделі безперервний мішок слів (CBOW). Єдина відмінність полягає в додатковому маркері абзацу, який зіставляється з вектором абзацу за допомогою матриці D .

В doc2vec конкатенація або середнє значення цього вектора з контекстом із трьох слів використовується для передбачення четвертого слова. Вектор абзацу представляє відсутню інформацію з поточного контексту і може виконувати функцію пам'яті теми абзацу. Після навчання вектор абзацу використовується як ознаки абзацу (наприклад, замість або на додаток до мішку слів) і подається до класифікатора для передбачення. Після публікації Doc2vec досягає нових результатів у виконанні кількох завдань текстової класифікації [3].

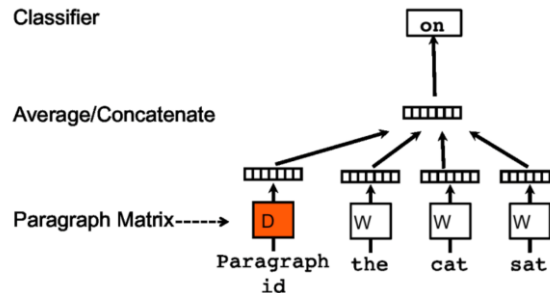


Рисунок 2 – Модель doc2vec

Моделі на основі повторюваних нейронних мереж розглядають текст як послідовність слів і призначені для захоплення залежностей слів і текстових структур для класифікації тексту. Однак звичайні моделі повторюваних нейронних мереж не працюють добре і часто мають низьку продуктивність нейронних мереж із прямим зв'язком. Серед багатьох варіантів повторюваних нейронних мереж найпопулярнішою архітектурою є довготривала пам'ять (LSTM), яка розроблена для кращого захоплення довгострокових залежностей. Довготривала пам'ять вирішує проблеми зникнення або вибуху градієнта, від яких страждають ванільні повторювані нейронні мережі, вводячи комірку пам'яті для запам'ятовування значень через довільні проміжки часу, а також три вентиля (вхідний шлюз, вихідний шлюз, вентиль забуття) для регулювання потоку інформації всередину та з неї. Велись роботи з удосконалення моделей повторюваних нейронних мереж і нейронних мереж з довготривалою пам'яттю для текстової класифікації шляхом захоплення більшої інформації, такої як деревоподібні структури природної мови, відношення між словами в тексті, теми документів тощо.

Варто зазначити, що повторювані нейронні мережі належать до широкої категорії - рекурсивні нейронні мережі. Рекурсивна нейронна мережа застосовує той самий набір ваг рекурсивно до вхідних даних структури для створення структурованого передбачення або векторного представлення на вхідних даних змінного розміру. У той час як повторювані нейронні мережі є рекурсивними нейронними мережами з лінійною ланцюговою структурою вхідних даних, існують рекурсивні нейронні мережі, які працюють на ієрархічних структурах, таких як дерева розбору речень природної мови, об'єднуючи дочірні уявлення в батьківські уявлення. Повторювані нейронні мережі є найпопулярнішими рекурсивними нейронними мережами для текстової класифікації, оскільки вони ефективні та прості у використанні – вони розглядають текст як послідовність маркерів, не вимагаючи додаткових міток структури, таких як дерева аналізу [4].

Моделі на основі повторюваних нейронних мереж навчаються розпізнавати шаблони в часі, тоді як моделі на основі згорткових нейронних мереж вчать розпізнавати шаблони в просторі. Моделі повторюваних нейронних мереж добре працюють для завдань природомовної обробки тексту, таких як тегування частин мови або QA, де потрібне розуміння дальньої семантики, тоді як моделі на базі згорткових нейронних мереж добре працюють там, де важливо виявляти локальні та інваріантні позиції шаблони. Ці шаблони можуть бути ключовими фразами, які виражають певний настрій, як-от «мені подобається» або тему, як-от «види, що знаходяться під загрозою зникнення». Таким чином, згорткові нейронні мережі моделі стали однією з найпопулярніших архітектур моделей для класифікації текстової інформації.

Одна з перших моделей класифікації текстової інформації на основі згорткових нейронних мереж використовує динамічний k -max-пулінг і називається динамічна модель на основі згорткової нейронної мережі (DCNN).

Як показано на рисунку 3, перший шар динамічної моделі на основі згорткової нейронної мережі створює матрицю речень, використовуючи вбудовування для кожного слова в реченні. Потім згорткова

архітектура, яка чергує широкі згорткові шари з шарами динамічного об'єднання, заданими динамічним k -тах-пулінгом, використовується для створення карти ознак над реченням, яка здатна чітко фіксувати короткі та дальні відношення слів і фраз. Параметр об'єднання k можна динамічно вибирати залежно від розміру пропозиції та рівня в ієрархії згортки [5].

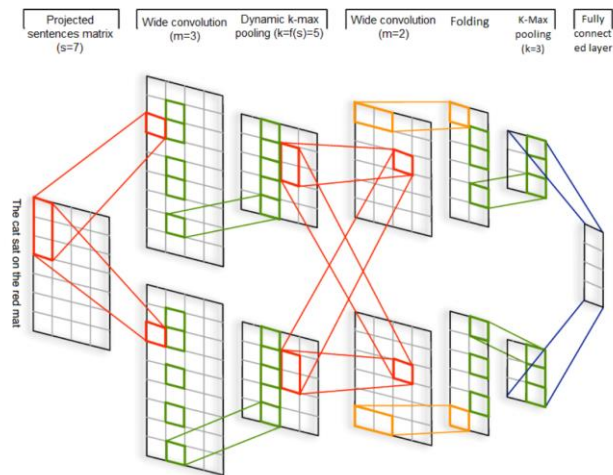


Рисунок 3 – Архітектура динамічної моделі на основі згорткової нейронної мережі

Моделі на основі згорткових нейронних мереж класифікують зображення або тексти, використовуючи послідовні шари згортки і об'єднання. Незважаючи на те, що операції об'єднання визначають основні особливості та зменшують обчислювальну складність операцій згортки, вони втрачають інформацію щодо просторових відносин і, ймовірно, неправильно класифікують об'єкти на основі їх орієнтації або пропорції.

Нещодавно капсульні мережі були застосовані до розв'язання задач класифікації текстів, де капсули адаптовані для представлення речення або документа у вигляді вектора. Модель складається з чотирьох шарів: (1) n -грамовий згортковий шар, (2) шар капсули, (3) згорткий капсульний шар і (4) повністю з'єднаний шар капсули. Автори експериментують із трьома стратегіями для стабілізації процесу динамічної маршрутизації, щоб пом'якшити перешкоди шумових капсул, які містять довідкову інформацію, таку як стоп-слова або слова, які не мають відношення до жодної категорії документів. Вони також досліджують дві архітектури капсул, Capsule-A і Capsule-B, які зображені на рисунку 4. Capsule-A схожа на CapsNet. Capsule-B використовує три паралельні мережі з фільтрами з різними розмірами вікон на згортковому шарі n -грам, щоб дізнатися більш повне представлення тексту [6].

Хоча тексти природною мовою мають послідовний порядок, вони також містять внутрішні структури графів, такі як дерева синтаксичного та семантичного аналізу, які визначають синтаксичні та семантичні відносини між словами в реченнях. Однією з найбільш ранніх моделей побудованих на основі графів що розроблені для природомової обробки тексту, є TextRank. Автори пропонують представити текст природною мовою у вигляді графіка $G(V, E)$, де V позначає набір вузлів, а E — набір ребер серед вузлів. Залежно від наявних додатків вузли можуть представляти текстові одиниці різних типів, наприклад, слова, словосполучення, цілі речення тощо. Аналогічно, ребра можна використовувати для представлення різних типів відносин між будь-якими вузлами, наприклад, лексичні чи семантичні відносини, контекстне накладання тощо.

Сучасні графічні нейронні мережі (GNN) розробляються шляхом розширення підходів глибокого навчання для даних графів, таких як текстові графіки, які використовує TextRank. Глибокі нейронні мережі, такі як згорткові нейронні мережі, повторювані нейронні мережі та автокодери були узагальнені протягом останніх кількох років для обробки складних даних графів. Наприклад, двовимірна згортка згорткової нейронної мережі для обробки зображень узагальнюється для виконання згортки графіка, беручи середнє зважене значення інформації про околиці вузла. Серед різних типів нейронних мереж згортки, такі як графові згорткові мережі (GCNs) та їх варіанти, є найпопулярнішими, оскільки вони ефективні та зручні для компонування з іншими нейронними мережами, а також досягли найсучасніших результатів. У багатьох програмах графові згорткові мережі є ефективним варіантом згорткових нейронних мереж на графах.

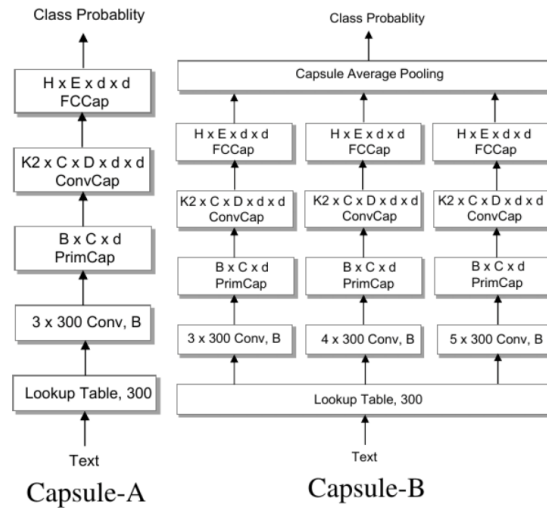


Рисунок 4 – Схема капсул А і В для класифікації текстів

Типовим застосуванням графових згорткових мереж в природомовній обробці тексту є задача класифікації текстової інформації. Графові згорткові мережі використовують взаємозв'язки документів або слів для визначення міток документів.

На рисунку 5 зображена модель, заснована на ієрархічній таксономії та CNN-капсулі з графом уваги. Однією з унікальних особливостей моделі є використання ієрархічних відносин між мітками класів, що у попередніх методах вважаються незалежними.

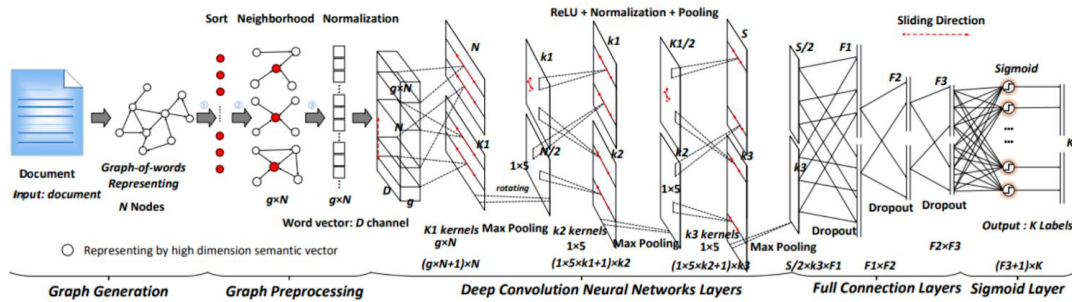


Рисунок 5 – Архітектура моделі на основі ієрархічної таксономії та CNN-капсулі з графом уваги

Розглянемо інший підхід до побудови такої нейронної мережі. Для мережі будується єдиний текстовий граф для корпусу на основі спільного зустрічання слів і взаємовідношень слів у документі, після чого тренується текстова графова нейронна мережа для корпусу, як показано на рисунку 6.

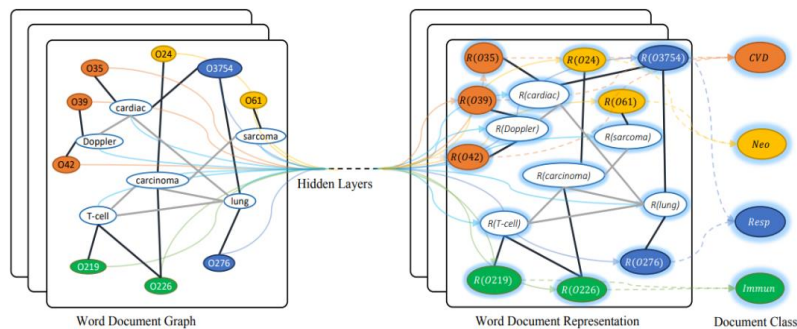


Рисунок 6 – Текстова графова нейронна мережа

Текстова графова нейронна мережа ініціалізується за допомогою однієї гарячої заміни для слова і документа, а потім спільно вивчає вбудовування як для слів, так і для документів, під контролем відомих міток класів для документів. Побудова графової нейронної мережі (GNN) для великомасштабного текстового корпусу коштує дорого. Проводяться дослідження зі зниження вартості моделювання шляхом зменшення складності моделі або зміни стратегії навчання моделі. Прикладом першого є модель проста згортка графів (SGC), де глибока згортка графової нейронної мережі спрощується шляхом багаторазового видалення нелінійності між послідовними шарами та згортання отриманих функцій (матриць wag) в одну лінійну трансформацію. Прикладом останнього є графова нейронна мережа на текстовому рівні. Замість того, щоб будувати графік для всього текстового корпусу, графова нейронна мережа на рівні тексту створює один графік для кожного фрагмента тексту, визначеного ковзаючим вікном на корпусі тексту, щоб зменшити споживання пам'яті під час навчання. Деякі з інших багатообіцяючих робіт на основі графових нейронних мереж включають GraphSag та контекстуалізовані нелокальні нейронні мережі [7].

Одним із вузьких місць обчислень, яким страждають повторювані нейронні мережі, є послідовна обробка тексту. Незважаючи на те, що згорткові нейронні мережі є менш послідовними, ніж повторювані нейронні мережі, обчислювальні витрати на захоплення зв'язків між словами в реченні також ростуть зі збільшенням довжини речення, подібно до повторюваних нейронних мереж. Трансформаторні моделі долають це обмеження, застосовуючи самоуважність, щоб паралельно обчислювати для кожного слова в реченні або документувати «оцінку уваги», щоб змоделювати вплив кожного слова на інше. Завдяки цій функції, трансформаторні моделі дозволяють набагато більше розпаралелювання, ніж згорткові нейронні мережі і повторювані нейронні мережі, що дає можливість ефективно навчати дуже великі моделі на великих обсягах даних на графічних процесорах.

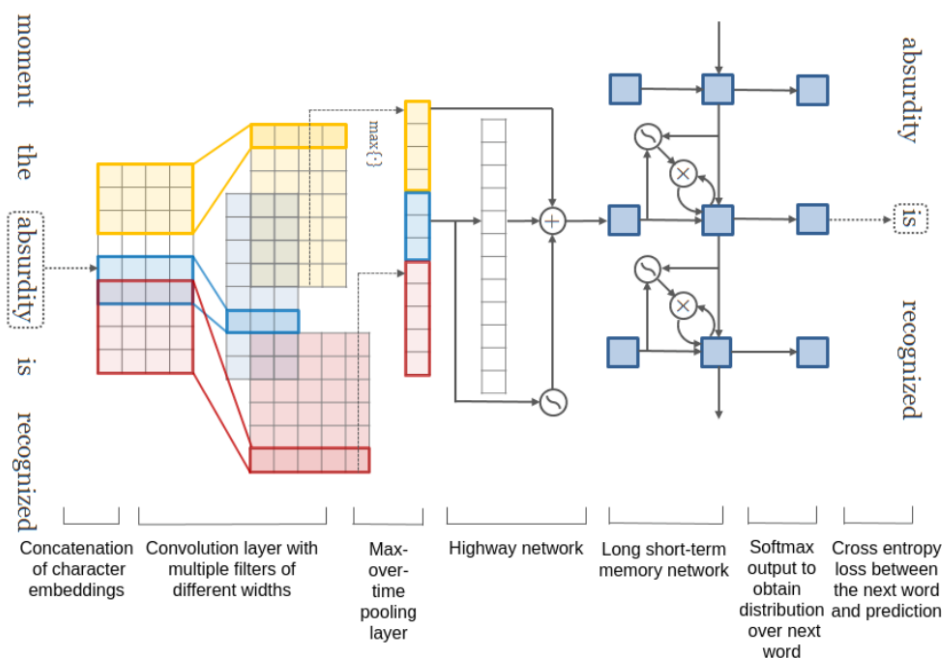


Рисунок 7 – Архітектура нейронної мережі з згортковими нейронними мережами та довготривалою пам'яттю

З 2018 року спостерігається зростання набору великомасштабних попередньо підготовлених мовних моделей (PLM) на основі трансформаторів. У порівнянні з попередніми контекстуалізованими моделями вбудовування, заснованими на згорткових нейронних мережах, підготовлені мовні моделі на базі трансформацій використовують набагато глибші архітектури мережі (наприклад, 48-шарові трансформатори) і попередньо навчаються на значно більшій кількості текстових корпусів, щоб вивчити контекстні текстові репрезентації шляхом передбачення слів, обумовлених їх контекстом. Ці підготовлені мовні моделі точно налаштовані за допомогою міток для конкретних завдань і створили новий рівень техніки в багатьох наступних завданнях природомовної обробки текстів, включаючи класифікацію текстової інформації.

Підготовлені мовні моделі можна згрупувати в дві категорії: підготовлені мовні моделі з авторегресією та автокодуванням. Однією з найбільш ранніх підготовлених мовних моделей з авторегресією є OpenGPT, односпрямована модель, яка прогнозує текстову послідовність слово за словом зліва направо (або справа наліво), причому передбачення кожного слова залежить від попередніх передбачень [8].

Для того, щоб обрати модель машинного навчання для вирішення задачі класифікації текстів підхід до вибору сильно змінюється залежно від характеру цільового завдання та домену, доступності міток у домені, затримки та обмежень пропускну здатності програм тощо. Хоча немає жодних сумнівів, що розробка текстового класифікатора є процесом проб і помилок, аналізуючи останні результати загальнодоступних тестів, було вирішено керуватись наступними кроками:

Вибір підготовленої мовної моделі. Використання підготовлених мовних моделей призводить до значних покращень у всіх популярних завданнях класифікації тексту, а підготовлені мовні моделі з автотодування часто працюють краще, ніж підготовлені мовні моделі з авторегресією (наприклад, OpenAI GPT).

Адаптація домену. Більшість підготовлених мовних моделей навчаються на текстових корпусах загального домену (наприклад, Web). Якщо цільовий домен різко відрізняється від загального домену, ми можемо розглянути можливість адаптації підготовлених мовних моделей з використанням внутрішньо доменних даних шляхом постійного попереднього навчання вибраної підготовленої мовної моделі загального домену. Для доменів із великою кількістю немаркованого тексту, таких як біомедицина, моделі мови попереднього навчання з нуля також можуть бути хорошим вибором.

Конструкція моделі для конкретного завдання. З урахуванням введеного тексту підготовлена мовна модель створює послідовність векторів у контекстному представленні. Потім один або кілька шарів, що стосуються завдання, додаються зверху, щоб створити кінцевий результат для цільового завдання. Вибір архітектури специфічних для завдання шарів залежить від характеру завдання, наприклад, коли необхідно охопити мовну структуру тексту. Нейронні мережі з прямим зв'язком розглядають текст як мішок слів, повторювані мовні моделі можуть фіксувати порядок слів, згорткові мовні моделі добре розпізнають шаблони, такі як ключові фрази, механізми уваги ефективні для визначення корельованих слів у тексті і графові нейронні мережі можуть бути хорошим вибором, якщо графічні структури природної мови (наприклад, розбір дерев) корисні для цільового завдання.

Точне налаштування для конкретного завдання. Залежно від наявності міток у домені, рівні, що стосуються завдання, можна тренувати окремо з фіксованою підготовленою мовною моделлю або разом із підготовленими мовними моделями. Якщо потрібно створити декілька подібних текстових класифікаторів (наприклад, класифікатори новин для різних доменів), точне налаштування для кількох завдань є хорошим вибором для використання мічених даних подібних доменів.

Стиснення моделі. Підготовлені мовні моделі дорогі в обслуговуванні. Їх часто потрібно стиснути, наприклад, шляхом перегонки знань, щоб задовольнити обмеження затримки та ємності в реальних програмах.

Точність і частота помилок. Це основні показники для оцінки якості моделі класифікації. Нехай TP, FP, TN, FN позначають істинно позитивний, хибнопозитивний, істинно негативний і хибнонегативний відповідно. Точність класифікації та рівень помилок визначені в формулі:

$$\text{Точність} = \frac{TP+TN}{N} \quad \text{Частота помилок} = \frac{FP+FN}{N}$$

де N – загальна кількість зразків. Очевидно, що Частота помилок = $1 - \text{Точність}$.

Точність / Відкликання / Оцінка F1. Це також основні показники, і вони використовуються частіше, ніж точність або частота помилок для незбалансованих наборів тестів, наприклад, більшість тестових зразків мають одну мітку класу. Оцінка F1 – це середнє гармонійне значення точності та запам'ятовування і досягає свого найкращого значення при 1 (ідеальна точність і запам'ятовування) і найгіршого при 0.

$$\text{Точність} = \frac{TP}{TP+FP}, \quad \text{Відкликання} = \frac{TP}{TP+FN}, \quad \text{Оцінка F1} = \frac{2 \cdot \text{Точність} \cdot \text{Відкликання}}{\text{Точність} + \text{Відкликання}}$$

Для задач класифікації з кількома класами ми завжди можемо обчислити точність і відкликання для кожної мітки класу і проаналізувати індивідуальну продуктивність міток класу або усереднювати значення, щоб отримати загальну точність і запам'ятовування.

Точна відповідність (EM). Показник точної відповідності є популярним показником для систем відповіді на запитання, який вимірює відсоток прогнозів, які точно відповідають будь-якій із основних істинних відповідей. EM є однією з основних метрик, що використовуються для SQuAD.

Середній взаємний ранг (MRR). MRR часто використовується для оцінки ефективності алгоритмів ранжирування в задачах НЛП, таких як ранжування запитів-документів і QA. MRR визначається в рівнянні:

$$MRR = \frac{1}{[Q]} \sum_{i=1}^Q \frac{1}{rank_i}$$

де Q – набір усіх можливих відповідей, а $rank_i$ – рейтингова позиція відповіді, що відповідає дійсності.

Висновки

За останні кілька років класифікація текстової інформації досягла значного прогресу за допомогою моделей глибокого навчання. Було запропоновано кілька нових ідей (таких як нейронне вбудовування, механізм уваги, самоувага, Transformer, BERT і XLNet), які призвели до швидкого прогресу за останнє десятиліття. Незважаючи на наявний прогрес, є ще проблеми, які потрібно вирішити.

Хоча останнім часом було зібрано ряд великомасштабних наборів даних для загальних завдань текстової класифікації, залишається потреба в нових наборах даних для більш складних завдань, таких як класифікація текстів для багатомовних документів і для надзвичайно довгих документів.

Включення загальних знань в моделі глибокого навчання може значно покращити продуктивність моделі, майже так само, як люди використовують загальні знання для виконання різних завдань.

Більшість сучасних нейронних мовних моделей вимагають значного обсягу пам'яті для навчання та висновків. Ці моделі повинні бути стиснуті, щоб відповідати обмеженням обчислень і зберігання граничних додатків. Це можна зробити шляхом побудови моделей за допомогою дистиляції знань, або за допомогою методів стиснення моделей.

Отже, за результатами аналізу найбільш популярних моделей глибокого навчання, які були розроблені за останні шість років, показано, що цей напрям досліджень дозволив суттєво покращити сучасний технологічний рівень та ефективність розв'язання задач класифікації текстової інформації.

References

- [1] Bisikalo O. System for definition of indicator characteristics of social networks participants Profiles / Oleg Bisikalo, Anton Kontsevoi // Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2020). – CEUR Workshop Proceedings Volume 2604, 2020. – Lviv, Ukraine, April 23-24, 2020. – Pp. 77-88. – ISSN: 16130073.
- [2] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. MIT press, 2016.
- [3] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2. Association for Computational Linguistics, 2012.
- [4] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in Proceedings of the 2013 conference on empirical methods in natural language processing, 2013.
- [5] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in Advances in neural information processing systems, 2015.
- [6] W. Zhao, H. Peng, S. Eger, E. Cambria, and M. Yang, "Towards scalable and reliable capsule networks for challenging NLP applications," in ACL, 2019.
- [7] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in neural information processing systems, 2017.
- [8] Y. Sun, S. Wang, Y.-K. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "Ernie 2.0: A continual pre-training framework for language understanding." in AACL, 2020.

Стаття надійшла: 14.06.2022.

Відомості про авторів

Концевой Антон Александрович – аспірант факультету інтелектуальних інформаційних технологій та автоматизації.

Бісікало Олег Володимирович – доктор технічних наук, професор, факультет інтелектуальних інформаційних технологій та автоматизації.

A. O. Kontsevoi, O. V. Bisikalo

ANALYSIS OF DEEP LEARNING MODELS FOR TEXT INFORMATION CLASSIFICATION TASKS

Vinnitsia National Technical University, Vinnitsia

УДК 004.056

В. І. Маліновський, Л. М. Куперштейн

АНАЛІЗ ЗАГРОЗ БЕЗПЕКИ МІКРОКОНТРОЛЕРІВ

Вінницький національний технічний університет, Вінниця

Анотація. В статті розглянуто і приведено матеріали окремих досліджень щодо основних проблем безпеки у мікроконтролерах, які функціонують в складі систем управління як загальних, так і спеціалізованих пристроїв. Зокрема в матеріалах досліджень проаналізовані основні тенденції розвитку загроз безпеки і наведено основні вектори атак. Проаналізовано закордонний та вітчизняний досвід прояву несанкціонованих впливів та кіберзагроз в основні найбільш критичні місця архітектури мікроконтролерів. Аналіз показав, що базовими точками прояву загроз безпеки втручань в роботу мікроконтролера є: регістри, стек, АЛП, пам'ять (EEPROM та Flash), порти введення-виведення, схеми та інтерфейси передачі даних в МК, схеми додаткового функціоналу підключення до зовнішніх периферійних пристроїв і канали підключення осцилятора тактової частоти. Крім того, проведено дослідження основних загроз пам'яті мікроконтролерної системи, а саме: втручання із прямим доступом до пам'яті, доступ до регістрів керування і доступ до буфера мікроконтролера, переповнення стеку буфера, віддалений запуск коду, зовнішній доступ та атаки по вторинним каналам (в т.ч. по зовнішнім лініям передачі даних у МК), зміна порядку адресації в МК, зміна/підміна значень адрес і нумерації стеку, втручання в роботу регістрів даних та індикації стану портів введення/виведення мікроконтролера. Також досліджено ряд механізмів захисту мікроконтролерів, які у комплексі дозволяють знизити ризики несанкціонованих впливів на мікроконтролерну систему. До них відносяться такі: циклічний контроль надмірності коду, моніторинг живлення і моніторинг ресурсів, використання ізоляваності і контролю функціональності системи тактування, контроль цілісності та достовірності вмісту пам'яті, контроль зовнішніх фізичних і електричних параметрів МК, віртуалізація основного обчислювального процесу та його багаторівневе резервування копіюванням і відновленням попередніх станів, Використання криптографічних систем і алгоритмів обробки даних, використання багаторівневої програмно-апаратної ізоляції.

Ключові слова: кіберзахист, кіберзагроза, інформаційна безпека, вразливість, мікроконтролер.

Abstract. In the article the materials of individual studies of the main security problems in microcontrollers operating as part of control systems of both general and specialized devices were considered and analyzed. In particular, the main trends of security threats were analyzed and the main attacks vectors were presented. Foreign and domestic experience of manifestation of unauthorized influences and cyber threats in the main most critical places of microcontroller architecture is analyzed. The analysis showed that the basic places of security threats and interventions to the microcontroller are: registers, stack, LPA, memory (EEPROM and Flash), input-output ports, circuits and interfaces for data transfer to the MC, circuits for additional functionality for connecting to external peripheral devices and channels for connecting the clock oscillator. In addition, explorations were carried out on the main threats of the microcontroller system memory, namely: interference with direct memory access, access to control registers and access to the microcontroller buffer, buffer stack overflow, remote code execution, external access and attacks on secondary channels (including via external data lines in the MC), changing the order of addressing in the MC, changing / replacing the values of addresses and stack numbering, interfering with the work of data registers and indicating the state of the input / output ports of the microcontroller. A number of protection mechanisms for microcontrollers have also been studied, which together can reduce the risks of unauthorized actions on the microcontroller system. These include: cyclic code redundancy control, power monitoring and resource monitoring, the using of isolation and control of the functionality of the clock system, control of the integrity and reliability of the memory contents, control of external physical and electrical parameters of the microcontroller, virtualization of the main computing process and its multi-level redundancy and restoration of previous states, the using of cryptographic systems and data processing algorithms, the using of multi-level software and hardware isolation.

Key words: cyber protection, cyber threat, information security, vulnerability, microcontroller.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-21-32>.

Вступ

Сучасні технології інформаційних систем на базі мікроконтролерів та мереж досить активно набувають динамічного стрімкого розвитку в останні роки і впроваджуються у все ширші сфери людської діяльності. Останні роки розвитку цифрових та інформаційних технологій набув особливо великих темпів: від впровадження в традиційні сфери автоматизації і цифрових технологій систем загального призначення до розвитку прогресивних спеціалізованих систем із мікроконтролерними архітектурами та штучним інтелектом, систем робототехніки і автоматики у вигляді конкретних науково-промислових вітчизняних та закордонних розробок.

Разом із тим набуває розвитку і швидких темпів ескалація сучасних кібератак у різних галузях [1-4], зокрема і на галузь цифрових інформаційних технологій мікроконтролерних систем. Кількість кібератак пропорційно збільшується із ростом цифрових технологій, на що вказує закордонний та вітчизняний досвід [3-20]. Для нейтралізації і вчасного попередження, проведення аналізу процесів функціонування мікроконтролерів (МК) і нейтралізації цих загроз використовується різні методи [3, 4, 5, 8, 11], зокрема й в більшості комплексні системи інформаційного захисту – ізоляція області роботи МК, підходи моніторингу, криптостійкі надійні алгоритми, антивірусні платформи і мережеві системи аналізу трафіку [4, 5, 21-23]. Ефективність цих методів, підходів і систем та досить часто й затрати на їх реалізацію не у повній мірі дозволяють отримати необхідний рівень безпеки враховуючи сучасні загрози «0»-го дня і рівень сучасного шпигунського і хакерського програмного забезпечення для МК і систем управління на їх основі [8, 10-16].

Метою статті є проведення аналізу сучасних та найбільш актуальних загроз безпеці мікроконтролерів в результаті їх функціонування в складі як спеціалізованих так і загальних систем управління для більш повного та цілісного розуміння можливих недоліків у їх захисті та подальшої оцінки ризиків їх використання.

Проблема інформаційної безпеки сучасних МК систем і потенційні шляхи її вирішення

Досить часто виникають задачі локального захисту даних і забезпечення безпеки мікроконтролерів на локальному рівні, наприклад в кінцевих пристроях чи сенсорах Інтернету речей (IoT) [6]. Такі кінцеві пристрої або датчики (сенсори) являють собою кінцеві інформаційні вузли із МК і з досить складною архітектурою та програмно-апаратною основою (досить часто являють собою окремих WEB-сервер) із підключенням по каналам із різними мережевими протоколами до центральних системи управління. Для задач захисту таких застосунків в локальних місцях неможливо та/або не ефективно використовувати складну і досить вартісну окрему інфраструктуру із комплексним антивірусним захистом або комплексну систему захисту інформації (програмно-апаратну платформу, або суто програмну антивірусну платформу), так як локальне розміщення і технічний рівень кінцевої локальної точки інформаційного пристрою не є повноцінною обчислювальною системою [7].

Середовище кіберзагроз МК продовжує зростати, оскільки додається все більше варіацій застосованих взаємопов'язаних МК пристроїв. Екосистема мікропроцесорних пристроїв, мікроконтролерів і в т.ч. пристроїв Інтернету речей на базі МК в сучасному світі збільшується із збільшенням кількості елементів, взаємозв'язків та обсягів даних. Те, як сучасні компанії, які експлуатують МК пристрої, справляються із більшістю відомих інформаційних ризиків і загроз, часто залежить від стратегічного підходу до створення попереджувального плану інформаційних втручань і кібератак на пристрої із МК, замість того, щоб розраховувати на відповідну швидку реакцію до та після інциденту, в мінімально можливий час. Уникнути «сліпих зон» безпеки при збереженні пильності в інформаційній системі — завдання, яке здатні оцінити більшість організацій. Для прикладу сучасні засоби інформаційних втручань в промислові МК пристрої дозволяють успішно реалізовувати шкідливий функціонал і втручання в процеси МК шкідливим ПЗ [5], наприклад: Stuxnet, Flame, miniFlame, Duqu, Gauss, Reign, Wiper, Shamoon, яке експлуатує вразливості програмного коду МК систем і систем індустріального контролю в складі АСУ ТП на базі мікроконтролерів. Це дозволяє впроваджувати шкідливий код і здійснювати інформаційні втручання із порушенням штатного функціоналу МК засобів і їх функціонування в цілому. За даним компанії Cisco Systems Inc. [8], яка є лідером галузі телекомунікацій і займає передові позиції в галузі кібербезпеки інформаційних систем – розвиток засобів для здійснення інформаційних втручань і атак, а також їх функціоналу і широти сфери застосувань значно перевищує рівні розробки сучасних засобів виявлення, попередження і захисту (IPS/IDS/SecD). І особливо це стосується загрозу у формі шкідливого ПЗ і спеціалізованого вузько орієнтованого шкідливого ПЗ для МК систем, яке експлуатує вразливості Meltdown та Spectre [9-13].

Важливою є проблема, що полягає у збільшенні ескалації загроз та збільшенні збоїв у появі незначених помилок при виконанні операцій ділення чисел з плаваючою комою, причиною яких була відсутність декількох входжень в таблицю пошуку, використовувану для прискорення обчислень [3, 4]. Цей недолік, хоч і проявлявся рідко і не на всіх вхідних даних в МК, отримав широкий розголос і привів до відкликання окремих лінійок мікроконтролерів і обладнання, в якому вони використовувались. Це звісно призвело до великих втрат для компаній і корпорації виробників.

Також, експертами в галузі кібербезпеки виявлено атаку, що дозволяє непомітно модифікувати пам'ять в середовищах віртуалізації ресурсів мікропроцесорів і мікроконтролерів високої продуктивності [10]. Сучасні кіберзагрози і інформаційні ризики від діяльності кваліфікованих хакерів із спеціалізованим ПЗ досить значні. Останні дослідження показують можливість зламу ключів захищених віртуальних середовищ і середовищ віртуалізації МК із непомітним встановленням шкідливого ПЗ (шкідливі програмні модулі) [8-13,15-18]. Так, група дослідників з Амстердамського технічного університету виявили нову атаку, що дозволяє модифікувати пам'ять віртуальної машини [15, 16]. Вчені представили докладний опис атаки, що отримала назву Flip Feng Shui (дослівно можна перекласти як «перевернутий фен-шуй»), згідного якого хакери можуть зламати ключі захищених віртуальних машин і області захищеної пам'яті в МК системах або непомітно встановити шкідливе ПЗ на МК системи. Суть атаки полягає у тому, що спочатку зловмисники отримують доступ до пам'яті і до її віртуального середовища в МК. Потім записується копія сторінки пам'яті, яка за службовими даними вже існує у вразливій області пам'яті. З метою економії місця ідентичні сторінки об'єднуються і записуються в одну і будуть збережені в одному і тому ж місці пам'яті на фізичній машині, забезпечуючи хакерам можливість вносити зміни в основну пам'ять обчислювальної системи. Атака можлива завдяки вразливості Rowhammer [16 - 18], що дозволяє здійснювати маніпуляції з бітами (bit-flipping) і змінювати вразливі комірки пам'яті в МК системі. Даний тип атак стосується більше високорозрядних МК систем (із 32- та 64-розрядністю). Атаки були продемонстровані на операційних системах Debian і Ubuntu для процесорів і зовнішньої пам'яті у високорозрядних МК системах. В ході першої з них їм вдалося підключитися до віртуальної машини через скомпрометовану SSH-сесію (дослідники змінили один біт відкритого RSA-ключа жертви [15-16]). У другому випадку за допомогою модифікованого URL і програми apt-get дослідники встановили шкідливий програмний пакет.

Для вирішення таких проблем безпеки потрібна розробка окремих алгоритмічних і організаційних підходів або невеликих програмно-апаратних систем безпеки для МК (пристроїв або засобів мікропроцесорного захисту), які дозволять захистити інформаційні процеси в МК, в кінцевих точках введення/виведення даних в МК, які можуть забезпечити ефективний, недорогий і дієвий локальний захист від більшої частини кіберзагроз.

Розробка і впровадження таких локальних підходів і способів або кінцевих пристроїв «обв'язки» (зовнішньої периферії МК) є актуальною і перспективною задачею для забезпечення локального захисту даних кінцевих інформаційних точок і алгоритмічних процесів обробки даних в МК, що в кінцевому варіанті дозволить підвищити надійність і стабільність їх функціонування.

За функціональним призначенням такі пристрої та/або заходи повинні реалізовувати наступні функції:

- швидкий аналіз і виявлення інформаційних загроз в локальному середовищі оточення МК та/або в зовнішньому пристрої/сенсорі, до якого він підключається ;
- поведінковий аналіз трафіку і інформаційних процесів і функціонального стану локального зовнішнього пристрою/датчика кінцевої інформаційної точки;
- виявлення «інформаційних аномалій» - нехарактерних станів і аномальної поведінки роботи мікропроцесорного пристрою і зовнішнього середовища навколо нього;
- виявлення втручань та неідентифікованих підключень та/або вторгнень в систему і канали зв'язку;
- наявність надійних і стійких функціональних алгоритмів мікропрограм функцій і програмного коду функцій і процедур, у тому числі в аспекті криптографії;
- оперативне інформування центрального пристрою/центральної системи безпеки про внутрішню загрозу або потенційну загрозу або спроби/намагання вторгнення до системи;
- «прозорість» для інформаційного трафіку і непомітність за фізичними і інформаційними характеристиками;
- швидка і ефективна обробка в режимі реального часу, або режимі, наближеному до реального часу, для оперативного опрацювання співмірному за швидкістю основного трафіку від кінцевої інформаційної точки;
- оперативне знешкодження або мінімізація ризиків від кіберзагроз та забезпечення захисту локальної кінцевої точки від вторгнень або помилок у роботі.

Необхідність швидкої обробки даних від локальної кінцевої точки і моніторингу процесів із попередньою адаптацією до актуального стану в часі є запорукою впровадження системи на базі мікропроцесорних засобів захисту із власною функціональною малою (орієнтованою на МК) програмно-апаратною архітектурою захисту, що орієнтована на конкретні локальні задачі і процеси. Розробка і впровадження інноваційних перспективних засобів мікропроцесорного захисту для мінімізації та ліквідації інформаційних загроз в кінцевих локальних точках і пристроях, які б могли працювати автономно і відокремлено від основної системи, де розгорнута комплексна система захисту дозволило б побудувати ефективну і більш захищену інформаційну інфраструктуру сучасних систем обробки і передачі даних. Такі підходи і методи автономного локального захисту відрізняються відносно невеликою вартістю, орієнтовані під конкретні локальні задачі дозволяють підвищувати захист даних у кінцевих точках і можуть стати частиною комплексної стратегії захисту даних на інформаційних об'єктах.

Сучасні загрози безпеки у мікроконтролерах

Основними загрозами в сучасних архітектурах мікроконтролерів є [5, 7, 9-22]:

- прямий і опосередкований доступ до пам'яті, доступ до регістрів, буфера ОЗП, тощо;
- переповнення /буфера, зчитування буфера при несанкціонованому доступі до нього;
- віддалене виконання коду, та/або зовнішній доступ до ліній передачі даних у МК, зчитування із зовнішніх ліній передачі даних в МК ;
- зміна порядку адресації в МК, зміна/підміна значень адрес;
- окремі вразливості ядра та інших компонент, вразливості архітектури, вразливості і вплив на процеси роботи арифметико-логічного пристрою (АЛП) мікроконтролера (в т.ч. і мікропроцесора);
- доступ до ресурсів МК та до окремих регістрів (в т.ч. конфігураційних із зовні), пряму втручання/пересилка команд керування і передачі даних;
- переповнення стеку адрес, переповнення пам'яті, пряма зчитування значень стека, злам та несанкціоноване втручання в ядро системи;
- несанкціоноване втручання і зчитування і надсилання команд і даних із ліній портів мікроконтролера;
- втручання у роботу спеціальних регістрів даних та індикації стану портів введення/виведення МК;

- зміна слідування порядку команд управління та/або перехоплення їх і потоків даних як у ядрі та/або області ядра мікропроцесорної системи, так і у зовнішній периферії;
- несанкціоноване зовнішнє втручання в роботу ліній передачі даних та/або вторинних ліній – зовнішніх ліній передачі інформації і інтерфейсів в мікроконтролері. Сюди також можна віднести несанкціоноване (стороннє) пересилання/ зчитування команд керування МК, зчитування інформаційних потоків та/або окремих послідовностей блоків даних прийому/передачі даних до/від МК;
- загрози і атаки, що полягають у блокуванні обчислювального процесу за сторонніми зовнішніми вхідним і вихідним каналами в т.ч. із втручанням по вторинним функціональним каналам, і такими як енергоживлення;
- загрози «нульового дня» і загрози запуску шкідливого коду шляхом впровадження в основну підпрограму (в т.ч. загрози запуску «сліпих/порожніх» циклів в підпрограмі, зміна і переповнення пам'яті МК шляхом запуску ресурсоємного програмного коду тощо);
- інші потенційні загрози фізичного і прямого електромагнітного впливу на мікропроцесорну систему.

Місця основних загроз в архітектурі мікроконтролера наведено на рисунку 1.

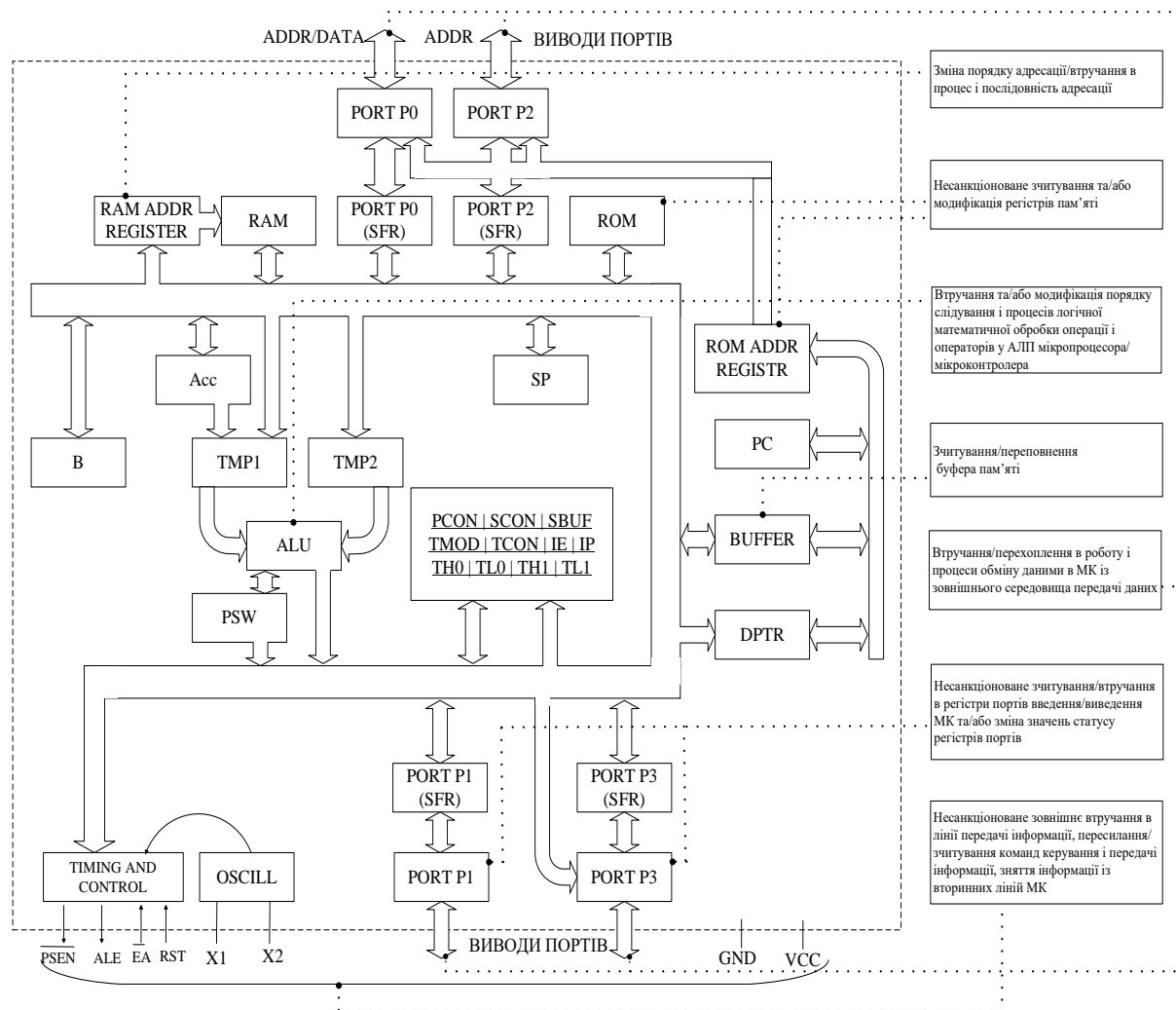


Рисунок 1 – Місця основних загроз в архітектурі мікроконтролера

Прояв основних загроз в архітектурі МК зумовлений недосконалістю окремих блоків, вузлів і зв'язків, а також наявністю вразливостей і незахищених місць і точок впровадження впливу та інформаційних втручань. До самих основних і небезпечних інформаційних загроз, та самих інтенсивних по характеру прояву в МК – є загрози втручання в пам'ять на різних рівнях. Іншим по інтенсивності прояву та характеру впливу є загрози по вторинним (побічним) каналам, які також проявляються.

Основні загрози пам'яті в МК архітектурі наведено на рисунку 2. Загрози пам'яті МК відносяться до основних фундаментальних кіберзагроз в МК і часто накладаються на рівень інформаційних загроз

ядра. Зони зчитування та запису даних при несанкціонованому втручанні і впровадженні стороннього функціоналу по побічним каналам в роботу мікроконтролера показано на рисунку 3.

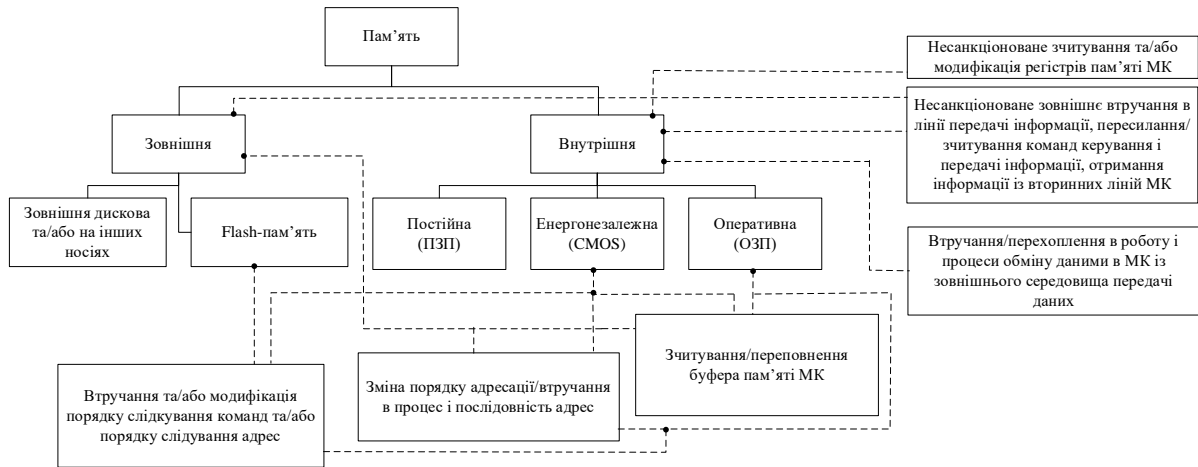


Рисунок 2 – Основні загрози пам'яті мікроконтролерної системи

Загрози пам'яті в МК можуть бути поділені на :

- втручання із прямим доступом до пам'яті (DMA-Direct Memory Acces);
- доступ до регістрів керування і доступ до буфера мікроконтролера;
- переповнення стеку буфера;
- зчитування буфера пам'яті;
- віддалений запуск коду;
- зовнішній доступ та атаки по вторинним каналам, а також по зовнішнім лініям передачі даних у МК;
- зміна порядку адресації в МК, зміна/підміна значень адрес і нумерації стеку;
- втручання в роботу регістрів даних та індикації стану портів введення/виведення мікроконтролера.

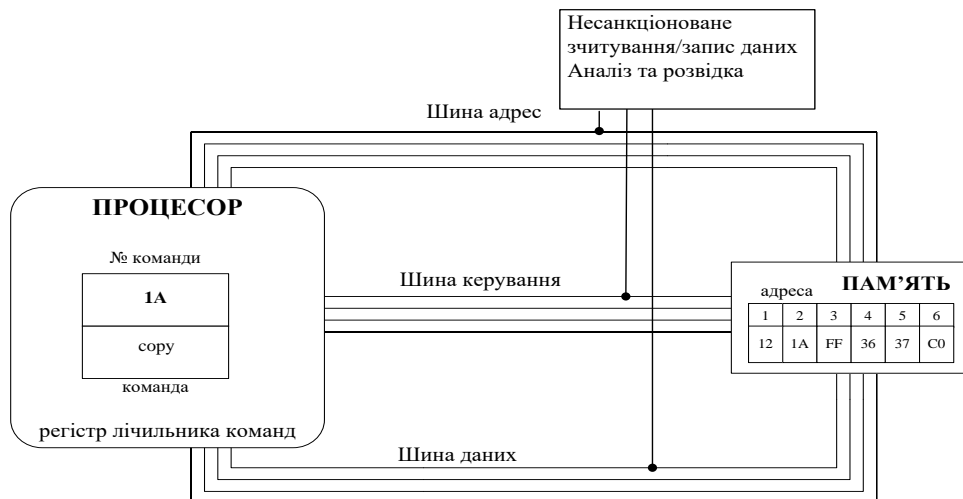


Рисунок 3 – Ілюстрація зон зчитування та запису даних при несанкціонованому втручанні в роботу мікроконтролера

Основними зонами несанкціонованого зчитування даних (рис. 3) є шина даних, шина адрес, шина керування, в яких дані представлені і можуть бути інтерпретовані при зчитуванні. Основний напрямок атак – втручання в шину адрес і шину керування: а) на рівні регістрів управління і регістрів стану; б) на рівні модифікації і з прямим підключенням сигналів до електричних ліній відповідних шин (якщо вони зовнішні і не входять у внутрішню топологію електронної інтегральної схеми МК).

Основні «слабкі» місця в архітектурі мікроконтролера для реалізації кібератак наведено на рисунку 4.

Базовими точками прояву загроз безпеки (рис. 4) і втручань в роботу мікроконтролера є: регістри; стек; АЛУ; пам'ять (EEPROM та Flash); порти введення-виведення і схеми та інтерфейси передачі даних в МК; схеми додаткового функціоналу підключення до зовнішніх периферійних пристроїв і канали підключення осцилятора тактової частоти.

Основні тенденції у МК-системах свідчать про підвищення ризиків впровадження кіберзагроз для рівня апаратного функціонування мікроконтролерів. Все більше значення і вагу з точки зору кібербезпеки функціоналу МК і засобів має стабільність та автентичність внутрішніх інформаційних процесів.

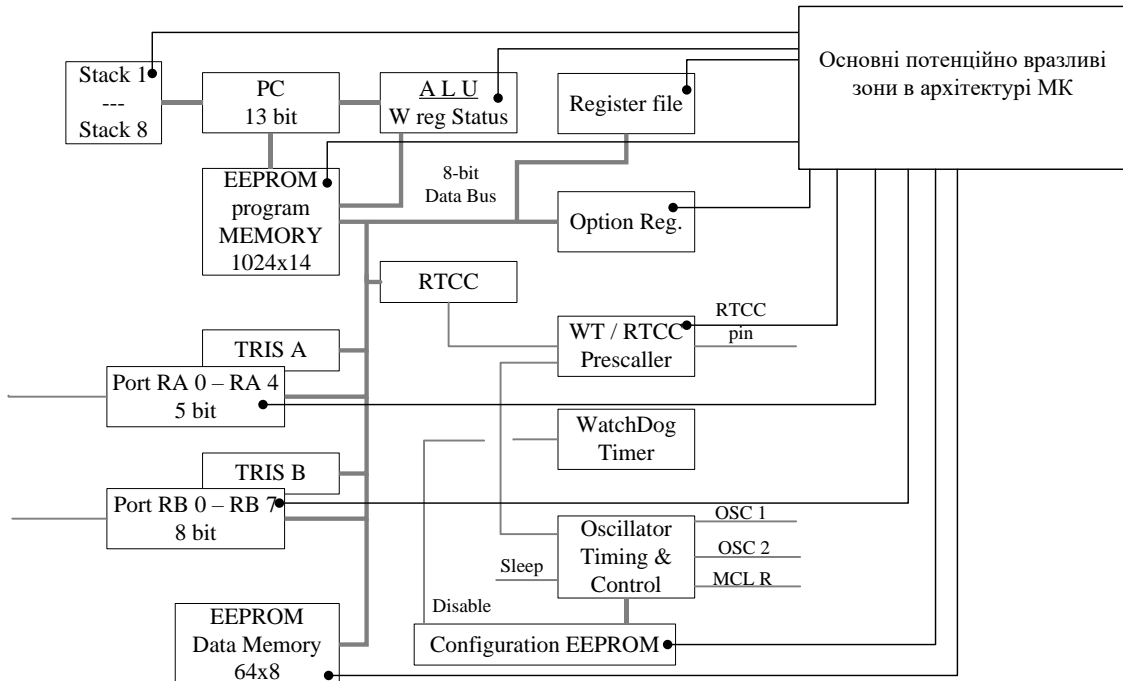


Рисунок 4 – Основні «слабкі» місця в архітектурі мікроконтролера для реалізації кібератак

До основних вразливостей, що проявляються в МК на фізичному рівні можна віднести [4, 9, 13, 14, 16, 17]:

- вразливості через вплив на команди і підміна команд керування;
- зміна порядку слідування команд і зміна технологічного циклу і алгоритму мікропрограми;
- недосконалість коду програми і потенційно наявні «слабкі» місця в машинному коді ПЗ ;
- вразливості в архітектурі МК, потенційно-небезпечні і можливі комбінації команд МК;
- вразливості пам'яті;
- не захищеність коду у ПЗП та ОЗП, кеш-пам'яті. Також не захищеність трактів АЦП і ЦАП;
- вразливості пов'язані із можливістю прямого читання буфера і його переповнення;
- вразливості ядра і таймерної системи, а також системи переривань мікропроцесора;
- вразливості і генерація замкнених пустих циклів і циклів переповнення пам'яті МК;
- неправомірне і нелогічне шкідливе використання механізму зовнішніх переривань і механізму апаратного і програмного скидання (функції: «reset»);
- вразливості «0»-го дня – не виявлені сучасні вразливості МК;
- випадкові загрози та/або недосконалості і помилки в програмному коді, наявність «не опрацьованих критичних» та/або проблемних місць в програмному коді;
- недосконалість і незахищеність архітектури МК.

На рисунку 5 наведено основні канали надходжень кіберзагроз із зовнішнього середовища.

На рисунку 6 наведено фрагмент сучасної електронної схеми із мікроконтролером та позначено найбільш ймовірні місця реалізації кібератак. Електронна схема МК є вразливим місцем в електричній частині і дозволяє при наявності фізичного доступу здійснювати підключення до ліній і портів МК. Основними потенційними кіберзагрозами є атаки зовнішніми каналами:

- 1) інформаційні втручання із зовнішнього середовища через порти;
- 2) через інтерфейси МК;
- 3) через схему і лінії управління;
- 4) втручання в механізм ланки тактового генератора частот МК.

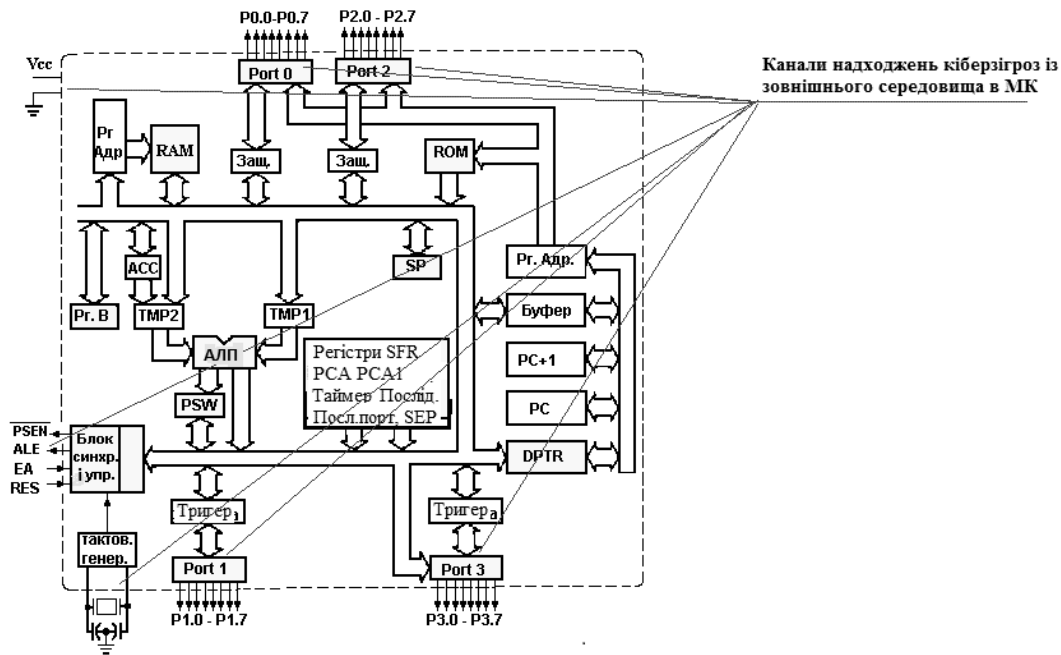


Рисунок 5 – Канали надходжень кіберзагроз із зовнішнього середовища у пам'ять і внутрішню структуру і систему МК (на базі архітектури RISC)

Найбільш високі за інтенсивністю >52% складають загрози типу 1) та 2), близько >25% складають загрози втручання в лінії керування - пункт 3). Втручання і підключення до схеми ланки тактового генератора частот МК як правило не велике >5-8% із врахуванням специфіки втручання в механізм формування тактів МК.

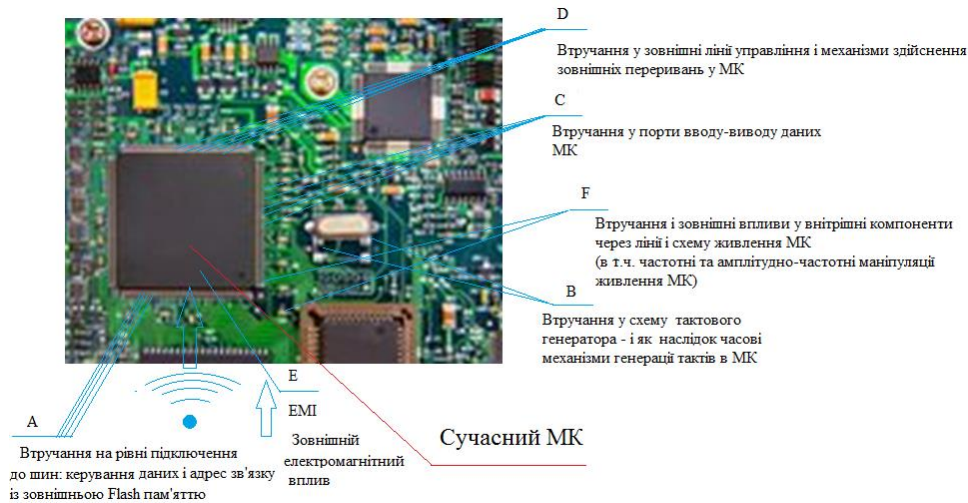


Рисунок 6 – Зовнішній вигляд сучасного МК в складі електронної схеми

Основними і найбільш значимими зовнішніми загрозами і інформаційними втручаннями у МК є (рис. 6) [9-18, 20-21]:

- втручання у зовнішні лінії і механізми переривань МК;
- втручання у порти введення-виведення даних у МК;
- втручання і зовнішні впливи на схему МК через схему живлення (в тому числі різні модуляції живлення);
- втручання у схему тактового генератора МК;
- втручання на рівні підключення до шин передачі даних і керування пам'яттю МК, а також обміну даними із іншими блоками і зовнішньою периферією;
- зовнішній електромагнітний вплив на МК.

Електронна схема і зовнішні електричні підключення МК до сторонніх периферійних пристроїв (в т.ч. зовнішньої Flash та/або EEPROM – пам'яті) є вразливим місцем і дозволяє при наявності фізичного доступу і підключення, або шляхом зовнішніх електромагнітних наведень здійснювати втручання та інформаційних вплив на процес оброблення даних в МК та здійснювати різноманітні атаки на нього.

Підходи до забезпечення безпеки процесів передачі і оброблення інформації в мікроконтролерах

Оскільки інформаційні загрози в мікропроцесорних трактах систем в т.ч. і мікропроцесорних пристроях Інтернету речей (IoT) досить часто є складними і мають комплексний характер і етапність проведення, то й рішення, спрямовані на захисти обчислювального процесу і алгоритмів роботи мікропрограм контролера керування також повинні мати комплексний підхід.

Для забезпечення закриття потенційно небезпечних критичних місць архітектури МК використовують одиничні і комплексні підходи для організації необхідного стану захищеності [9, 19-23]:

- Підходи, що ґрунтуються на апаратній основі, в яких використовується циклічний контроль надмірності коду (cyclic redundancy check calculate), тобто обчислюється контрольна сума, яка виявляє помилки при передачі або зберіганні даних. Це не тільки забезпечує перевірку цілісності коду, а й означає, що сигнатура може бути розрахована під час його роботи.

- Моніторинг живлення і моніторинг ресурсів – ще один метод із високим ступенем захисту. Для визначення причини скидання і, таким чином, забезпечення скидання тільки за допомогою автентифікованого доступу до системи управління статусом прапора POR/ PDR / BOR / PVD. Статуси, які забезпечує система: power-on reset – скид при увімкненні; power-down reset – скид при вимкненні; brown-out reset – скид при зниженні напруги живлення; programmable voltage detector - програмований детектор напруги. Для ефективного виявлення маніпуляцій та ведення журналу це доповнюється функцією «Read-While-Write» – читання під час запису, тобто зчитування одного слова під час запису іншого слова).

- Підходи, що передбачають використання ізольованості і контролю функціональності CSS (Clock Security System – «система безпеки тактування») заснована на тому, що якщо при використанні зовнішнього генератора (у мікроконтролерах серії ST32 він позначений як HSE) як джерело сигналу тактової частоти, що забезпечує стабільність роботи системи і гарантованість її «не зависання» в невизначеному стані, а зможе виконати якісь дії, SYSCLK або PLL (система фазового автопідстроювання частоти), відбудеться зрив генерації, то CSS автоматично переключить всю систему працювати від вбудованого RC-генератора (у мікроконтролерах серії ST32 він позначений як HSI). Таким чином, якщо щось трапиться з тактовими сигналами, можна перевести об'єкт управління мікроконтролером в безпечний стан. Крім того, сторожовий таймер (Watchdog) та віконний сторожовий таймер (Window Watchdog) також контролюють часові вікна незалежно один від одного.

- Контроль цілісності та достовірності вмісту пам'яті, що забезпечуються перевіркою та виправленням помилок коду Error Correction Code та перевіркою парності. Тут також забезпечується додатковий захист від атак, спрямованих на недопущення зараження систем помилками коду.

- Контроль зовнішніх фізичних і електричних параметрів МК. Наприклад, датчик температури безперервно вимірює температуру середовища, що оточує мікроконтролер. Це необхідно для того, щоб переконатися, що вона залишається в зазначеному діапазоні, і таким чином уникнути ризику пошкодження при спеціальному тривалому нагріванні.

- Використання сучасних інноваційних підходів до захисту мікропроцесорних систем – використання алгоритмів віртуалізації основного обчислювального процесу та його багаторівневе резервування копіюванням/фіксуванням і відновленням попередніх t_{i-1} , t_{i-2} , ..., t_{i-n} станів обчислювального процесу. У випадку настання кіберзагрози вектор параметрів і стану обчислювального процесу відновлюється із попередніх значень обчислювальних параметрів до моменту настання кібератаки в МК системі. У випадку настання кіберзагрози вектор параметрів і стану обчислювального процесу відновлюється із попередніх значень в часових проміжках t_{i-1} ; t_{i-2} ; t_{i-m} . t_m тобто:

$$\varphi(t_i, f(x, y, z, t, n, d)t_i) = \varphi'(t_i, f(x, y, z, t, n, d, t_i)) - \mu \Delta t_{i-1} (f(t_{i-1}, f(x, y, z, t, n, d, t_{i-1}))) ;$$

$$\varphi'(t_i, f(x, y, z, t, n, d, t_i)) = \sum_{j=0}^n \binom{n}{t} F(\varphi'(t_i, f(x, y, z, t, n, d, t_i)) - \mu \Delta t_{i-1} (f(dx, dy, dz, dt, dn, dd, t_{i-1})))$$

де $\varphi(t_i, f(x, y, z, t, n, d)t_i)$ – поточний вектору функції обчислювального процесу групи параметрів даних події при настанні кіберзагрози; $\varphi'(t_i, f(x, y, z, t, n, d, t_i))$ – актуальний параметр реальної функції у реальному часі; $\varphi'(t_i, f(x, y, z, t, n, d, t_i)) - \mu \Delta t_{i-1} (f(t_{i-1}, f(x, y, z, t, n, d, t_{i-1})))$ – функція попереднього стану вектору параметрів обчислювального процесу для групи параметрів даних x, y, z, t, n, d до події на-

стання кіберзагрози та/або кібератаки (вектором стабільних параметрів x, y, z, t, n, d – вважаються параметри, які не піддані кібератаці та/або інформаційному впливу відносно змінених параметрів x', y', z', t', n', d' в результаті інформаційного впливу та/або кібератаки); $f(t_{i-1}, f(x, y, z, t, n, d, t_{i-1}))$ – значення функції різниці зміни параметрів обчислювального процесу на дискретному проміжку часу Δt_{i-1} і коефіцієнта μ відносно попереднього стану із різницею в часі t_{i-1} ; t_i – актуальний час; $f(x, y, z, t, n, d, t_i)$ – значення функції обчислювального процесу в актуальному (реальному, поточному) часі із залежністю параметрів обчислювального процесу $x, y, z, t, n, d \in M_n$, де M_n – поле множини параметрів даних і змінних в обчислювальному процесі. $F(\varphi'(t_i, f(x, y, z, t, n, d, t_i)))$ – поточний вектору функції обчислювального процесу групи параметрів даних події при настанні кіберзагрози для окремої групи пристроїв периферії МК і взятої сукупності значень комплексної функції параметрів змінних і констант обчислювального процесу для локальної області пристроїв та/або периферії та або окремої області в МК; $f(t_{i-1}, f(x, y, z, t, n, \Delta t_{i-1}, t_{i-1}))$ – функція різниці зміни параметрів обчислювального процесу на дискретному проміжку часу Δt_{i-1} відносно попереднього стану за попередній проміжок часу t_{i-1} , взятої для окремої групи пристроїв та/або периферії та або окремої області в МК і взятої сукупності значень комплексної функції обчислювального процесу; n – сукупність пристроїв та/або периферії та або окремої області в МК.

Даний підхід в складі методу реалізується за рахунок відновлення попереднього стану обчислювального процесу, який може бути описаний правою частиною верхнього виразу формули (1), тобто за рахунок відновлення параметрів і значень функції із збережених копій і параметрів функції попередніх значень в пам'яті і за допомогою інших додаткових методів і засобів. Недоліком даного підходу є потреба у значній мірі додаткових ресурсів мікроконтролера і в т.ч. пам'яті для резервування попередніх станів обчислювального процесу.

– Використання криптографічних систем і алгоритмів обробки даних в мікроконтролерних системах із надійним криптографічним захистом. Даний підхід потребує зокрема спеціалізованої архітектури МК із криптографічною периферією (кодер/декодер) і відноситься до числа спеціалізованих надійних МК систем;

– Використання багаторівневої програмно-апаратної ізоляції, до якої відноситься: ізоляція програмного коду і методів доступу до потоків даних і потоків програмного коду команд і даних; фізична ізоляція електричної частини мікропроцесорної системи; інформаційна ізоляція мікропроцесорної системи; електрична ізоляція і в т.ч. електромагнітна ізоляція системи мікроконтролера; фільтрація вторинних шумів до/від МК, електрична ізоляція і фільтрація ліній живлення і ліній передачі даних від/до зовнішніх кіл, такі як датчики та/або кола управління; перевірка і ретельна кореляція програмного коду перед програмуванням/оновленням на предмет виявлення вразливостей в системі мікроконтролера; перевірка та моніторинг стану МК системи; використання шифрування і кодування даних для МК із підвищеним рівнем захисту (використовується в захищених і кіберстійких мікроконтролерних системах).

Для захисту від інформаційних втручань та впливу, кібератак і кіберзагроз в мікроконтролери автоматизованих і автоматичних систем, а також та IoT пристроїв в контексті сучасних підходів Industry X.0 та робототехніки, кіберфізичних систем, все більш актуальними стають комплексні підходи захисту, в т.ч. захист апаратного периметра підключення МК і втручання в роботу мікропрограм і ПЗ. Основні зусилля направляються на захист мікропрограм і пам'яті мікроконтролерів. Але не достатньо зусиль і уваги приділяється захисту інформаційним втручанням по вторинним каналам і лініям зв'язку із МК. Сучасні підходи передбачають визначення і використання захищених зон і периметру МК. Використовуються рішення в галузі автентифікації і криптографічного захисту архітектури МК, використання зон «0»-вої довіри при роботі МК платформи і взаємодії із іншими модулями.

Деякі сімейства мікроконтролерів вже містять багато функцій безпеки, а також функції забезпечення безпеки власного ПЗ. Річ у тім, що мікроконтролери є основними компонентами серед управління в підключених системах. Їхні постачальники вже використовують процеси розробки та сертифікацію за відповідними стандартами безпеки. А постачальники напівпровідників також гарантують, що можуть запропонувати своїм клієнтам комплексне безпечне рішення.

Висновки

Аналіз поширеності мікроконтролерів у сучасному світі довів свою затребуваність і актуальність використання. На сьогодні не можливо уявити роботу електронних пристроїв без мікроконтролерів. Це як і спеціалізовані обчислювальні пристрої, як пристрої загального призначення, в тому числі і різноманітні побутові пристрої. Оскільки мікроконтролери обробляють інформацію різного характеру і взаємодіють із різними периферійними пристроями, вони стають жертвами численних різноманітних кібератак через недоліки або помилки допущені при їх розробці. При цьому складність ситуації ще у тому, що виправити виявлені вразливості програмних шляхом не завжди виявляється можливим. У такому випадку

уся серія пристроїв стає непридатною до використання. Аналіз досліджень показав, що основні вразливості МК на фізичному рівні пов'язані із командами керування, мікропрограмою, пам'яттю та системою переривань. Серед атак найбільш розповсюдженими є зовнішні втручання у порти введення-виведення даних, впливи на схему МК через схему живлення, втручання у схему тактового генератора МК, підключення до шин передачі даних і керування пам'яттю МК, зовнішній електромагнітний вплив на МК та ін. Як показало дослідження загрози безпеки мікроконтролерам можуть бути досить критичними і потрібно занадто прискіпливо підходити до процесів їх проектування, тестування та виробництва адже одна вразливість може призвести до компрометації усього пристрою на його основі. При організації захисту доцільно притримуватися комплексного підходу для забезпечення багаторівневого захисту системи.

Список літератури

- [1] В. А. Лужецький, А. Д. Кожухівський, О. П. Войтович, *Основи інформаційної безпеки*. Вінниця: ВНТУ, 2013, 221 с.
- [2] Концепція технічного захисту інформації в галузі зв'язку України. [Електронний ресурс]. Режим доступу: <https://zakon.rada.gov.ua/laws/show/1126-97-%D0%BF#Text>. Дата звернення: 15 серпня 2022.
- [3] John R. Vacca, *Computer and Information Security Handbook*. Burlington, USA: Morgan Kaufmann Publishers, 2017, 1280 p.
- [4] С. Г. Антонов, С. М. Климов, "Методика оценки рисков нарушения устойчивости функционирования программно-аппаратных комплексов в условиях информационно-технических воздействий", *Надежность*, Том 17, №1, С. 32-39. 2017.
- [5] Software Security Guidance. [Online]. Available: <https://www.intel.com/content/www/us/en/developer/topic-technology/software-security-guidance/overview.html>. Accessed on: August 15, 2022.
- [6] V. S . Kharchenko, Internet of Things for Industry and Human Application. In Volumes 1-3. Volume 1. Fundamentals and Technologies. Ministry of Education and Science of Ukraine, National Aerospace University KhAI, 2019, 605p.
- [7] V. V. Sklyar, V. V. Yatskiv, N. G. Yatskiv, Dependability and Security Internet of Things: Practicum. Ministry of Education and Science of Ukraine, National Aerospace University "KhAI", Ternopil National Economic University, 2019, 98 p.
- [8] Cisco cybersecurity reports. [Online]. Available: https://www.cisco.com/c/en_hk/products/security/security-reports.html. Accessed on: August 15, 2022.
- [9] Meltdown and Spectre: Which systems are affected by Meltdown? [Online]. Available: <https://meltdownattack.com/#faq-systems-meltdown>. Accessed on: August 15, 2022.
- [10] The Anatomy of Security Microcontrollers for IoT Applications. [Online]. Available: <https://www.digkey.com/en/articles/the-anatomy-of-security-microcontrollers-for-iot-applications>. Accessed on: August 15, 2022.
- [11] Speculative Processor Vulnerability [Online] Available: <https://developer.arm.com/Arm%20Security%20Center/Speculative%20Processor%20Vulnerability>. Accessed on: August 15, 2022.
- [12] Cache Speculation Side-channels white paper. ARM Developer Forum. Specification. [Online]. Available: <https://developer.arm.com/documentation/102816/0205>. Accessed on: March 8, 2022.
- [13] Kernel Side-Channel Attack using Speculative Store Bypass - CVE-2018-3639. [Online]. Available: <https://access.redhat.com/security/vulnerabilities/ssbd>. Accessed on: March 8, 2022.
- [14] ISO/IEC, «Information technology — Security techniques-Information security risk management» ISO/IEC FIDIS 27005:2008. [Online]. Available: <https://www.iso.org/standard/42107.html>. Accessed on: August 15, 2022.
- [15] Modern security for microcontrollers. [Online]. Available: <https://get.meriac.com/docs/eSAME-MicrocontrollerSecurity.pdf>. Accessed on: August 15, 2022.
- [16] S. Yegulalp. Rowhammer hardware bug threatens to smash notebook security. [Online]. Available: <https://www.infoworld.com/article/2894497/rowhammer-hardware-bug-threatens-to-smash-notebook-security.html>. Accessed on: August 15, 2022.
- [17] K. Bains, J. Halbert, C. Mozak, T. Schoenborn and etc. "Row hammer refresh command", U.S. Patent Appl. 2014/0059287 A1, Feb. 27, 2014. [Online]. Available: <https://patents.google.com/patent/US20140059287>. Accessed on: August 15, 2022.
- [18] Cisco Systems security advisory. Row Hammer Privilege Escalation Vulnerability. [Online]. Available: <https://training.ti.com/core-cybersecurity-concepts-and-their-relation-microcontroller-security-hardware> Accessed on: August 15, 2022.
- [19] Core cybersecurity concepts and their relation to microcontroller security hardware. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/csa/cisco-sa-20150309-rowhammer.html>. Accessed on: August 15, 2022.

- [20] S. Govindavajhala, A. W. Appel. "Using Memory Errors to Attack a Virtual Machine". [Online]. Available: <https://www.cs.princeton.edu/~appel/papers/memerr.pdf>. html. Accessed on: August 15, 2022.
- [21] Yuan Xiao, Yinqian Zhang, Radu Teodorescu, Speechminer: a Framework for investigating and measuring speculative execution vulnerabilities. [Online]. Available: <https://arxiv.org/pdf/1912.00329.pdf>. Accessed on: August 15, 2022.
- [22] Introduction to STM32 microcontrollers security. [Online]. Available: https://www.st.com/resource/en/application_note/dm00493651-introduction-to-stm32-microcontrollers-security-stmicroelectronics.pdf. Accessed on: August 15, 2022.

Стаття надійшла: 04.09.2022.

References

- [1] V. A. Luzhetskyyi, A. D. Kozhukhivskyyi, O. P. Voitovych, *Osnovy informatsiinoi bezpeky*. Vynnytsia: VNTU, 2013, 221 p. [in Ukrainian].
- [2] Kontsepsiia tekhnichnoho zakhystu informatsii v haluzi zviazku Ukrainy. [Online]. Available: <https://zakon.rada.gov.ua/laws/show/1126-97-%D0%BF#Text>. Accessed on: August 15, 2022.
- [3] John R. Vacca, *Computer and Information Security Handbook*. Burlington, USA: Morgan Kaufmann Publishers, 2017, 1280 p.
- [4] S.H. Antonov, S.M. Klymov, "Metodyka otsenky ryskov narusheniya ustoichyvosti funktsionirovaniya programmno-apparatnykh kompleksov v usloviakh ynformatsyonno-tekhnicheskikh vozdeistviy", *Nadezhnost*, Tom 17, №1, 32-39 pp. 2017 [in Russian].
- [5] Software Security Guidance. [Online]. Available: <https://www.intel.com/content/www/us/en/developer/topic-technology/software-security-guidance/overview.html>. Accessed on: August 15, 2022.
- [6] V. S . Kharchenko, Internet of Things for Industry and Human Application. In Volumes 1-3. Volume 1. Fundamentals and Technologies. Ministry of Education and Science of Ukraine, National Aerospace University KhAI, 2019, 605p.
- [7] V. V. Sklyar, V. V. Yatskiv, N. G. Yatskiv, Dependability and Security Internet of Things: Practicum. Ministry of Education and Science of Ukraine, National Aerospace University "KhAI", Ternopil National Economic University, 2019, 98 p.
- [8] Cisco cybersecurity reports [Online]. Available: https://www.cisco.com/c/en_hk/products/security/security-reports.html. Accessed on: August 15, 2022.
- [9] Meltdown and Spectre: Which systems are affected by Meltdown? [Online]. Available: <https://meltdownattack.com/#faq-systems-meltdown>. Accessed on: August 15, 2022.
- [10] The Anatomy of Security Microcontrollers for IoT Applications. [Online]. Available: <https://www.digikey.com/en/articles/the-anatomy-of-security-microcontrollers-for-iot-applications>. Accessed on: August 15, 2022.
- [11] Speculative Processor Vulnerability. [Online] Available: <https://developer.arm.com/Arm%20Security%20Center/Speculative%20Processor%20Vulnerability>. Accessed on: August 15, 2022.
- [12] Cache Speculation Side-channels white paper. ARM Developer Forum. Specification. [Online]. Available: <https://developer.arm.com/documentation/102816/0205>. Accessed on: March 8, 2022.
- [13] Kernel Side-Channel Attack using Speculative Store Bypass - CVE-2018-3639 [Online]. Available: <https://access.redhat.com/security/vulnerabilities/ssbd>. Accessed on: March 8, 2022.
- [14] ISO/IEC, «Information technology – Security techniques-Information security risk management» ISO/IEC FIDIS 27005:2008. [Online]. Available: <https://www.iso.org/standard/42107.html>. Accessed on: August 15, 2022.
- [15] . Modern security for microcontrollers. [Online]. Available: <https://get.meriac.com/docs/eSAME-MicrocontrollerSecurity.pdf>. Accessed on: August 15, 2022.
- [16] S. Yegulalp. Rowhammer hardware bug threatens to smash notebook security. [Online]. Available: <https://www.infoworld.com/article/2894497/rowhammer-hardware-bug-threatens-to-smash-notebook-security.html>. Accessed on: August 15, 2022.
- [17] K. Bains, J. Halbert, C. Mozak, T. Schoenborn and etc., "Row hammer refresh command", U.S. Patent Appl. 2014/0059287 A1, Feb. 27, 2014. [Online]. Available: <https://patents.google.com/patent/US20140059287>. Accessed on: August 15, 2022.
- [18] Cisco Systems security advisory. Row Hammer Privilege Escalation Vulnerability. [Online]. Available: <https://training.ti.com/core-cybersecurity-concepts-and-their-relation-microcontroller-security-hardware> Accessed on: August 15, 2022.
- [19] Core cybersecurity concepts and their relation to microcontroller security hardware. [Online]. Available: <https://www.cisco.com/c/en/us/support/docs/csa/cisco-sa-20150309-rowhammer.html>. Accessed on: August 15, 2022.

- [20] S. Govindavajhala, A. W. Appel. "Using Memory Errors to Attack a Virtual Machine". [Online]. Available: <https://www.cs.princeton.edu/~appel/papers/memerr.pdf>. html. Accessed on: August 15, 2022.
- [21] Yuan Xiao, Yinqian Zhang, Radu Teodorescu, Speechminer: a Framework for investigating and measuring speculative execution vulnerabilities. [Online]. Available: <https://arxiv.org/pdf/1912.00329.pdf>. Accessed on: August 15, 2022.
- [22] Introduction to STM32 microcontrollers security. [Online]. Available: https://www.st.com/resource/en/application_note/dm00493651-introduction-to-stm32-microcontrollers-security-stmicroelectronics.pdf. Accessed on: August 15, 2022.

Відомості про авторів

Маліновський Вадим Ігорович – кандидат технічних наук, доцент кафедри захисту інформації.

Куперштейн Леонід Михайлович – кандидат технічних наук, доцент кафедри захисту інформації.

V. I. Malinovskyi, L. M. Kupershtein

**SECURITY THREATS ANALYSIS OF
MICROCONTROLLERS**

Vinnitsia National Technical University, Vinnitsia

УДК 681.3.07

О. В. Сілагін, І. Р. Арсенюк, В. І. Месюра, С. В. Кукунін

ПОКРАЩЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ УПРАВЛІННЯ КРИПТОВАЛЮТНИМИ АКТИВАМИ

Вінницький національний технічний університет, Вінниця

Анотація. Робота присвячена удосконаленню дуже актуальної та важливої у наш час програмно-апаратної технології управління криптовалютними активами. Ця технологія, як і додатки, що її реалізують, одержала назву «криптовалютний гаманець». З метою визначення пріоритетних напрямків удосконалення даної технології було детально проаналізовано сучасні програмно-апаратні рішення існуючих додатків по управлінню криптовалютними активами. Аналіз та вибір методологічних а також технологічних рішень здійснювався, у першу чергу, із застосуванням критеріїв функціональності, зручності та безпеки. У результаті аналізу було показано, що найефективнішою, за вищевказаними критеріями, є технологія, що базується на створенні окремих, так званих «не касодіальних» мобільних додатків із прив'язкою до відбитку пальця або FACE ID. За результатами здійсненого аналізу було також виявлено, що жодне із наведених технологічних рішень не передбачає реалізацію функції автоматизованого оцінювання рівня привабливості пропозицій із продажу криптовалюти, а також подальшого ранжування пропозицій за цим показником. Саме тому, з метою розширення функціоналу інформаційної технології управління криптовалютними активами було розроблено нову інформаційну технологію, що передбачає уведення функцій оцінювання рівня привабливості пропозицій із продажу криптовалюти, а також ранжування пропозицій провайдерів, залежно від результатів такого оцінювання. Крім того, наведено етапи процесу оцінювання привабливості наявних пропозицій. Запропонована технологія, у свою чергу, є складовою частиною технології криптовалютного гаманця та використовує апарат нечітких множин. Також, у роботі формалізована та змодельована задача оцінювання рівня привабливості пропозицій продажу криптовалюти із використанням теорії нечіткої логіки та нечітких множин.

Ключові слова: блокчейн технології, управління криптовалютними активами, функціонал, автоматизоване оцінювання рівня привабливості пропозицій, ранжування пропозицій.

Abstract. The work is devoted to the improvement of a very relevant and important in our time hardware and software technology for managing cryptocurrency assets. This technology, as well as applications that implement it, is called "cryptocurrency wallet". In order to determine the priority areas for improving this technology, modern software and hardware solutions of existing applications for managing cryptocurrency assets were analyzed in detail. The analysis and selection of methodological and technological solutions was carried out, first of all, using the criteria of functionality, convenience and security. As a result of the analysis, it was shown that the most effective, according to the above criteria, is the technology based on the creation of separate, so-called "non-custodial" mobile applications linked to a fingerprint or FACE ID. According to the results of the analysis, it was also found that none of the above technological solutions provides for the implementation of the automated assessment function of the level of proposals attractiveness for the sale of cryptocurrency, as well as further ranking of proposals by this indicator. That is why, in order to expand the functionality of the information technology of cryptocurrency assets management, a new information technology has been developed, which provides for the introduction of functions for assessing the level of offer attractiveness for the sale of cryptocurrency, as well as ranking the provider offers, depending on the results of such an assessment. In addition, the stages of the process of assessing the attractiveness of existing proposals are given. The proposed technology, in turn, is an integral part of the cryptocurrency wallet technology and uses the fuzzy set apparatus. Also, the paper formalized and modeled the problem of assessing the level of attractiveness of cryptocurrency offer sales using the fuzzy logic theory and fuzzy sets.

Key words: blockchain technologies, cryptocurrency asset management, security, automatic search for the best exchange rate.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-33-43>.

Вступ

На початку третього тисячоліття з'явилися революційні веб-технології, які не лише розширили поняття валютних та контрактних відносин, а й мали також величезний соціальний ефект у тому, що повернули людству довіру до «електронного документу». Мова йде про блокчейн технології та пов'язані з ними криптовалюти. Блокчейн – це децентралізована база даних, заснована на одноранговій мережі, загальному реєстрі та криптографії публічного та приватного ключів. Коли цифрова угода здійснюється у блокчейн, вона групується у криптографічно захищеному блоці з іншими угодами, які відбулися в останні декілька хвилин і розсилається усією мережею. Підтверджений блок транзакцій потім датується та додається до ланцюга у лінійному, хронологічному порядку. Нові блоки перевірених транзакцій пов'язані з більш старими блоками, утворюють ланцюжок блоків, які показують кожну транзакцію, досягнуту в історії цього блокчейну. Увійшовши в блокчейн-мережу, користувач підключається до інших комп'ютерів мережі для того, щоб обмінюватися з ними даними: блоками і записами. Отримавши нові дані, кожен користувач перевіряє їх коректність, і, переконавшись у достовірності, зберігає їх у себе, а також передає коректні дані далі мережею [1 – 5].

Найбільшого поширення блокчейн технології набули у сфері створення віртуальних грошей – криптовалют, а це, у свою чергу, дало поштовх для створення нового класу веб-додатків для управління криптовалютними активами, що одержали назву «криптовалютних гаманців».

Актуальність

Отже криптовалютний гаманець – це, по суті, інформаційна технологія, реалізована у програмному додатку, за допомогою якого можна управляти криптовалютою або криптовалютами певного користувача. Основними функціями гаманця є зберігання, а також продаж та закупівля криптовалюти від інших користувачів. Але, на відміну від класичних банківських електронних кабінетів, криптовалютний гаманець має свою специфіку, пов'язану з використаннями блокчейн технологій. На сьогоднішній день від-

чувається дефіцит зручних, надійних і широкофункціональних сервісів з обслуговування криптовалют, тому тема дослідження є досить важливою та актуальною.

Аналіз відомих рішень

Хоча з моменту появи першої криптовалюти минуло лише близько 12 років, вже склалася певна класифікація криптовалютних гаманців. Так, за ступенем універсальності вони можуть бути одновалютними та мультивалютними, за типом зберігання криптовалюти їх можна розділити на «гарячі» та «холодні»; за типом зберігання приватних ключів – на «кастодіальні» та «не кастодіальні», а за типом інсталяції – на локальні, мобільні, серверні, апаратні, браузерні. У найпростішому варіанті – це просто дані, які забезпечують доступ до свого рахунку. Ці дані, залежно від типу гаманця, можуть являти собою стандартну пару «емейл + пароль», приватний ключ або seed-фразу.

Основним завданням гаманця є можливість зберігати, а також надсилати та отримувати криптовалюту від інших людей.

Відмінність між гарячим і холодним гаманцем полягає в тому, що гарячий гаманець працює при підключенні до інтернету, а холодний може працювати і без такого доступу. Гарячі електронні гаманці менш захищені, оскільки існує ризик крадіжки ваших персональних даних через інтернет, проте, при цьому вони більш затребувані серед користувачів. Холодні електронні гаманці, відповідно, більш безпечні.

Суть кастодіальних гаманців полягає у тому, що вони не дають доступу до свого приватного ключа, а просто зберігають його на своєму централізованому сервері. Найчастіше таке рішення надають криптовалютні біржі. Його перевага полягає в тому, що можна відновити доступ до облікового запису через пошту, якщо пароль був загублений. Недолік полягає у тому, що обліковий запис може бути заморожено у разі якогось втручання, а для розморожування, користувача можуть попросити пройти процедуру KYC (Know Your Customer – знай свого клієнта) підтвердження особистості. Також користувач може втратити гроші під час хакерських атак, що останнім часом є досить частим явищем.

Некастодіальні електронні гаманці працюють інакше. Вони надають повний контроль над своїми приватними ключами, не використовуючи сервер. Величезною перевагою такого рішення є те, що кошти належать тільки користувачеві. Ніхто інший не зможе ними заволодіти без його seed-фрази. Однак, у цьому полягає і недолік такого гаманця, адже, якщо seed-фраза буде втрачена, то доступ до гаманця вже ніяк не повернути.

Локальний (десктопний) гаманець – це програма, яка встановлюється на стаціонарний комп'ютер або ноутбук. Даний вид гаманців є одним з найскладніших для користувачів, але при цьому володіє найкращими показниками безпеки та анонімності. Слід відзначити, що найчастіше їх використовують досвідчені користувачі або компанії, які проводять розробки на блокчейн-технологіях. Десктопні електронні гаманці можна розділити на 2 види: товстий та тонкий.

- У випадку використання товстого гаманця передбачається завантаження на комп'ютер повної копії блокчейна. За фактом товстий гаманець криптовалют це повна нода мережі, яка не лише дозволяє вам керувати своїм рахунком, а й підтримує роботу блокчейна. З огляду на те, що об'єм блокчейн-біткоїна займає вже близько 250 Гб – для роботи гаманця, відповідно, потрібно високопродуктивне «залізо».

- Тонкий гаманець, на відміну від товстого, займає на комп'ютері всього кілька мегабайт пам'яті та встановлюється за пару хвилин. Це програма-клієнт, для роботи якої не потрібно завантажувати на комп'ютер увесь блокчейн. Він дозволяє створювати адреси криптовалют і виконувати транзакції. З блокчейном тонкі гаманці взаємодіють не безпосередньо, як товсті гаманці, а через сервер розробників програми. Тому вони вважаються менш захищеними, але, натомість, набагато зручнішими у використанні.

Апаратний гаманець криптовалют це окремий пристрій, що на вигляд нагадує «флешку». Такий блокчейн гаманець служить для «холодного» зберігання криптовалют і підключається до інтернету тільки тоді, коли потрібно зробити транзакцію. Апаратні гаманці надають зручний доступ до блокчейну з високим ступенем захисту, оскільки приватні ключі зберігаються тільки у пам'яті самого пристрою. Незважаючи на їх вартість (від 60 до 100 доларів), вони дозволяють здійснювати транзакції таким чином, що хакери не можуть до них дістатися. У випадку втрати такого гаманця ніхто крім вас не зможе нічого зробити із засобами, при цьому ви з легкістю зможете відновити до них доступ через seed-фразу на новий пристрій. Тому по співвідношенню надійності та зручності використання лідирують апаратні гаманці.

Web- або браузерні гаманці – ще один, досить простий тип гаманців для використання, він не вимагає від користувача якихось особливих знань у криптовалютах. Його основні переваги:

- користуватися гаманцем можна на різних пристроях, незалежно від вашого місця знаходження, головне, щоб був вільний доступ до Інтернету;
- немає необхідності у скачуванні усіх блоків мережі, що економить багато часу та дисковий простір;

- у більшості, подібні сервіси пропонують своїм користувачам додаткові зручності, такі як відсутність комісії на перекази між користувачами, надсилання монет іншим його користувачам на адресу електронної пошти або номер телефону.

Однак, ми повинні пам'ятати, що такі гаманці мають «кастодіальне» рішення. При використанні Web-гаманця доступ до коштів має і сторонній сервіс. Тому їх збереження залежить вже не тільки від самого користувача.

Мобільні гаманці криптовалют – це додатки, які можна встановити на мобільні пристрої (смартфони, планшети). Потрібно відзначити, що вони увібрали в себе всі кращі якості від перерахованих вище видів гаманців, адже можуть бути «не кастодіальними», досить анонімними і при цьому надають доступ до криптовалют у будь-якій точці світу де є Інтернет. Оскільки це окремих додаток, то найчастіше розробники наділяють його ще й корисними додатковими функціями. Що стосується безпеки, то мобільні гаманці займають «золоту середину», оскільки крім звичайного PIN-коду можуть мати прив'язку до відбитку пальця або FACE ID (зазвичай налаштовується користувачем за бажанням).

Деякі з наведених технічних рішень пропонують пошук мінімального криптовалютного курсу або ранжування продавців за величиною курсу продаж, але поки ще ніхто не запропонував функціонал з багатокритеріального ранжування пропозицій з продажу криптовалюти, з урахуванням усіх суттєвих особливостей ринку торгівлі криптовалютою. Комплексний показник, що враховує всі ці особливості можна було б назвати рівнем «привабливості» пропозиції. Для прийняття рішень на основі «привабливості» пропозиції доцільно було б використати одну з існуючих інтелектуальних технологій ідентифікації [6, 7].

Мета

Метою дослідження є покращення існуючої технології управління криптовалютними активами «криптовалютний гаманець» за рахунок розширення її функціоналу. А саме, введення функцій оцінювання рівня «привабливості» пропозицій з продажу криптовалюти та ранжування пропозицій провайдерів у залежності від результатів оцінювання, а також розробка технології, що їх реалізує.

Задачі

Для досягнення поставленої мети потрібно розв'язати такі задачі:

1. Сформулювати перелік умов, технічних особливостей та ринкових характеристик, які суттєво впливають на рівень «привабливості».
2. Вибрати та застосувати одну з інтелектуальних технологій ідентифікації.
3. Змодельовати базу знань та механізм виведення результату.
4. Покращити існуючу інформаційну технологію управління криптовалютними активами за рахунок додавання нового функціоналу у вигляді технології з оцінювання рівня «привабливості» пропозицій провайдерів з продажу криптовалюти та ранжування цих пропозицій.

Розв'язання задач

Формулювання умов, технічних особливостей та ринкових характеристик, які суттєво впливають на рівень «привабливості» пропозиції

На відміну від звичайних «класичних» валют, криптовалюти несуть у собі деякі специфічні відмінності, пов'язані з процедурами їх створення, забезпечення, передачі та зберігання. По суті кожна з них є віртуальною готівкою, образом унікальної комп'ютерної технології, «законсервованої» у вигляді ланцюжків блокчейн, направлених ациклічних графів, консенсусних реєстрів (ledger) та ін. До таких відмінностей можна віднести:

- криптовалюти не належать жодній державі або державним союзним утворенням;
- не мають централізованого управління та емітентів;
- забезпечують анонімність користувачів;
- у більшості фіксована верхня межа загального обсягу емісії;
- можлива як емісія, так і демісія;
- великий час підтвердження транзакцій;
- платіж (передача криптовалюти між адресами) відбувається без посередників (банків та операторів платіжних систем);
- платіж є незворотній – немає механізму скасування підтвердженої операції;
- менш залежні від коливань світової та локальних економік;
- практично не використовуються у банківській діяльності для створення цінних паперів;
- їх розповсюдження та використання безпосередньо залежить від технологічної зрілості регіонів.

Ураховуючи ці відмінності, ми можемо сформулювати певний онтологічний словник в області знань, яку можна охарактеризувати як оцінювання «привабливості» пропозицій з продажу криптовалюти.

Це буде по кожній із криптовалют: x_1 – вид криптовалюти; x_2 – об'єм продажу; x_3 – курс продажу криптовалюти; x_4 – величина комісійних; x_5 – стабільність криптовалюти; x_6 – наявність або відсутність

фіксації курсу на час обробки запиту; x_7 – швидкість проведення операції; x_8 – надійність виконання запиту, що означає те, що запит точно буде оброблено і неможливість його відхилення, що також може призвести до втрати коштів через стрибки курсу.

Одночасно він може бути представлений як вхідний вектор змінних $X(x_1, x_2, \dots, x_8)$ для технології ідентифікації, на виході якої одержуємо показник рівня «привабливості» пропозиції Y , так, як це наведено на рис. 1.



Рисунок 1 – Технологія ідентифікації рівня привабливості

Аналіз та вибір інтелектуальної технології ідентифікації

Серед інтелектуальних технологій, за допомогою яких можна розв'язати усі задачі проекту, можна виділити штучні нейронні мережі (ШНМ), генетичні алгоритми та нечітку логіку. Розглянемо їх детальніше.

ШНМ традиційно використовують у ряді різноманітних задач, наприклад комп'ютерного зору, розпізнавання мовлення, машинного перекладу, соціально-мережевої фільтрації, відеоіграх, та у медичному діагностуванні.

Недоліком цього методу є відсутність твердих правил щодо вибору швидкості навчання та розміру мережі для вирішення конкретного завдання, невизначеність у підборі кількості нейронів у шарі мережі та кількості шарів ШНМ. Також цей підхід потребує проведення дуже великої кількості експериментів.

Генетичні алгоритми можуть використовуватись для пошуку рішень у дуже великих і важких просторах пошуку.

Відразу ж відзначимо, що застосування генетичного алгоритму в даній задачі є недоцільним, тому що критерій добору хромосом і використання процедур є евристичним, що зовсім не гарантує знаходження найкращого рішення. Іншим недоліком є велика обчислювана складність.

Нечітка логіка [6] є узагальненням класичної логіки на випадок, коли істинність розглядається як лінгвістична змінна, що приймає значення типу: "дуже істинно", "більш-менш істинно", "не дуже хибно" і т. д.

Зазначені лінгвістичні значення (терми) представляються нечіткими множинами. Нечіткі множини – це засоби формалізації природно-лінгвістичних висловлювань та логічних висновків. Ідея, що лежить в основі формалізації причинно-наслідкових зв'язків між змінними «входи-виходи», полягає в описі цих зв'язків на природній мові з використанням теорії нечітких множин та лінгвістичних змінних. Моделі об'єктів будуються шляхом проектування та налаштування нечітких баз знань, що представляють собою сукупності лінгвістичних висловлювань типу ЯКЩО <входи>, ТО <виходи>. Налаштовуючи нечітку базу знань через зміну параметрів функцій належності лінгвістичних термів можна ідентифікувати нелінійні залежності з необхідною точністю. Нечіткі межі множини кількісних значень, що відповідають певному лінгвістичному терму змінних вхідного вектору з рис. 1, є серйозною запорукою для використання нечітких баз знань і нечіткого логічного виводу в задачі оцінювання «привабливості» пропозицій з продажу криптовалют.

Серед трьох, незалежних одна від одної, теорій ідентифікацій та прийняття рішень – нечітких множин, нейронних мереж та генетичних алгоритмів, для задачі оцінювання «привабливості» пропозицій з продажу криптовалют, найбільш доцільним є застосування теорії нечітких множин та нечіткої логіки. Основою створеної технології є формування із застосуванням методів нечіткої логіки, матричної бази знань та застосуванням до неї продукційної системи навчання та виводу. Головною перевагою застосу-

вання інтелектуальної технології нечітких множин є можливість налаштування та адаптації бази знань до конкретних умов прийняття рішень, а саме навчання.

Ієрархічна модель логічного виведення (формалізація та фазифікація моделі)

Нами розглядається об'єкт з одним виходом та n входами вигляду:

$$y = f_y(x_1, x_2, \dots, x_n), \quad (1)$$

де y – вихідна змінна; x_1, x_2, \dots, x_n – вхідні змінні.

Змінні x_1, x_2, \dots, x_n та y можуть бути кількісними і якісними.

Для встановлення залежності (1) будемо розглядати вхідні змінні $x_j, i = \overline{1, n}$ та вихідну змінну y як лінгвістичні змінні [6].

Для оцінки лінгвістичних змінних $x_j, i = \overline{1, n}$ та y будемо використовувати якісні терми з наступних терм-множин:

$A_i = \{a_i^1, a_i^2, \dots, a_i^{l_i}\}$ – терм-множина змінної $x_j, i = \overline{1, n}$, $D = \{d_1, d_2, \dots, d_m\}$ – терм-множина y , де a_i^p – p -й лінгвістичний терм змінної $x_j, p = \overline{1, l_j}, i = \overline{1, n}$. d_j – j -й лінгвістичний терм змінної y , m – число різноманітних рішень області, що розглядається. Потужності терм-множин $A_i, i = \overline{1, n}$ у загальному випадку можуть бути відмінними, тобто $l_1 \neq l_2 \neq \dots \neq l_n$.

Назви окремих термів $a_i^1, a_i^2, \dots, a_i^{l_i}$ можуть також відрізнятися один від одного для різноманітних лінгвістичних змінних $x_j, i = \overline{1, n}$.

Лінгвістичні терми $a_i^p \in A_i$ та $d_j \in D, p = \overline{1, l_j}, i = \overline{1, n}, j = \overline{1, m}$ будемо розглядати як нечіткі множини.

Для зручності одночасного оперування та створення експертних висновків згрупуємо ті показники, що формують оцінювання вартості купівлі криптовалюти в окрему групу, а ті, що оцінюють ризики під час купівлі криптовалюти в іншу, та замінимо лінійну модель вектора вхідних змінних (див. рис. 1) на ієрархічну. Ієрархічний взаємозв'язок між вхідними параметрами, групами вхідних параметрів і вихідною змінною (інтегральним показником), представимо у вигляді дерева рішень (рис. 2), якому відповідає система відношень (2) – (4). Одночасно фазифікуємо нечіткі змінні, а саме поставимо їм у відповідність деяку кількість нечітких лінгвістичних термів, кожен з яких представляє нечітку множину значень певного показника. Функції належності цих нечітких термів будуть визначені пізніше.

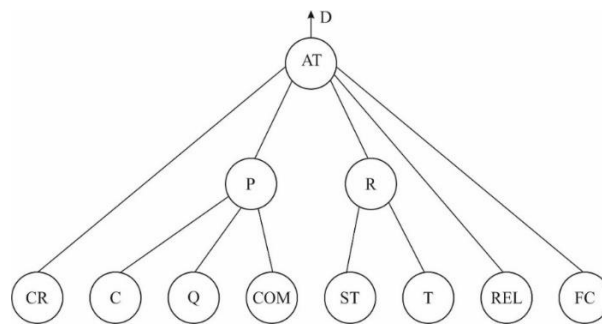


Рисунок 2 – Ієрархічне дерево рішень

$$AT = f_A(CR, P, R, FC, REL); \quad (2)$$

$$P = f_P(C, Q, COM); \quad (3)$$

$$R = f_R(T, ST), \quad (4)$$

де: AT (attractiveness) – нечітка логічна вихідна змінна, глобальний показник привабливості пропозиції, залежить від виду криптовалюти, локального показника оцінювання рівня вартості пропозиції криптовалюти, локального показника оцінювання рівня ризику втрат від нестабільності курсу та швидкості підтвердження транзакції, можливості фіксації курсу та показника рівня довіри до продавця, приймає значення одного з 11 лінгвістичних термів, які представляють популярну шкалу оцінювання товарів і послуг 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, що використовується для рейтингування товарів та послуг;

CR (cryptocurrency) – чітка вхідна змінна, код криптовалюти, представляє собою порядковий номер криптовалюти з таблиці криптовалют;

P (price) – нечітка логічна вихідна змінна, локальний показник оцінювання рівня вартості пропозиції криптовалюти, з врахуванням показника курсу, показника об'єму пропонованої валюти, показника величини комісійних, приймає значення: М – мала, МС – менше середньої, С – середня, БС – більше середньої, В – велика;

R (risk) – нечітка логічна вихідна змінна, локальний показник оцінювання рівня ризику втрат від нестабільності курсу та швидкості підтвердження транзакції, залежить від нестабільності курсу та часу підтвердження транзакції, приймає значення: М – малий, МС – менше середнього, С – середній, БС – більше середнього, В – великий;

C (course) – нечітка логічна вхідна змінна, показник рівня курсу за яким продається криптовалюта, формується по відношенню до математичного очікування курсових пропозицій по видам криптовалют. Приймає значення: М – малий, МС – менше середнього, С – середній, БС – більше середнього, В – великий.

Q (quantity) – нечітка логічна вхідна змінна, кількість пропонованої криптовалюти. Приймає значення: М – мала, МС – менше середньої, С – середня, БС – більше середньої, В – велика;

SOM (commission) – нечітка логічна вхідна змінна, показник величини комісійних під час проведення транзакції, формується по відношенню до математичного очікування комісійних провайдерів. Приймає значення: М – малий, МС – менше середнього, С – середній, БС – більше середнього, В – великий;

ST (stability) – нечітка логічна вхідна змінна, показник стабільності курсу криптовалюти. Приймає значення: М – мала, МС – менше середньої, С – середня, БС – більше середньої, В – висока;

T (time) – нечітка логічна вхідна змінна, показник величини часу на підтвердження запиту, формується по відношенню до усередненого часу обробки та підтвердження запиту. Приймає значення: М – малий, МС – менше середнього, С – середній, БС – більше середнього, В – великий;

REL (reliability) – нечітка логічна вхідна змінна, показник надійності продавця, демонструє рівень довіри до продавця в тому, що запит буде підтверджено, приймає значення: НВ – невідомий, М – малий, С – середній, В – великий;

FC (fixation of the course) – чітка вхідна бінарна змінна, показник фіксації курсу на час підтвердження транзакції, приймає значення 0, 1 (істина). При значенні 1 відключає вплив локального показника R.

Моделювання нечіткої бази знань

При моделюванні бази знань представимо співвідношення значень вхідних та вихідних змінних у вигляді таблиці (матричне представлення) [6, 7]. Задача експерта – розбити при цьому всі можливі доцільні комбінації значень восьми вхідних змінних на 11 груп, що відповідають значенням (термам) вихідної змінної.

Візьмемо N можливих доцільних комбінацій (за думкою експерта), які зв'язують входи і виходи об'єкта ідентифікації, і розподілимо їх таким чином:

$$N = k_1 + k_2 + \dots + k_m,$$

де k_j – число експериментальних даних, які відповідають вихідному рішенням $d_j, j = \overline{1, m}$, m – число вихідних рішень (для нашого випадку $m = 11$), причому, в загальному випадку $k_1 \neq k_2 \neq \dots \neq k_m$.

Передбачається, що $N < l_1 \cdot l_2 \cdot \dots \cdot l_n$, тобто число відібраних експериментальних даних менше повного перебору різних станів вхідного вектору X. Де $l_i (i = \overline{1, n})$ – число термів лінгвістичної змінної x_i , а n – кількість вхідних змінних.

Пронумеруємо N доцільних комбінацій таким чином:

11, 12, $1k_1$ – номери комбінацій вхідних змінних для вирішення d_1 ;

21, 22, $2k_2$ – номери комбінацій вхідних змінних для вирішення d_2 ;

...

$j1, j2, jk_j$ – номери комбінацій вхідних змінних для вирішення d_j ;

...

$m1, m2, mk_m$ – номери комбінацій вхідних змінних для вирішення d_m .

Матрицею знань для нашого випадку буде таблиця, сформована за такими правилами (табл. 1):

1) Розмірність матриці дорівнює $(n + 1) \times N$, де $(n + 1)$ – число стовбців, а $N = k_1 + k_2 + \dots + k_m$ – число рядків ($n = 8, m = 11$).

2) Перші n стовбців матриці відповідають вхідним змінним $x_i, i = \overline{1, n}$, а $(n + 1)$ – й стовбець відповідає значенням d_j вихідної змінної $y (j = \overline{1, m})$.

3) Кожен рядок матриці являє собою деяку комбінацію значень вхідних змінних, віднесених експертом до одного з можливих значень вихідної змінної y . При цьому: перші k_1 рядків відповідають значенню вихідної змінної $y = d_1$, другі k_2 рядків $y = d_2, \dots$, останні k_m рядків – значенню $y = d_m$.

4) Елемент a_i^{jp} , який стоїть на перетині i -го стовбця і jp -го рядка відповідають лінгвістичній оцінці параметра x_i у рядку нечіткої бази даних з номером jp . При цьому лінгвістична оцінка a_i^{jp} вибирається із терм-множини, яка відповідає змінній x_i , тобто $a_i^{jp} \in A_i$, $i = \overline{1, n}$, $j = \overline{1, m}$, $p = \overline{1, k_j}$.

Таблиця 1 – Узагальнена матриця знань

Номер вхідної комбінації значень	Вхідні змінні				Вихідна змінна
	x_1	x_2	$\dots x_i \dots$	x_n	
11	a_1^{11}	a_2^{11}	$\dots a_i^{11} \dots$	a_n^{11}	d_1
12	a_1^{12}	a_2^{12}	$\dots a_i^{12} \dots$	a_n^{12}	
\dots $1k_1$	$a_1^{1k_1}$	$a_2^{1k_1}$	$\dots a_i^{1k_1} \dots$	$a_n^{1k_1}$	
\dots					
$j1$	a_1^{j1}	a_2^{j1}	$\dots a_i^{j1} \dots$	a_n^{j1}	d_j
$j2$	a_1^{j2}	a_2^{j2}	$\dots a_i^{j2} \dots$	a_n^{j2}	
\dots jk_j	$a_1^{jk_j}$	$a_2^{jk_j}$	$\dots a_i^{jk_j} \dots$	$a_n^{jk_j}$	
\dots					
$m1$	a_1^{m1}	a_2^{m1}	$\dots a_i^{m1} \dots$	a_n^{m1}	d_m
$m2$	a_1^{m2}	a_2^{m2}	$\dots a_i^{m2} \dots$	a_n^{m2}	
\dots mk_m	$a_1^{mk_m}$	$a_2^{mk_m}$	$\dots a_i^{mk_m} \dots$	$a_n^{mk_m}$	

Уведена матриця знань визначає систему логічних висловлювань типу «ЯКЩО-ТО, ІНАКШЕ», які пов'язують значення вхідних змінних $x_1 \div x_n$ з одним із можливих типів вирішення d_j , $j = \overline{1, m}$:

ЯКЩО $(x_1 = a_1^{11}) \text{ I } (x_2 = a_2^{11}) \text{ I } \dots \text{ I } (x_n = a_n^{11})$ АБО
 $(x_1 = a_1^{12}) \text{ I } (x_2 = a_2^{12}) \text{ I } \dots \text{ I } (x_n = a_n^{12})$ АБО...
 $(x_1 = a_1^{1k_1}) \text{ I } (x_2 = a_2^{1k_1}) \text{ I } \dots \text{ I } (x_n = a_n^{1k_1})$,

ТО $y = d_1$, ІНАКШЕ

ЯКЩО $(x_1 = a_1^{21}) \text{ I } (x_2 = a_2^{21}) \text{ I } \dots \text{ I } (x_n = a_n^{21})$ АБО
 $(x_1 = a_1^{22}) \text{ I } (x_2 = a_2^{22}) \text{ I } \dots \text{ I } (x_n = a_n^{22})$ АБО...
 $(x_1 = a_1^{2k_2}) \text{ I } (x_2 = a_2^{2k_2}) \text{ I } \dots \text{ I } (x_n = a_n^{2k_2})$,

ТО $y = d_2$, ІНАКШЕ

ЯКЩО $(x_1 = a_1^{m1}) \text{ I } (x_2 = a_2^{m1}) \text{ I } \dots \text{ I } (x_n = a_n^{m1})$ АБО
 $(x_1 = a_1^{m2}) \text{ I } (x_2 = a_2^{m2}) \text{ I } \dots \text{ I } (x_n = a_n^{m2})$ АБО...
 $(x_1 = a_1^{mk_m}) \text{ I } (x_2 = a_2^{mk_m}) \text{ I } \dots \text{ I } (x_n = a_n^{mk_m})$,

ТО $y = d_m$.

(5)

де $d_j (j = \overline{1, m})$ – лінгвістична оцінка вихідної змінної y , яка визначається з терм множини D ; a_i^{jp} – лінгвістична оцінка змінної x_i в p -ому рядку j -ї диз'юнкції, яка вибирається з терм множини A_i , $i = \overline{1, n}$, $j = \overline{1, m}$, $p = \overline{1, k_j}$; k_j – кількість правил, які визначають значення вихідної змінної $y = d_j$.

Будемо називати подібну систему логічних висловлювань *нечіткою базою знань*.

Фрагмент нечіткої бази знань для локального показника Р, де $y = P$, $x_1 = C$, $x_2 = Q$, $x_3 = \text{СОМ}$, буде мати такий вигляд:

ЯКЩО $(C = M) \text{ I } (Q = M) \text{ I } (\text{СОМ} = M)$ АБО
 $(C = \text{МС}) \text{ I } (Q = M) \text{ I } (\text{СОМ} = M)$ АБО

$$\begin{aligned}
 & (C = M) \text{ I } Q = MC \text{ I } (COM = M) \text{ АБО} \\
 & (C = M) \text{ I } Q = M \text{ I } (COM = MC) \text{ АБО} \\
 & (C = MC) \text{ I } Q = MC \text{ I } (COM = M), \\
 \text{ТО } P = M, \text{ ІНАКШЕ,} \\
 & \text{ЯКЩО } (C = M) \text{ I } Q = MC \text{ I } (COM = MC) \text{ АБО} \\
 & (C = MC) \text{ I } Q = M \text{ I } (COM = MC) \text{ АБО} \\
 & (C = MC) \text{ I } Q = MC \text{ I } (COM = MC) \text{ АБО} \\
 & (C = C) \text{ I } Q = MC \text{ I } (COM = MC) \text{ АБО} \\
 & (C = MC) \text{ I } Q = C \text{ I } (COM = MC) \text{ АБО} \\
 & (C = MC) \text{ I } Q = MC \text{ I } (COM = C), \\
 \text{ТО } P = MC, \text{ ІНАКШЕ,} \\
 & \dots \\
 & \text{ЯКЩО } (C = B) \text{ I } Q = B \text{ I } (COM = BC) \text{ АБО} \\
 & (C = B) \text{ I } Q = BC \text{ I } (COM = B) \text{ АБО} \\
 & (C = BC) \text{ I } Q = B \text{ I } (COM = B) \text{ АБО} \\
 & (C = B \text{ I } Q = B) \text{ I } (COM = B), \\
 \text{ТО } P = B
 \end{aligned} \tag{6}$$

Аналогічно формуємо нечітку базу знань для локального показника R, де $y = R$, $x_1 = T$, $x_2 = ST$.

$$\begin{aligned}
 & \text{ЯКЩО } (T = M) \text{ I } (ST = M) \text{ АБО} \\
 & (T = MC) \text{ I } (ST = M) \text{ АБО} \\
 & (T = M) \text{ I } (ST = MC), \\
 \text{ТО } R = M, \text{ ІНАКШЕ,} \\
 & \text{ЯКЩО } (T = MC) \text{ I } (ST = MC) \text{ АБО} \\
 & (T = C) \text{ I } (ST = MC) \text{ АБО} \\
 & (T = MC) \text{ I } (ST = C), \\
 \text{ТО } R = MC, \text{ ІНАКШЕ,} \\
 & \dots \\
 & \text{ЯКЩО } (T = B) \text{ I } (ST = MB) \text{ АБО} \\
 & (T = MB) \text{ I } (ST = B) \text{ АБО} \\
 & (T = B) \text{ I } (ST = B), \\
 \text{ТО } R = B
 \end{aligned} \tag{7}$$

А також фрагмент бази знань для глобального показника AT, де $y = AT$, $x_1 = CR$, $x_2 = P$, $x_3 = R$, $x_4 = FC$, $x_5 = REL$.

$$\begin{aligned}
 & \text{ЯКЩО } (CR = 1) \text{ I } (P = M) \text{ I } (R = M) \text{ I } (FC = 0) \text{ I } (REL = B) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = M) \text{ I } (R = M) \text{ I } (FC = 0) \text{ I } (REL = B) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = M) \text{ I } (R = M) \text{ I } (FC = 0) \text{ I } (REL = B) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = M) \text{ I } (R = M) \text{ I } (FC = 0) \text{ I } (REL = B), \text{ ТО} \\
 \text{AT} = 10, \text{ ІНАКШЕ,} \\
 & \text{ЯКЩО } (CR = 1) \text{ I } (P = M) \text{ I } (R = M) \text{ I } (FC = 0) \text{ I } (REL = B) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = M) \text{ I } (R = M) \text{ I } (FC = 0) \text{ I } (REL = B) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = M) \text{ I } (R = M) \text{ I } (FC = 0) \text{ I } (REL = B) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = M) \text{ I } (R = M) \text{ I } (FC = 0) \text{ I } (REL = B), \text{ ТО} \\
 \text{AT} = 9, \text{ ІНАКШЕ,} \\
 & \dots \\
 & \text{ЯКЩО } (CR = 1) \text{ I } (P = B) \text{ I } (R = B) \text{ I } (FC = 0) \text{ I } (REL = HB) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = B) \text{ I } (R = BC) \text{ I } (FC = 0) \text{ I } (REL = HB) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = B) \text{ I } (R = B) \text{ I } (FC = 0) \text{ I } (REL = M) \text{ АБО} \\
 & (CR = 1) \text{ I } (P = BC) \text{ I } (R = B) \text{ I } (FC = 0) \text{ I } (REL = M), \text{ ТО} \\
 \text{AT} = 0.
 \end{aligned} \tag{8}$$

З використанням операцій \cup (АБО) і \cap (І) система логічних висловлювань (4) може бути подана у більш компактному вигляді:

$$\bigcup_{p=1}^{k_j} [\bigcap_{i=1}^n (x_i = a_i^{jp})] \rightarrow y = d_j, j = \overline{1, m}. \tag{9}$$

Таким чином шукане співвідношення (1), яке встановлює зв'язок між вхідними параметрами x_i та вихідною змінною y , формалізовано у вигляді системи нечітких логічних висловлювань (6) – (8), яка базується на уведеній нами матриці знань (табл. 1).

Функції належності лінгвістичних термів

За визначенням [6], функція належності $\mu^T(x)$ характеризує суб'єктивну міру (в діапазоні $[0, 1]$) впевненості експерта в тому, що чітке значення x відповідає нечіткому терму T . Найбільше розповсюдження в практичних додатках [8, 9] отримали трикутні, трапецієподібні, дзвоноподібні (гаусові) функції належності, параметри яких дозволяють змінювати форму функцій.

У роботі [7] запропонована проста та зручна для налаштування аналітична модель функцій належності змінної x довільному нечіткому терму T у вигляді:

$$\mu^T(x) = \frac{1}{1 + \left(\frac{x-b}{c}\right)^2}, \quad (10)$$

де b і c – параметри налаштування; b – координата максимуму функції $\mu^T(b) = 1$; c – коефіцієнт концентрації – розтягу функції (рис. 3).

Для нечіткого терма T число b являє собою найбільш можливе значення змінної x .

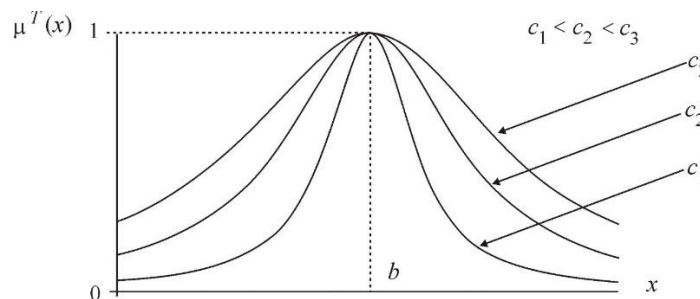


Рисунок 3 – Модель функції належності

Моделювання механізму виведення результату

У попередньому матеріалі ми визначилися:

- з множиною рішень $D = \{d_1, d_2, \dots, d_m\}$, які відповідають вихідній змінній y ;
- вектором вхідних змінних $X = (x_1, x_2, \dots, x_n)$;
- сформували матрицю знань.

Для виведення результату, скористаємося методом, що запропонований у роботі [7]. Ідея методу полягає у використанні нечітких логічних рівнянь. Ці рівняння будуються на базі матриці знань чи ізоморфній їй системі логічних висловлювань (5 – 7) і дозволяють обчислювати значення функцій належності різних розв'язків при фіксованих значеннях вхідних змінних об'єкта. Ці нечіткі логічні рівняння отримані з нечіткої бази знань (4) шляхом заміни лінгвістичних термів a_i^{jp} і d_j на відповідні функції належності, а операції \cup та \cap – на операції \vee і \wedge . Детально механізм нечіткого логічного виведення результату описаний в [7].

Розробка технології оцінювання рівня «привабливості» пропозицій з продажу криптовалюти та ранжування пропозицій в залежності від результатів оцінювання

Як говорилося раніше, розроблювана технологія рейтингування продавців криптовалюти за рівнем «привабливості» є складовою частиною загальної технології по управлінню криптовалютними активами під назвою «криптовалютний гаманець» і покращує цю технологію за рахунок розширення функціоналу.

Вихідними даними для роботи технології є:

1. Запит на купівлю криптовалют певного виду

$$\text{LIST_ORDER} (\text{CR1}(V1), \text{CR2}(V2), \dots, \text{CRN}(VN)), \quad (11)$$

де $\text{CR1}, \text{CR2}, \dots, \text{CRN}$ – види криптовалют; а $V1, V2, \dots, VN$ – відповідно, об'єми закупівлі різних видів криптовалют.

2. Перелік активних на даний момент провайдерів, що вийшли на ринок з пропозиціями на продаж певного виду криптовалют в певних об'ємах

$$\text{LIST_CR1}(\text{PROVIDER1}(Q1), \text{PROVIDER2}(Q2), \dots, \text{PROVIDERM}(QM)), \quad (12)$$

де PROVIDER1, PROVIDER2, ... , PROVIDERM – ідентифікатори провайдерів; Q1, Q2, ... , QM – відповідні об'єми продажу криптовалют.

Результатом роботи технології є список LIST_CR1_RAN цих же провайдерів, ранжований за рівнем «привабливості».

Кожен провайдер характеризується наступними атрибутами, які мають суттєвий вплив на привабливість пропозиції з продажу криптовалюти: PROVIDER (CR (cryptocurrency), C (course), Q (quantity), COM (commission), ST (stability), T (time), REL (reliability), FC (fixation of the course)), де CR – вид криптовалюти на продаж, C – курс продажу криптовалюти, Q – об'єм продажу, COM – величина комісійних, ST – стабільність криптовалюти, T – час проведення операції, REL – надійність виконання запиту, рівень довіри в тому, що запит точно буде оброблено і неможливість його відхилення, що також може призвести до втрати коштів через стрибки курсу, FC – наявність або відсутність фіксації курсу на час обробки запиту.

Під час роботи технологія використовує матрицю значень вхідних та вихідних змінних (табл. 1) та ізоморфну їй матрицю функцій належності, що в сукупності формує базу знань. Для заповнення матриці знань використовуються продукційні правила (6 – 8), які створюються із залученням експертів з біржової діяльності ринку криптовалют. Експертами задаються також первинні коефіцієнти b та c функцій належності – це буде етап «грубого» або структурного налаштування. Етап «тонкого» налаштування здійснюється автоматизовано у середовищах для моделювання систем на нечіткій логіці за методологіями та технологіями, описаними у [7]. Крім нечіткої бази знань технологія використовує також базу даних на продавців та ринкові характеристики криптовалют, яка у процесі експлуатації може поповнюватись та корегуватись. Так на основі інформації з бази даних формуються останні 4 показники, такі як ST (stability), T (time), REL (reliability), FC (fixation of the course).

Структуру технології наведено на рис. 5.



Рисунок 5 – Етапи роботи технології оцінювання привабливості пропозицій

Першим етапом роботи технології є формування списків провайдерів, що виступили з пропозиціями по певному виду криптовалюти LIST_CR1, LIST_CR2, ... , LIST_CRN.

Другим етапом є формування функцій належності в залежності від конкретного списку провайдерів по певній валюті.

Третій етап – нечітке логічне виведення результату оцінювання привабливості пропозиції провайдера

Четвертий етап – формування черг провайдерів (ранжування) для закупівлі криптовалют відповідно до рівня «привабливості» LIST_CR1_RAN, LIST_CR2_RAN, ... , LIST_CRN_RAN .

Висновки

У результаті проведених досліджень покращено інформаційну технологію управління криптовалютними активами за рахунок розширення її функціоналу, а саме: уведення функцій оцінювання рівня «привабливості» пропозицій з продажу криптовалюти та ранжування пропозицій провайдерів, залежно від результатів оцінювання. Також розроблена інформаційна технологія, що їх реалізує.

Наукова новизна:

- покращено існуючу інформаційну технологію управління криптовалютними активами «криптовалютний гаманець»;
- уперше формалізована та змодельована засобами нечіткої логіки задача оцінювання привабливості пропозицій з продажу криптовалюти;
- на основі теорії нечітких множин розроблена нова інформаційна технологія оцінювання привабливості пропозицій з продажу криптовалюти.

Список літератури

- [1] Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>. Accessed on: September 10, 2022.

- [2] Christidis Konstantinos, Michael Devetsikiotis, *Blockchains and Smart Contracts for the Internet of Things*. [Online]. Available: <http://ieeexplore.ieee.org/iel7/6287639/6514899/07467408.pdf?arnumber=7467408>. Accessed on: September 10, 2022.
- [3] Boohyung Lee, "Blockchain-based secure firmware update for embedded devices in an Internet of Things environment", *The Journal of Supercomputing*, pp. 1-16. 2016.
- [4] E. C. Ferrer, "The blockchain: a new framework for robotic swarm systems", arXiv preprint arXiv:1608.00695, 2016.
- [5] Andreas M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies*. "O'Reilly Media, Inc.", 2014, 298 p.
- [6] Л. Заде, *Понятие лингвистической переменной и ее применение к принятию приближенных решений*. М.: Мир, 1976, 167 с.
- [7] О. П. Ротштейн, *Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі*. Вінниця: «УНІВЕРСУМ – Вінниця», 1999, 320 с.
- [8] H. J. Zimmermann, *Fuzzy Sets, Decision Making and Expert Systems*. Kluwer: Dordrecht, 1987, 335p.
- [9] О. М. Роїк, А. В. Поплавський, "Нечіткий підхід до вирішення задачі ідентифікації кольорових відтінків", *Оптико-електронні інформаційно-енергетичні технології*, № 2, с. 39-42. 2016.

Стаття надійшла: 20.05.2021.

References

- [1] Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: <https://bitcoin.org/bitcoin.pdf>. Accessed on: September 10, 2022.
- [2] Christidis Konstantinos, Michael Devetsikiotis, *Blockchains and Smart Contracts for the Internet of Things*. [Online]. Available: <http://ieeexplore.ieee.org/iel7/6287639/6514899/07467408.pdf?arnumber=7467408>. Accessed on: September 10, 2022.
- [3] Boohyung Lee, "Blockchain-based secure firmware update for embedded devices in an Internet of Things environment", *The Journal of Supercomputing*, pp. 1-16. 2016.
- [4] E. C. Ferrer, "The blockchain: a new framework for robotic swarm systems", arXiv preprint arXiv:1608.00695, 2016.
- [5] Andreas M. Antonopoulos, *Mastering Bitcoin: unlocking digital cryptocurrencies*. "O'Reilly Media, Inc.", 2014, 298 p.
- [6] L. Zade, *Ponyatie lingvisticheskoy peremennoy i ee primenenie k prinyatiyu priblizhennyih resheniy*. M.: Mir, 1976, 167 s [in Russian].
- [7] O. P. Rotshtein, *Інтелектуальні технології ідентифікації: нечіткі множини, генетичні алгоритми, нейронні мережі*. Вінниця: «UNIVERSUM – Вінниця», 1999, 320 p. [in Ukrainian].
- [8] H. J. Zimmermann, *Fuzzy Sets, Decision Making and Expert Systems*. Kluwer: Dordrecht, 1987, 335p.
- [9] O. M. Roik, A. V. Poplavskiy, "Nechitkiy pidkhid do vyrishennia zadachi identyfikatsii kolorovykh vidtinkiv", *Optoelectronic Information-Energy Technologies*, № 2, pp. 39-42. 2016 [in Ukrainian].

Відомості про авторів

Сілагін Олексій Віталійович – кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук.

Арсенюк Ігор Ростиславович – кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук.

Месюра Володимир Іванович – кандидат технічних наук, доцент, професор кафедри комп'ютерних наук.

Кукунін Сергій Сергійович – провідний програміст, Spotlight Media Labs, Inc.

O. V. Silagin, I. R. Arseniuk, V. I. Mesiura, S. S. Kukunin

DESIGNING OF THE SYSTEM OF DIGITAL CORRECTION AND BITMAP IMAGES IMPROVING QUALITY IN THE RADIOGRAPHY FIELD

Vinnitsia National Technical University, Vinnitsia

УДК 004.624

П. В. Ставицький, В. В. Войтко, О. Н. Романюк

АНАЛІЗ ІНСТРУМЕНТІВ МЕТАПРОГРАМУВАННЯ В МОВАХ ПРОГРАМУВАННЯ ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ

Вінницький національний технічний університет, Вінниця

Анотація. Розглянуто та проаналізовано сучасні методи метапрограмування, що використовуються у мовах загального призначення. Підхід метапрограмування використовується для багатьох сценаріїв роботи з кодом програм, важливим серед них є генерування вихідного коду. Кожна мова програмування включає окремий набір інструментарію для вирішення завдань метапрограмування. Одним з методів метапрограмування є побудова процесорів анотацій, проте тут не визначаються особливості генерування коду. Іншим методом є побудова плагінів компіляторів, що є можливим у мовах програмування типу Kotlin. Інтерфейс плагінів може надавати доступ до багатьох стадій загального когнітивного навантаження. На противагу їм можна використовувати рядки з механізмом інтерполяції, що забезпечують декларативність, проте тут відсутня валідація типів. Мови програмування типу MetaOCaml, Scala реалізують механізми метапрограмування й багатоетапного програмування, зокрема, на рівні синтаксису з використанням конструкцій цитування і зрощування та механізмів вбудовування і макросів. Макроси присутні також у мовах C та C++, вони за допомогою директив препроцесора дозволяють виконувати попереднє перетворення коду перед основною стадією компіляції. Коректне поєднання елементів метапрограмування дозволить створити універсальний підхід до використання функціоналу декларативного метапрограмування, що надасть потужний інструментарій для масштабування обсягів згенерованого коду і підвищить якість кінцевого програмного продукту.

Ключові слова: метапрограмування, генерування коду, директиви препроцесора, макроси, Java, Kotlin, C, C++, MetaOCaml, Scala, цитування, зрощування.

Abstract. Modern methods of metaprogramming used in general-purpose languages are considered and analyzed. The metaprogramming approach is used for many scenarios of working with program code, but one of them is source code generation. Each programming language includes a unique set of tools for solving metaprogramming tasks. One of the methods of metaprogramming is the construction of annotation processors, however, they do not define the specifics of code generation. Another method is writing compiler plugins, which is possible in programming languages like Kotlin. The plugin interface can provide access to many stages of program compilation, but their disadvantage is the increased complexity of programs, which significantly affects the speed of development and the performance of the resulting software. Technologies such as KotlinPoet and JavaPoet allow generating the text of source programs with partial type safety at the level of statements. The disadvantage of such technologies is the significant difference between the code of the program that performs the generation and the original generated code, which increases the complexity of working with this tool and the overall cognitive load. In contrast to them, it is possible to use strings with an interpolation mechanism that provides declarativeness, but at the cost of type validation. Programming languages such as MetaOCaml and Scala implement mechanisms of metaprogramming and multi-stage programming, in particular, at the syntax level using quoting and splicing constructions and mechanisms of inlining and macros. Macros are also present in the C and C++ languages, they allow preprocessor directives to perform pre-transformation of the code before the main stage of compilation. The correct combination of metaprogramming elements will allow creating a universal approach to the use of a declarative metaprogramming functionality, which will provide a powerful toolkit for scaling the amount of generated code and increase the quality of the final software product.

Key words: metaprogramming, code generation, preprocessor directives, macros, Java, Kotlin, C, C++, MetaOCaml, Scala, quoting, splicing.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-44-50>.

Вступ

Основною задачею метапрограмування є отримання програмного коду як вихідних даних у результаті виконання інших програм [1]. Інструменти, які реалізують підходи метапрограмування, є розповсюдженими у великій кількості мов програмування загального призначення [2]. Кожна мова реалізує певний набір функцій, а у багатьох випадках перелік інструментів метапрограмування є унікальним для окремих мов [3].

Актуальність

Використання підходів метапрограмування є розповсюдженою практикою під час розробки програмного забезпечення при використанні будь-яких мов програмування загального призначення [1-2].

Генерування коду з використанням метапрограмування дозволяє автоматизувати процес написання повторюваних фрагментів, зменшуючи кількість коду, написаного вручну, сприяє уникненню помилок та проведенню оптимізацій [3].

Переваги такого підходу є наочними під час роботи з програмним забезпеченням, що складається з мільйонів рядків коду, адже кількість необхідного згенерованого коду за допомогою різних технологій значно зростає. Тому тема статті, що присвячена аналізу інструментів метапрограмування в мовах програмування загального призначення, є актуальною з огляду на формування рекомендацій щодо ефективного використання інструментарію метапрограмування в процесі розробки масштабних програмних проєктів.

Мета

Підвищення ефективності розробки програмного забезпечення шляхом формулювання й використання універсального підходу до реалізації функціоналу метапрограмування, а саме генерування коду як складової мов програмування загального призначення. Універсальність полягає у тому, що такий підхід може бути застосований до будь-якої мови програмування з мінімальними синтаксичними відмінностями і дозволить досягти поставленої мети під час генерування коду.

Задачі

1. Аналіз підходів метапрограмування, що використовуються в індустрії розробки програмного забезпечення, як складових мов програмування загального призначення.
2. Розробка підходів до універсального декларативного метапрограмування, які можуть бути застосованими як інструментарій розширення до існуючих мов програмування загального призначення без необхідності модифікації їх компіляторів та інтерпретаторів.

Розв'язання задач

Більшість мов програмування загального призначення реалізують ті чи інші підходи метапрограмування. Важливо розглянути та проаналізувати їх переваги й обмеження для розробки універсального рішення щодо генерування коду, що дозволить розширити синтаксис існуючих мов програмування загального призначення.

У рамках дослідження розглянемо низку інструментів метапрограмування, що застосовуються в різних мовах програмування та реалізуються за допомогою різних методів і підходів. Важливо розглянути кожний з таких підходів та визначити компоненти, які було б доцільно використати під час генерування вихідного коду як результату виконання цільової програми (рис. 1).



Рисунок 1 – Класифікація базових інструментів метапрограмування

У процесі аналізу інструментів метапрограмування мови Java окремо виділимо рефлексію. Рефлексія – це функція мови Java, що дозволяє програмам досліджувати або аналізувати себе та маніпулювати внутрішніми властивостями програми. Наприклад, клас Java може отримати імена всіх своїх методів і відобразити їх.

Метод рефлексії дозволяє виконувати певні модифікації програмного коду під час його виконання, але рефлексія не надає можливості генерування тексту нового коду для виконання компілятором пізніше. Крім того, використання рефлексії сповільнює процес виконання програми, тому не завжди є бажаним рішенням для поставлених завдань у метапрограмуванні.

Мови Java та Kotlin можуть використовувати анотації також для генерування тексту програмного коду. Процесори анотацій типу APT, KAPT та KSP дозволяють додавати логіку генерування коду, виходячи з відповідних компонентів програмних засобів, що вже використовують наперед визначені анотації [1]. Приклад програми, що виконує генерування декларованої сигнатури функції на основі відповідного інтерфейсу, наведено на рисунку 2.

На рисунку 2 зображено два фрагменти коду, де перший є складовою програми, що виконує генерування коду, а другий демонструє результуючий згенерований код. Декларативність такого підходу полягає в тому, що можна прослідкувати певну схожість між двома наведеними фрагментами коду. Перший фрагмент описує сигнатуру згенерованої функції. Список полів інтерфейсу з першого фрагменту є ідентичним списку аргументів згенерованої функції. Проте такий підхід є лише частково-декларативним, оскільки не дозволяє з аналогічним успіхом генерувати тіла функцій, вирази, здійснювати динамічне декларування функцій з використанням аргументів для варіативної кількості аргументів згенерованої функції й динамічного імені функції.

```

@Function(name = "example_function")
interface ExampleFunction {
    val arg1: String
    val arg2: Map<String, List<*>>
}

>
fun myAmazingFunction(
    arg1: String,
    arg2: Map<String, List<*>>
) {
    // ...
}

```

Рисунок 2 – Приклад використання процесора анотацій для частково декларативного генерування коду мови Kotlin

З іншого боку, метод декларативного метапрограмування може бути застосований під час написання процесора анотацій. При застосуванні декларативного підходу метапрограмування генерування коду відбуватиметься з використанням відповідних шаблонів, що описують загальну структуру генерованого коду, дозволяючи модифікувати певні деталі для кожної ітерації (рис. 3).

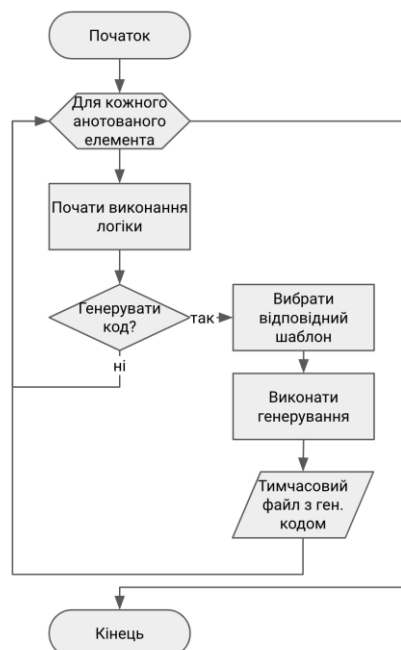


Рисунок 3 – Алгоритм генерування коду в ході роботи процесора анотацій з використанням декларативного методу метапрограмування на основі шаблонів

На рисунку 3 продемонстровано місце декларативного метапрограмування в процесі написання процесорів анотацій. Ключовим елементом є застосування декларативних шаблонів, що можуть бути використані кожною з ітерацій обробки певного анотованого елемента. Шаблони можуть містити код, що є ідентичним до бажаного згенерованого, і дозволяти використовувати аргументи для деталізації окремих складових результуючого коду. Використання процесорів анотацій відповідає на питання “Коли генерувати код?”, проте як саме відбуватиметься цей процес — залишається на розсуд розробника.

Технології JavaPoet та KotlinPoet у поєднанні з процесорами анотацій можуть підвищити ефективність генерування коду. Їх інтерфейс дозволяє досягти чіткої типізації на стадії написання логіки генерування коду, що дозволяє попередити значну кількість помилок на ранніх стадіях розробки. Приклад генерування тривіальної програми за допомогою KotlinPoet наведено на рисунку 4.

На рисунку 4 наведено приклад генерування простої програми, яка друкує повідомлення в консоль. Верхня частина рисунку демонструє фрагмент програми, що виконує генерування, а нижня частина містить згенерований код. Як можна побачити, код між такими фрагментами суттєво відрізняється навіть при написанні простої програми. Недоліком технології KotlinPoet є те, що фрагменти коду, які містять логіку генерування, значно відрізняються від вихідного коду, що підвищує когнітивне навантаження під час роботи з кодом та ускладнює його підтримку. Тому пропонуємо при використанні декларативного

підходу застосовувати шаблони коду зі збереженням високого рівня валідації типів, аналогічно до KotlinPoet.

```
val file = FileSpec.builder("", "Example")
    .addFunction(
        FunSpec.builder("main")
            .addStatement("println(%P)", "Example print!")
            .build()
    )
    .build()
file.writeTo(System.out)

>
fun main() {
    println("Example print!")
}
```

Рисунок 4 – Приклад імперативного генерування коду з KotlinPoet, що демонструє суттєву різницю між кодом програми, що генерує код, та вихідним кодом

Розглянувши проблему з іншого боку, зазначимо, що використання технології ktlint [2] може слугувати іншою точкою входу для генерування коду. На відміну від процесора анотацій, що виконує свою роботу під час компілювання коду, ktlint є статичним аналізатором коду, завданням якого є знаходження помилок у стилі написання програм та забезпечення автоматизації рефакторингу коду. В основі роботи з таким інструментом лежить набір правил, що валідує коректність використання тих чи інших конструкцій мови програмування Kotlin. Виконання рефакторингу відбувається окремо від стадії компіляції та виконання програми, що дозволяє ефективно генерувати код у тих випадках, коли початковий варіант не задовольняє стилістичним вимогам. Тоді проводиться аналіз абстрактного синтаксичного дерева (АСД), що будується на основі існуючого коду. Маючи доступ до АСД, існує можливість генерування нового коду з модифікаціями. Приклад створення власного правила для таких задач наведено на рисунку 5.

```
class CodeModificationRule : Rule("code-modification") {
    override fun visit(
        node: ASTNode,
        autoCorrect: Boolean,
        emit: (offset: Int, errorMessage: String, canBeAutoCorrected: Boolean) -> Unit
    ) {
        // Генерування коду на основі компонентів синтаксичного дерева
    }
}
```

Рисунок 5 – Приклад побудови правила ktlint для модифікування/генерування коду

Рисунок 5 демонструє приклад використання технології ktlint під час генерування коду та створення власного набору правил для цього. Існує можливість створення власного правила для вирішення завдань з генерування коду. Для цілей модифікації коду можна використовувати стандартний інтерфейс ktlint, що забезпечує доступ до абстрактного синтаксичного дерева існуючого коду в функціях типу visitor, що надає можливість його модифікації. Як і процесор анотацій, ktlint надає можливість визначити, коли саме необхідно виконувати генерування, проте він не надає інструментів для підвищення ефективності такого процесу. Декларативний підхід метапрограмування може бути застосовано як один з інструментів для генерування коду за допомогою ktlint. Під час створення декларативних шаблонів коду аргументами для них можуть бути компоненти синтаксичного дерева коду перед модифікацією (ASTNode). Аналогічні інструменти ktlint доступні на інших платформах у вигляді feross/standard для JavaScript, gofmt для мови програмування Go.

Компілятори мов програмування також виконують генерування коду в процесі компілювання для оптимізації програм. Синтаксис багатьох мов програмування забезпечує відповідні елементи для виконання такого генерування коду. Наприклад, мови програмування типу Kotlin, Scala, C++ використовують модифікатор inline [3] як вбудовування. Це усуває накладні витрати на з'єднання викликів і відкриває значні можливості для оптимізації. Оскільки функції з модифікатором inline не будуть використані за

посиланням, а їх тіло буде буквально скопійоване у місця їх виклику, схожий механізм можна використати для підходу декларативного метапрограмування. Скопійований фрагмент коду є коректним компільованим кодом, що, в свою чергу, надає усі функції статичної типізації та валідації. Якщо застосовувати такий код як шаблон і використати його як основу для генерування нового коду, можна забезпечити декларативність зі збереженням коректної типізації. Потрібно забезпечити варіативність шаблонів залежно від переданих аргументів та умов для кожної з ітерацій генерування нових файлів або фрагментів коду. Отримати доступ до копіювання таких шаблонів можна за допомогою власного плагіна компілятора, що може дозволити аналізувати і модифікувати внутрішню структуру коду програми в процесі її компіляції.

Мови програмування C та C++ використовують аналогічний підхід перед етапом роботи компілятора, а не під час його роботи. Таке рішення може бути досягнуто за допомогою макросів та директив препроцесора [4]. Функціонал макросів можна також розглянути в якості механізму для генерування коду на основі шаблонів зі схожими можливостями до зрощування. Проте замість доступу до внутрішньої структури коду тут можна виконати попередню його обробку та модифікацію. Таким чином, існує можливість додавання власного синтаксису до будь-якої з існуючих мов програмування. За допомогою власного препроцесора можна модифікувати код, який буде шаблоном, у такий, що буде зрозумілим оригінальному компілятору мови програмування.

Функціональна мова програмування OCaml початково не містить потужних засобів метапрограмування та генерування коду. Проте її розширення під назвою MetaOCaml [5] вирішує цю задачу шляхом використання концепцій цитування (quoting) та зрощування (splicing) [6]. Підхід є схожим до механізму інтерполяції рядків у мовах програмування типу Kotlin, де під час декларування семантики рядка можна динамічно вбудовувати посилання на змінні або вирази, значення яких буде отримано під час виконання програми. Аналогічно MetaOCaml дозволяє поєднувати функціональну логіку з вбудованими декларативними фрагментами генерованого коду.

Якщо використовувати підхід цитування та зрощування в якості розширення існуючих мов програмування, їх синтаксис може різнитися в залежності від особливостей конкретної мови. Проте, новий функціонал генерування коду додано як розширення оригінального компілятора без офіційних його оновлень.

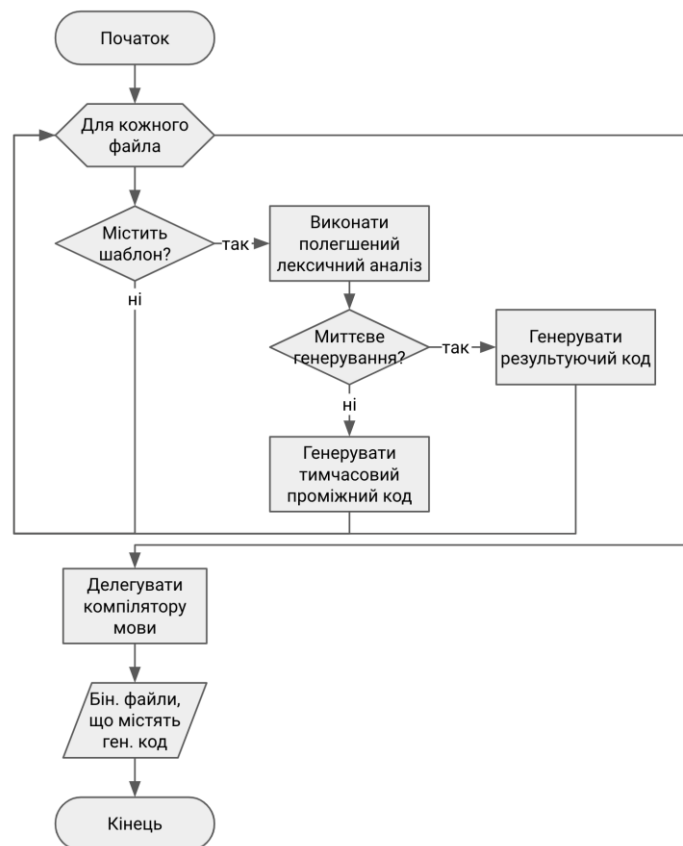


Рисунок 6 – Алгоритм використання макросів у процесі генерування коду

Цитування та зрошування зазвичай розширює синтаксис мов програмування. Під час розробки методу метапрограмування очевидно немає доступу до модифікації компіляторів існуючих мов програмування. Використовуючи підхід, аналогічний до макросів, існує можливість попередньої обробки коду, яка реалізує модифікований синтаксис і генерування проміжного коду, що є зрозумілим для оригінального компілятора мови програмування. Далі компілятор матиме змогу згенерувати бажаний код як результат виконання завдання. Такий алгоритм може використовуватися для кожного файлу, що містить розширений синтаксис цитування та зрошування, генеруючи проміжний код та делегуючи його компілювання і виконання оригінальному компілятору цільової мови програмування (див. рис. 6).

Під час обробки доданого синтаксису до існуючої мови програмування немає необхідності в повному синтаксичному і лексичному аналізі коду, тому спрощеного лексичного аналізу з визначенням символів цитування та зрошування буде достатньо. Далі існує можливість миттєвого генерування коду. У випадку, коли потрібна можливість відкладеного генерування коду під час виконання програми, генерується проміжний відповідний код у вигляді функцій, які генеруватимуть код на вимогу. Крім того, ті файли з кодом, що не містять додаткового синтаксису, можуть повністю ігноруватися.

Результати порівняльного аналізу методів метапрограмування на предмет можливості повернення згенерованого коду як вихідних даних та безпеки типів наведено у таблиці 1.

Таблиця 1 - Порівняльний аналіз методів метапрограмування

Метод метапрограмування	Надають можливість повертати згенерований код як вихідні дані	Забезпечують безпеку типів
Рефлексія	ні	так
Плагіни компілятора	ні	так
Процесори анотацій	частково	частково
Технології типу KotlinPoet	так	так
Статичні аналізатори коду	частково	частково
Вбудовування	ні	так
Макроси	так	так
Підхід інтерполяції рядків	так	ні
Підхід цитування та зрошування	так	так

Методи, які використовують процесори анотацій та статичні аналізатори коду, ідентифіковані значенням "частково", оскільки вони забезпечують платформу/фреймворк для генерування коду, але не визначають деталей. Це означає, що для генерування коду необхідно додатково обрати інший метод метапрограмування.

Висновки

Для створення ефективного інструмента генерування коду, що вирішує проблемні моменти наведених рішень, необхідно сфокусуватися на поєднанні різних можливостей розглянутих підходів до реалізації декларативного методу метапрограмування з дотриманням безпечної типізації.

Процесори анотацій мов програмування Kotlin або Java можуть бути використані для часткової реалізації методу декларативного метапрограмування. Під час розробки таких процесорів бажано використовувати інші технології, що надають потрібний функціонал. Схожим чином статичні аналізатори коду можуть організувати фреймворк для генерування коду, але не уточнювати, як саме це необхідно робити.

Методи зрошування та макросів мають багато спільного і дозволяють виконувати генерування/копіювання коду, проте їх ключовою відмінністю є стадія компіляції, протягом якої вони виконуються. Використання плагінів компілятора може надати доступ до внутрішнього стану програми під час компілювання, що дозволить використання певних кодових конструкцій як шаблонів для генерування інших фрагментів коду. Також макроси дозволяють надати можливість використання декларативних шаблонів коду окремо від стадії компілювання, дозволяючи розширювати синтаксис розповсюджених мов програмування за допомогою компонентів цитування та зрошування.

Таким чином, коректне поєднання елементів метапрограмування дозволить створити ефективний інструмент для вирішення завдань генерування коду, надаючи універсальний підхід до його застосування для великої кількості мов програмування загального призначення.

References

- [1] Juárez-Martínez, Ulises and José Oscar Olmedo-Aguirre, *Annotation Processing as Local Variable Crosscutting*, 2008.
- [2] Grégory Lureau, *Ktlint: a great 1st experience*. 2020. [Online]. Available: <https://www.glureau.com/2020/05/26/Ktlint-Moshi-Introduction/>. Accessed on: Sept. 4, 2022.
- [3] Theodoros Theodoridis, Tobias Grosser, and Zhendong Su. "Understanding and exploiting optimal function inlining," *27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '22) (2022)*. Association for Computing Machinery, New York, NY, USA, 977–989.
- [4] Medeiros, Flávio M., Márcio Ribeiro, Rohit Gheyi, Sven Apel, Christian Kästner, Bruno Ferreira, Luiz Carvalho and Balduino Fonseca, "Discipline Matters: Refactoring of Preprocessor Directives in the #ifdef Hell," *IEEE Transactions on Software Engineering*, 44 (2018), pp. 453-469.
- [5] Oleg Kiselyov, "The Design and Implementation of BER MetaOCaml – System Description," *FLOPS (2014)*.
- [6] Pouya Larjani, "On Meta Programming and Code Generation in F#," 2010/4/2.

Стаття надійшла: 20.09.2022.

Відомості про авторів

Ставицький Павло Валерійович – аспірант кафедри програмного забезпечення.

Войтко Вікторія Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення.

Романюк Олександр Никифорович – доктор технічних наук, професор, завідувач кафедри програмного забезпечення.

P. V. Stavytskyi, V. V. Voitko, O. N. Romanyuk

ANALYSIS OF METAPROGRAMMING CAPABILITIES IN GENERAL-PURPOSE PROGRAMMING LANGUAGES

Vinnitsia National Technical University, Vinnitsia

УДК 004.8

Д. А. Ткачик, Р. Н. Кветний

АНАЛІЗ ЕМОЦІЙНОГО ЗАБАРВЛЕННЯ ТЕКСТУ ДЛЯ ПРОГНОЗУВАННЯ ДАНИХ НА ФІНАНСОВИХ РИНКАХ

Вінницький національний технічний університет, Вінниця

Анотація. Прогнозування даних на фінансових ринках є актуальним завданням в сучасному світі, оскільки змога передбачати напрямки руху ринку допомагає інвесторам уникати очевидних ризиків та позбутись додаткових фінансових витрат. Розроблено багато різних торгівельних платформ для швидкого отримання доступу до великих масивів історичних даних, що дозволяє аналізувати фінансовий ринок з будь якої точки планети та в реальному часі, використовуючи тільки ноутбук чи стаціонарний персональний комп'ютер. Такі платформи дозволяють розробляти спеціальні стратегії та підходи на основі фундаментального або технічного аналізу, які враховують новини про ту чи іншу компанію, її дохід, капіталізацію та суму дивідендів, які вона повинна вчасно виплачувати, а також різні технічні індикатори. Новини про ту чи іншу компанію допомагають визначити потенційному інвестору певні ризики, зокрема, кадрові, виробничі чи, найчастіше в сучасних реаліях, репутаційні. Тому при формуванні фундаментального аналізу велику роль грає аналіз текстів новин, і саме тому найбільш ефективно це можна зробити за допомогою використання нейронних мереж. За останнє десятиліття нейронні мережі завдяки технологічним інноваціям та розробкам займають важливе місце в аналізі великих масивів даних, зокрема текстових. Оскільки кожна новина про ту чи іншу компанію, яка є об'єктом для потенційного інвестора чи трейдера, має в собі певне емоційне забарвлення, наприклад, позитивне чи негативне, то його можна визначити за допомогою саме навченої нейронної мережі, що допоможе вносити коректні прогнози на фінансових ринках та розробляти ефективні стратегії. В комплексі з технічним аналізом розробка та дослідження такого підходу до прогнозування може давати точні результати. Саме тому наукове дослідження в цьому напрямку є актуальним. В даній науковій статті обґрунтовано актуальність розробки системи прогнозування даних на основі аналізу емоційного забарвлення текстів. Розроблено програмне забезпечення на основі мови програмування Python. Описано результати дослідження, а також представлено вихідні дані роботи розробленої програми з отриманою точністю аналізу. За результатами наукового дослідження зроблено висновки.

Ключові слова: прогнозування, дані, нейронні мережі, фінансові ринки, фондова біржа, Python, інформаційні технології, емоційне забарвлення тексту, заголовок, інтернет трейдинг, штучний інтелект.

Abstract. Data forecasting in financial markets is an urgent task in today's world, as the ability to predict the direction of market movement helps investors avoid obvious risks and get rid of additional financial costs. Many different trading platforms have been developed to quickly gain access to large amounts of historical data, which allows you to analyze the financial market from anywhere on the planet and in real time, using only a laptop or a stationary personal computer. Such platforms allow you to develop special strategies and approaches based on fundamental or technical analysis, which take into account news about a particular company, its income, capitalization and the amount of dividends it should pay on time, as well as various technical indicators. News about this or that company helps a potential investor identify certain risks, in particular, personnel, production or, most often in modern realities, reputational. Therefore, the analysis of news texts plays an important role in the formation of fundamental analysis, and that is why it can be done most effectively using neural networks. Over the last decade, thanks to technological innovations and developments, neural networks have taken an important place in the analysis of large data sets, in particular text. Since every piece of news about one or another company, which is an object for a potential investor or trader, has a certain emotional color, for example, positive or negative, it can be determined with the help of a specifically trained neural network, which will help make correct predictions on financial markets and develop effective strategies. In combination with technical analysis, the development and research of such a forecasting approach can produce accurate results. That is why scientific research in this direction is relevant. This scientific article substantiates the relevance of developing a data forecasting system based on the analysis of the emotional coloring of texts. The software was developed based on the Python programming language. The results of the research are described, as well as the output data of the developed program with the obtained accuracy of the analysis are presented. Based on the results of the scientific research, conclusions were drawn.

Key words: forecasting, data, neural networks, financial markets, stock exchange, Python, information technology, sentiment analysis, headline, internet trading, artificial intelligence.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-51-58>.

Вступ

Процеси глобалізації інформаційної сфери та цифрової трансформації, розвиток інформаційного суспільства, впровадження передових інформаційно-комунікаційних технологій відкривають для трейдерів нові можливості для торгівлі цінними паперами. Використовуючи засоби Інтернету, учасники біржової торгівлі мають надшвидкий доступ до різних даних щодо ринку, що забезпечує кращу дохідність та ефективність.

З розвитком інтернет-трейдингу з'являються нові методи та підходи до прогнозування даних, які з кожним разом дають більшу точність та ефективність. Аналіз емоційного забарвлення тексту має безліч застосувань: наприклад такі дані дозволяють передбачити поведінку біржових трейдерів щодо конкретної компанії з відгуків в соціальних мережах чи при аналізі відповідних новин.

Аналіз емоційного забарвлення тексту означає визначення погляду чи емоції, що стоїть за ситуацією. В основному, це аналітика за фрагментом тексту чи промови або іншого способу спілкування. Кожний сектор економіки, з яким стикається клієнт (роздрібна торгівля, телекомунікації, фінанси тощо), зацікавлений у визначенні настроїв своїх клієнтів, незалежно від того, позитивні ці настрої чи ні.

Актуальність

Розробка систем для прогнозування даних є актуальною, так як в сучасному комп'ютеризованому світі значна кількість людей має доступ до інтернету, а також володіє персональним комп'ютером, що

забезпечує швидкий доступ до фондових бірж та дозволяє переглядати котирування цін в реальному часі, не виходячи з дому [1].

Завдяки розвитку нейронних мереж та підходів до обробки даних, стає можливим швидке та ефективно створення різних торгових стратегій та алгоритмів до прогнозування даних.

Аналіз емоційного забарвлення тексту може застосовуватися в будь-якій галузі, від фінансів і роздрібною торгівлі до сфери послуг та технологій. Нижче ми перерахували деякі з найпопулярніших способів використання аналізу настроїв у бізнесі [1]:

- 1) моніторинг соціальних мереж;
- 2) моніторинг бренду;
- 3) відгуки;
- 4) обслуговування клієнтів;
- 5) маркетингове дослідження.

Мета

Метою даної статті є розробка підходу для прогнозування даних на основі аналізу емоційного забарвлення тексту та приклад реалізації його з використанням мови Python. Впровадження таких засобів дозволить інтернет-трейдерам підвищити ефективність торговельних операцій на фондових біржах.

Результати досліджень

Аналіз емоційного забарвлення зосереджується на полярності тексту (позитивний, негативний, нейтральний), але він також виходить за межі полярності, щоб виявити конкретні почуття та емоції (злий, радісний, сумний тощо), терміновість (терміново, не терміново) і навіть наміри (зацікавлені, не зацікавлені) [2].

Залежно від того, як потрібно інтерпретувати ті чи інші новини, відгуки, коментарі, можна визначити та налаштувати свої категорії відповідно до потреб для аналізу тих чи інших настроїв [3]. Найпопулярніші підходи до аналізу емоційного забарвлення в тексті:

- поступовий аналіз настроїв;
- виявлення емоцій;
- аналіз настроїв на основі аспектів;
- багатомовний аналіз настроїв.

Якщо точність полярності важлива, то можна розглянути можливість розширення категорій полярності, щоб включити різні рівні як позитивних так негативних:

- дуже позитивно;
- позитивно;
- нейтрально;
- негативно;
- дуже негативно.

Це зазвичай називають градованим або поступовим аналізом настроїв, і його можна використовувати для інтерпретації 5-зіркових оцінок у огляді, наприклад, дуже позитивний це 5 зірок або дуже негативний це 1 зірка.

Виявлення емоцій дозволяє вийти за межі полярності, щоб виявити емоції, такі як щастя, розчарування, гнів і смуток.

Багато систем виявлення емоцій використовують лексикони (тобто списки слів і емоцій, які вони передають) або складні алгоритми машинного навчання.

Одним із недоліків використання лексиконів є те, що люди по-різному виражають емоції. Деякі слова, які зазвичай виражають гнів, як-от «погано» або «вбивати», також можуть виражати щастя.

Зазвичай, аналізуючи настрої текстів, потрібно знати, які конкретні аспекти чи особливості люди згадують у позитивному, нейтральному чи негативному ключі [4].

Ось де може допомогти аналіз настроїв на основі аспектів, наприклад, у цьому огляді продукту: «Час роботи батареї цього пристрою занадто малий», класифікатор на основі аспектів зможе визначити, що речення висловлює негативну думку щодо терміну служби акумулятора. відповідного продукту.

Багатомовний аналіз настроїв є більш складним. Це вимагає багато попередньої обробки даних та ресурсів. Більшість із цих ресурсів доступні в Інтернеті (наприклад, лексикони настроїв), тоді як інші потрібно створити, але потрібно знати, як кодувати, щоб їх використовувати.

Крім того, можна автоматично визначити мову в текстах за допомогою класифікатора мов, а потім навчити користувачку модель аналізу настроїв для класифікації текстів мовою за вибором.

Аналіз настроїв, інакше відомий як аналіз думок, працює завдяки обробці природної мови і алгоритмам машинного навчання, щоб автоматично визначити емоційний тон онлайн-розмов чи текстів.

Існують різні алгоритми, які можна застосувати в моделях аналізу настроїв, залежно від того, скільки даних потрібно проаналізувати, і наскільки точною очікується модель. Нижче ми розглянемо деякі з них більш детально.

Алгоритми аналізу настроїв поділяються на наступні три групи:

– на основі правил: ці системи автоматично виконують аналіз настроїв на основі набору правил, створених вручну;

– автоматичний: системи покладаються на методи машинного навчання;

– гібридні системи поєднують як засновані на правилах, так і автоматичні підходи.

Зазвичай система, на основі правил, використовує набір інструкцій, створених людиною, щоб допомогти визначити суб'єктивність, полярність або предмет певної точки зору чи позиції. Ці правила можуть включати різні методи, розроблені в комп'ютерній лінгвістиці, такі як:

- стемінг, лексема, тегування частини мови та синтаксичний аналіз;
- лексикони (тобто списки слів і виразів).

Наприклад як працює система на основі правил: визначає два списки поляризованих слів (наприклад, негативні слова, такі як поганий, найгірший, потворний тощо, та позитивні слова, такі як хороший, найкращий, красивий тощо), підраховує кількість позитивних і негативних слів, які зустрічаються в даному тексті. Якщо кількість позитивних слів більше, ніж кількість негативних слів, система повертає позитивний настрій, і навпаки. Якщо числа парні, система поверне нейтральні настрої.

Гібридні системи поєднують елементи методів, заснованих на правилах, і автоматичних методів в одну систему. Однією з переваг цих систем є те, що результати точніші [5].

На рисунку 1 зображено реалізацію класифікатора машинного навчання для аналізу емоційного забарвлення в тексті:

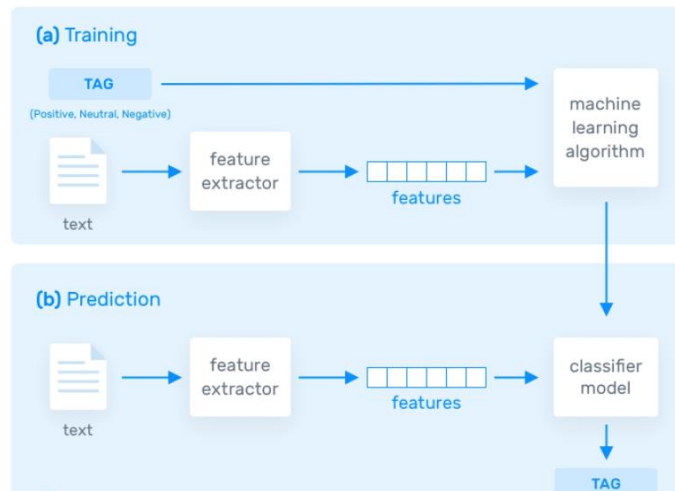


Рисунок 1 – Класифікатор машинного навчання

У процесі навчання (a) передається введений текст у вектор ознак. Пари векторів ознак і тегів (наприклад, позитивні, негативні або нейтральні) подаються в алгоритм машинного навчання для створення моделі.

У процесі прогнозування (b) ознаки використовуються для перетворення нерозпізнаних введених текстів у вектори ознак. Ці вектори ознак потім подаються в модель, яка генерує передбачені теги (негативні або нейтральні).

Першим кроком у класифікаторі тексту машинного навчання є перетворення вилученого тексту або його векторизації, і класичним підходом є bag-of-words або bag-of-ngrams [6].

Крок класифікації зазвичай включає статистичну модель, як

- наївний баєсів класифікатор;
- логістична регресія;
- опорні векторні машини;
- нейронні мережі.

Розробка системи для аналізу емоційного забарвлення тексту проводиться на основі мови програмування Python, а також бібліотеки vaderSentiment.

VADER (Valence Aware Dictionary and Sentiment Reasoner) — це інструмент аналізу настроїв на основі лексику та правил, який спеціально налаштований на емоції, виражені в соціальних мережах. Даний інструмент є повністю безкоштовним та відкритим, що значно спрощує його використання. Розроблений проект прогнозує, чи підвищиться або зменшиться ціна акцій компанії на основі заголовків головних новин.

Тестовий набір містить два датасети. На рисунку 3 зображено перший датасет – новини про фондовий ринок Доу-Джонса

	Date	Label	Top1	Top2	Top3	Top4	Top5
0	2016-07-01	0	b'Georgia 'downs two Russian warplanes' as cou...	b'BREAKING: Musharraf to be impeached.'	b'Russia Today: Columns of troops roll into So...	b'Russian tanks are moving towards the capital...	b'Afghan children raped with 'impunity,' U.N. ...
1	2016-06-30	1	b'Why wont America and Nato help us? If they w...	b'Bush puts foot down on Georgian conflict'	b'Jewish Georgian minister: Thanks to Israeli ...	b'Georgian army flees in disarray as Russians ...	b'Olympic opening ceremony fireworks 'faked''
2	2016-06-29	0	b'Remember that adorable 9-year-old who sang a...	b"Russia 'ends Georgia operation'"	b""If we had no sexual harassment we would hav...	b"Al-Qa'eda is losing support in Iraq because ...	b'Ceasefire in Georgia: Putin Outmaneuvers the...
3	2016-06-28	0	b' U.S. refuses Israel weapons to attack Iran:...	b"When the president ordered to attack Tskhinv...	b' Israel clears troops who killed Reuters cam...	b'Britain's policy of being tough on drugs is...	b'Body of 14 year old found in trunk; Latest (...
4	2016-06-27	1	b'All the experts admit that we should legalis...	b'War in South Osetia - 89 pictures made by a ...	b'Swedish wrestler Ara Abrahamian throws away ...	b'Russia exaggerated the death toll in South O...	b'Missile That Killed 9 Inside Pakistan May Ha...

Рисунок 3 – Дані нового заголовка про фондовий ринок Доу-Джонса

На рисунку 4 зображено другий датасет, що містить дані про ціну акцій на ринку Доу-Джонса.

```
df2= pd.read_csv('../input/stocknews/upload_DJIA_table.csv')
df2.head()
```

	Date	Open	High	Low	Close
0	2016-07-01	17924.240234	18002.380859	17916.910156	17949.369141
1	2016-06-30	17712.759766	17930.609375	17711.800781	17929.990234
2	2016-06-29	17456.019531	17704.509766	17456.019531	17694.679688
3	2016-06-28	17190.509766	17409.720703	17190.509766	17409.720703
4	2016-06-27	17355.210938	17355.210938	17063.080078	17140.240234

Рисунок 4 – Дані про ціну акцій на ринку Доу-Джонса

Потрібно об'єднати ці два датаести для бажаного набору даних. Результат об'єднання зображено на рисунку 5.

	Date	Label	Top1	Top2	Top3	Open	High	Low	Close
0	2016-07-01	0	b"Georgia 'downs two Russian warplanes' as cou...	b"BREAKING: Musharraf to be impeached."	b"Russia Today: Columns of troops roll into So...	11432.089844	11759.959961	11388.040039	11734.320312
1	2016-06-30	1	b'Why wont America and Nato help us? If they w...	b'Bush puts foot down on Georgian conflict'	b'Jewish Georgian minister: Thanks to Israeli ...	11729.669922	11867.110352	11675.530273	11782.349609
2	2016-06-29	0	b'Remember that adorable 9-year-old who sang a...	b"Russia 'ends Georgia operation'"	b""If we had no sexual harassment we would hav...	11781.700195	11782.349609	11601.519531	11642.469727

Рисунок 5 – Результат об'єднання двох вхідних датасетів

Далі з отриманого на рисунку 5 датасету потрібно видалити всі розриви та косі риси. Наступним важливим кроком є отримання суб'єктивності та полярності. Полярність означає силу думки [2]. Вона може бути позитивною або негативною. Якщо виникає з чимось пов'язане сильне позитивне почуття або емоція, наприклад, захоплення, довіра, любов; це справді матиме певну орієнтацію на всі інші аспекти існування цього об'єкта. Те ж саме стосується негативною полярності. Хорошим прикладом може бути такий: «Я не думаю, що куплю цей товар, тому що мій попередній досвід роботи з подібним товаром був не таким хорошим». Це матиме негативну полярність.

Сила позитивної і негативною полярностей може змінюватися в залежності від ситуації, але вони все ще вважаються позитивними або негативними. Користувачі зазвичай висловлюють свої емоції за допомогою таких слів, як «досить» або «трохи». Інструменти аналізу настроїв розглядатимуть їх як позитивні чи негативні. Найважливішим тут є те, що люди відчувають до чогось іншого через невербальне спілкування. Використання цих слів може вказувати на те, наскільки вони суб'єктивно оцінюють об'єкт [7].

Суб'єктивність означає ступінь особистої залученості людини в об'єкт. Тут найбільше значення мають особисті зв'язки та індивідуальний досвід з цим об'єктом, який може відрізнитися або не відрізнитися з точки зору когось іншого. Наприклад: «Я дуже задоволений своїм новим смартфоном, тому що він має найвищу продуктивність, доступну на ринку». Подібно до полярності, сильна суб'єктивність може бути негативною або позитивною. Твердження тут явно суб'єктивне, оскільки користувач насправді говорить про свій досвід і те, що він відчуває до об'єкта.

Інструменти аналізу настроїв повинні мати можливість розрізняти полярність і суб'єктивність, щоб правильно аналізувати думки користувачів. Думка може мати високий ступінь суб'єктивності, якщо вона висловлена як особистий досвід, тоді як низький ступінь може вказувати чужу точку зору на щось інше [8]. Настрої також можуть мати різний рівень полярності в різних способах спілкування – електронні листи, чати чи інші повідомлення можуть бути як позитивними, так і негативними; в деяких інших випадках вони можуть мати навіть нейтральну полярність.

Після визначення полярності та суб'єктивності слід провести оцінку на основі чотирьох критеріїв для кожного заголовку:

- негативний
- нейтральний
- позитивний
- комбінований (розраховано шляхом нормалізації наведених вище оцінок)

Після даних етапів отримуємо наступні дані, що зображені на рисунку 7:

	Label	Open	High	Low	Volume	Суб'єктивність	Полярність	Комбінований	Негативний	Позитивний	Нейтральний
0	0	11432.089844	11759.959961	11388.040039	212830000	0.267549	-0.048568	-0.9982	0.235	0.041	0.724
1	1	11729.669922	11867.110352	11675.530273	183190000	0.374806	0.121956	-0.9858	0.191	0.089	0.721
2	0	11781.700195	11782.349609	11601.519531	173590000	0.536234	-0.044302	-0.9715	0.128	0.056	0.816
3	0	11632.809570	11633.780273	11453.339844	182550000	0.364021	0.011398	-0.9809	0.146	0.066	0.788
4	1	11532.070312	11718.280273	11450.889648	159790000	0.375099	0.040677	-0.9882	0.189	0.094	0.717
...
1984	0	17355.210938	17355.210938	17063.080078	138740000	0.352311	-0.014015	-0.9644	0.148	0.094	0.758
1985	1	17190.509766	17409.720703	17190.509766	112190000	0.352649	0.046560	-0.9571	0.132	0.102	0.767
1986	1	17456.019531	17704.509766	17456.019531	106380000	0.389617	0.052622	-0.9975	0.225	0.091	0.684
1987	1	17712.759766	17930.609375	17711.800781	133030000	0.382566	0.011243	-0.9977	0.202	0.061	0.738
1988	1	17924.240234	18002.380859	17916.910156	82160000	0.320261	-0.035458	-0.9983	0.212	0.059	0.729

Рисунок 7 – Дані для створення та навчання моделі

За отриманими даними, що зображено на рисунку 7, створюємо модель для навчання, як показано на рисунку 8.

```
model= LinearDiscriminantAnalysis().fit(x_train, y_train)

predict= model.predict(x_test)
```

Рисунок 8 – Створення моделі для навчання

Після аналізу даних з допомогою інструментів vaderSentiment отримуємо підсумкову таблицю результатів навчання згенерованої моделі даних (рисунк 9).

	precision	recall	f1-score	support
0	0.86	0.79	0.83	193
1	0.82	0.88	0.85	205
accuracy			0.84	398
macro avg	0.84	0.84	0.84	398
weighted avg	0.84	0.84	0.84	398

Рисунок 9 – Результати роботи програми аналізу емоційного забарвлення тексту

Точність є одним з показників для оцінки моделей класифікації. Неформально, точність – це частка прогнозів, які модель отримала правильно. Формально точність має таке визначення:

$$\text{Точність} = \frac{\text{Кількість правильних прогнозів}}{\text{Кількість усіх прогнозів}} \quad (1)$$

Для двійкової класифікації точність також можна розрахувати в термінах позитивних і негативних показників таким чином:

$$\text{Точність} = \frac{TP+TN}{TP+TN+FP+FN}, \quad (2)$$

де TP = правильно позитивні результати, TN = правильно негативні результати, FP = помилкові позитивні результати та FN = помилкові негативні результати.

Проаналізувавши отримані результати, які зображені на рисунку 9, можна зробити висновок, що точність становить 0,84, або 84% (84 правильних прогнозів із 100 прикладів).

Висновки

Розробка систем для прогнозування даних є актуальною, так як в сучасному комп'ютеризованому світі значна кількість людей має доступ до інтернету, а також володіє персональним комп'ютером, що забезпечує швидкий доступ до фондових бірж та дозволяє переглядати котирування цін в реальному часі, не виходячи з дому [6].

Такі системи, як представлена у статті, призначені для полегшення аналізу ринку інвестором, вони допомагають фільтрувати ті чи інші ринки на фондових ринках для концентрації уваги на такі, що можуть принести більший і швидший прибуток.

Підхід, використаний в системі прогнозування даних, а саме аналіз емоційного забарвлення тексту, дозволяє підвищити ефективність прийняття рішень на фінансових біржах, зокрема, його можна використовувати в комплексі з іншими системами. Наприклад, система для прийняття рішень, яка базується на прогнозуванні історичних даних тільки за допомогою технічного аналізу, дає результат прогнозування приблизно 70%, як це зображено на рисунку 10 [9].

```
Total BUY positions: 505
Profit rate: 370 ( 73.26 % )
Stoploss rate: 134 ( 26.53 % )
```

Рисунок 10 – Результат роботи системи для підтримки прийняття рішень, що базується на технічному аналізі

Враховуючи точність описаного в статті підходу, результат роботи системи можна покращити щонайменше до 80%, за рахунок уникнення помилкових вхідних даних, навчання нейронної мережі та відбору більш стабільних фінансових об'єктів (компаній, біржових індексів і т.д.).

Даний підхід також може бути вдосконалений: зокрема, потрібно аналізувати не лише заголовки новин, а й самі новини. Також потрібно звернути увагу на критерії для поділу тексту: використовувати тільки позитивний та негативний, щоб уникати сумнівних прогнозів та покращити продуктивність.

Список літератури

- [1] F. Hamborg, K. Donnay, "NewsMTSC: A Dataset for (Multi-)Target-dependent Sentiment Classification in Political News Articles," Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Stroudsburg, PA, USA, 2021. [Online]. Available: <https://doi.org/10.18653/v1/2021.eacl-main.142>.
- [2] What is polarity and subjectivity in sentiment analysis? [Online]. Available: <https://www.quora.com/What-is-polarity-and-subjectivity-in-sentiment-analysis>.
- [3] Sentimental analysis using vader. [Online]. Available: <https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>.
- [4] Santur Y. Sentiment Analysis Based on Gated Recurrent Unit. 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 21–22 September 2019. 2019. [Online]. Available: <https://doi.org/10.1109/idap.2019.8875985>.
- [5] K. V. Nguyen et al., "UIT-VSFC: Vietnamese Students' Feedback Corpus for Sentiment Analysis," 10th International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City, 1–3 November 2018. 2018. [Online]. Available: <https://doi.org/10.1109/kse.2018.8573337>.
- [6] Д. Ткачик, О. Захарчук, "Розпізнавання емоційного забарвлення тексту для фундаментального аналізу на фінансових ринках," *Збірник наукових праць SCIENTIA, Валенсія, Королівство Іспанія: Європейська наукова платформа*. 2021.
- [7] Ł. Augustyniak et al., "Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis," *Entropy*, vol. 18, no. 1, p. 4. 2015. [Online]. Available: <https://doi.org/10.3390/e18010004>.
- [8] B. Liang et al., "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowledge-Based Systems*, vol. 235, p. 107643. 2022. [Online]. Available: <https://doi.org/10.1016/j.knosys.2021.107643>.
- [9] Д. А. Ткачик, Р. Н. Кветний, "Розробка ефективних комбінацій моделей технічного аналізу для прогнозування ринку," *Матеріали XLIX науково-технічної конференції підрозділів ВНТУ, Вінниця, 27-28 квітня 2020 р.* 2020. [Електронний ресурс]. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2020/paper/view/9600>.

Стаття надійшла: 18.09.2022.

References

- [1] F. Hamborg, K. Donnay, "NewsMTSC: A Dataset for (Multi-)Target-dependent Sentiment Classification in Political News Articles," Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Stroudsburg, PA, USA, 2021. [Online]. Available: <https://doi.org/10.18653/v1/2021.eacl-main.142>.
- [2] What is polarity and subjectivity in sentiment analysis? [Online]. Available: <https://www.quora.com/What-is-polarity-and-subjectivity-in-sentiment-analysis>.
- [3] Sentimental analysis using vader. [Online]. Available: <https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>.
- [4] Santur Y. Sentiment Analysis Based on Gated Recurrent Unit. 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 21–22 September 2019. 2019. [Online]. Available: <https://doi.org/10.1109/idap.2019.8875985>.
- [5] K. V. Nguyen et al., "UIT-VSFC: Vietnamese Students' Feedback Corpus for Sentiment Analysis," 10th International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City, 1–3 November 2018. 2018. [Online]. Available: <https://doi.org/10.1109/kse.2018.8573337>.
- [6] Tkachyk, O. Zakharchuk, "Rozpiznavannia emotsiinoho zabarvlennia tekstu dlia fundamentalnogo analizu na finansovykh rynkakh," *Zbirnyk naukovykh prats SCIENTIA, Valensiia, Korolivstvo Ispaniia: Yevropeiska naukova platforma*. 2021 [in Ukrainian].
- [7] Ł. Augustyniak et al., "Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis," *Entropy*, vol. 18, no. 1, p. 4. 2015. [Online]. Available: <https://doi.org/10.3390/e18010004>.
- [8] B. Liang et al., "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," *Knowledge-Based Systems*, vol. 235, p. 107643. 2022. [Online]. Available: <https://doi.org/10.1016/j.knosys.2021.107643>.
- [1] D. A. Tkachyk, R. N. Kvietyi, "Rozrobka efektyvnykh kombinatsii modelei tekhnichnogo analizu dlia prohnouzuvannia rynku," *Materialy XLIX nauково-tekhnichnoi konferentsii pidrozdiliv VNTU*,

Vinnytsia, 27-28 kvitnia 2020 r. 2020. [Online]. Available:
<https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2020/paper/view/9600> [in Ukrainian].

Відомості про авторів

Ткачик Денис Анатолійович – аспірант кафедри АІТ, факультет інтелектуальних інформаційних технологій та автоматизації.

Квєтний Роман Наумович – доктор технічних наук, професор.

D. A. Tkachyk, R. N. Kvyetnyy

**SENTIMENT ANALYSIS OF THE TEXT FOR FORECASTING
DATA IN FINANCIAL MARKETS**

Vinnytsia National Technical University, Vinnytsia

КОМП'ЮТЕРНІ СИСТЕМИ ТА КОМПОНЕНТИ

УДК 004.27

Н. А. Христинець

РЕАЛІЗАЦІЯ БАГАТОПОТОЧНОСТІ НА АРХІТЕКТУРІ
МУЛЬТИМЕДІЙНИХ ПРОЦЕСОРІВ NEXPERIA

Луцький національний технічний університет, Луцьк

Анотація. Розглянуто структуру мультипроцесорних систем на архітектурі сучасних мультимедійних процесорів Nexperia з 32-розрядним обчислювальним ядром. Досліджено сфери застосування мультимедійних процесорів, як процесорів загального призначення та їх функції обробки даних. Мультимедійні процесори використовують або функціональні архітектури з обмеженою гнучкістю, але вищою швидкістю та ефективністю, або програмовані архітектури з підвищеною гнучкістю. Проаналізовано архітектуру одного з процесорів Nexperia – TriMedia TM-1300 та наведено схему його основних компонент, принцип роботи центрального арбітражу шини процесора та способи нарощення його швидкодії. Виявлено, що удосконалені процесори загального призначення забезпечують підтримку мультимедіа шляхом включення нових мультимедійних інструкцій і їх паралельного виконання за допомогою підходу співпроцесора SIMD. Вони забезпечують підтримку мультимедіа, включаючи мультимедіа інструкції в набір інструкцій. Замість виконання певних мультимедійних функцій (наприклад стиснення та 3D графіки), мультимедійні процесори надають спеціально створені інструкції для підтримки загальних операцій у обробці відео. Ці інструкції включають підтримку 8-бітних типів даних (пікселів), ефективну адресацію даних і інструкції вводу/виводу. У статті розглянуто можливості програмної реалізації розпаралелення роботи процесорів за допомогою технологій паралельної обробки, яка досягається способом поділу одного виконання завдання на декілька незалежних менших завдань. Запропоновано програмну реалізацію роботи статичним та анонімним методом. Наведені коди програм та результати їх тестування. Доведено, що розбиття різних частин завдання між кількома обчислювальними ресурсами CPU дає змогу скоротити час виконання програми та вдосконалюють потенційну обчислювальну потужність роботи комп'ютерної системи.

Ключові слова: архітектура комп'ютерів, багатопроцесорні системи, мультимедійний процесор, паралельні обчислення, Nexperia, TriMedia.

Abstract. The structure of multiprocessor systems based on the architecture of modern Nexperia multimedia processors with a 32-bit computing core is considered. The fields of application of multimedia processors as general purpose processors and their data processing functions are studied. Multimedia processors use either functional architectures with limited flexibility but higher speed and efficiency, or programmable architectures with increased flexibility. The architecture of one of the Nexperia processors - TriMedia TM-1300 is analyzed and the diagram of its main components, the principle of operation of the central arbitration of the processor bus and methods of increasing its speed are given. Advanced general-purpose processors have been found to provide multimedia support by incorporating new multimedia instructions and executing them in parallel using a SIMD coprocessor approach. They provide multimedia support, including multimedia instructions in the instruction set. Instead of performing specific multimedia functions (such as compression and 3D graphics), multimedia processors provide purpose-built instructions to support general video processing operations. These instructions include support for 8-bit data types (pixels), efficient data addressing, and I/O instructions. The article examines the possibilities of software implementation of the parallelization of processors using parallel processing technologies, which is achieved by dividing one execution of a task into several independent smaller tasks. A software implementation of the work using a static and anonymous method is proposed. The program codes and the results of their testing are given. It has been proven that the division of different parts of the task between several computational resources of the CPU allows to reduce the execution time of the program and improves the potential computing power of the computer system.

Keywords: computer architecture, multiprocessor systems, multimedia processor, parallel computing, Nexperia, TriMedia.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-59-64>.

Вступ

Голландська компанія Nexperia, яка є брендом Philips, щорічно постачає понад 100 мільярдів товарів мікропроцесорної техніки у світі. Ці продукти визнані еталонами ефективності мікропроцесорів завдяки чудовому поєднанню їх розміру, потужності та продуктивності. Невеликий розмір процесора сприяє економії цінної енергії та простору на платі. Крім того, Nexperia – це мікропроцесорна техніка для унікальної групи чипів, які спрощують розробку мультимедійних пристроїв кожного нового покоління. Гнучкі рішення Nexperia допомагають виробникам задовольняти потреби ринку в інноваційних, привабливих продуктах на цільових споживчих і комунікаційних ринках, починаючи від високо-інтегрованих програмованих систем SoC і супутніх мікросхем до них, а також широкого кола системного програмного забезпечення.

Актуальність

Термін «потоків передача» у багатопроцесорній обробці зазвичай використовується для опису безперервних нескінченних потоків даних без початку чи кінця, які забезпечують постійну подачу цих даних. Таку подачу можна використовувати без необхідності попереднього їх завантаження. Потoki даних генеруються усіма типами джерел у різних форматах і обсягах: від програм, мережних пристроїв і файлів журналу сервера до активності на веб-сайті, банківських транзакцій і даних про місцезнаходження. Усі вони можуть бути агреговані для безперешкодного збору інформації в реальному часі та подальшої аналітики. Тому, програмна реалізація потокової передачі даних є **актуальною** задачею у сучасних багато потокових мультипроцесорних системах.

Мета

В основі процесорів Nexperia лежить архітектура векторного процесора обробки сигналів з довгим словом (VLIW). Архітектура процесорів з кількома обчислювальними пристроями VLIW характеризується тим, що одна інструкція процесора містить кілька операцій, які мають виконуватися паралельно. На час започаткування цієї архітектури стало зрозуміло, що процесорам загального призначення необхідна дуже велика обчислювальна потужність обробки сигналів у реальному масштабі часу. Такі прилади виходили дорогими, їм були потрібні спеціалізовані мікросхеми для підтримки операцій введення-виведення, і вони споживали багато енергії. Тому, **метою** розробки універсального та економного медіа-процесора стало прискорення роботи мультимедійних додатків як в автономному режимі, так і в обчислювальних системах із мікропроцесорами загального призначення. Багатопроцесорність передбачає як локальний, так і віддалений паралелізм, ефективно обходячи глобальне блокування інтерпретатора, використовуючи підпроцеси замість потоків. Завдяки цьому багатопроцесорна обробка дозволяє програмісту повністю використовувати кілька процесорів на одній машині. Такий підхід успішно реалізується як на Unix, так і на Windows.

Задачі на дослідження

Розглянемо архітектуру одного із процесорів сімейства Nexperia – TriMedia, TM-1300, який має 32-розрядне обчислювальне ядро VLIW/SIMD з вбудованою кеш-пам'яттю команд (32 Кбайт) та даних (16 Кбайт). Він працює з тактовою частотою до 166 МГц і має векторний співпроцесор, що реалізує алгоритм Хоффмана (для обробки MPEG-2). Також у його складі є співпроцесор зображень, що призначений для перетворення кольорового простору (YUV/RGB, 32-розрядний контролер зовнішньої пам'яті (до 64 Мбіт) SDRAM) та набір таймерів. Усі вбудовані периферійні пристрої та співпроцесори працюють незалежно від центрального процесора (ЦП) під керуванням DMA5. Інтерфейс PCI/XIO забезпечує зв'язок TM-1300 із шинами персональних комп'ютерів, а також стандартною мікропроцесорною периферією. Інші цифрові інтерфейси процесора: відеовхід та відеовихід ITU-656, звукові вхід (2 канали) та вихід (до 8 каналів) I2 S, звуковий вихід IEC958 (S/PDIF), керуючий порт I2 C, налагоджувальний порт JTAG, а також синхронний послідовний порт для підключення аналогових та цифрових модемів.

На базі приладів родини TriMedia з зазначеною архітектурою було розроблено високоякісні вироби побутового та промислового призначення. Процесор TM-1300 став основою для приладу PNX1300 нової архітектури Nexperia. Основними відмінностями PNX1300 стали знижені напруга живлення та енергоспоживання, збільшені тактові частоти ядра та основного ОЗП (до 200 і 183 МГц відповідно), підтримка 16- та 32-розрядного інтерфейсу SDRAM обсягом до 256 Мбіт (16 Мбіт). Також було спрощено порядок подачі напруги живлення, виправлено помилки в роботі інтерфейсу PCI та програми-завантажувача. Внаслідок цього, конкурентоспроможність приладів підвищилася. Програми, розроблені для процесорів TriMedia, Nexperia необхідно перекомпілювати, хоча сумісність на рівні вихідних кодів і бібліотек API гарантована.

Розв'язання задач

Головною ідеєю, реалізованою в сімействі процесорів Nexperia, є гнучка обчислювальна система, що має самостійну периферію і програмно адаптується до виконуваного додатку. Мета впровадження нової архітектури – максимально збільшити швидкістю обчислень. Крім того, чим точніше обчислювальний пристрій може бути налаштований на клас додатків, тим більше його ефективність (прив'язано з мікропроцесорами ЦПОС загального призначення). Система, реалізована на основі Nexperia, працює під управлінням компактного ядра операційної системи реального часу (pSOS), що виконується центральним процесором VLIW. Ключові моменти в роботі даної архітектури – поділ у часі роботи ЦП та периферії та «спілкування» всіх пристроїв через ОЗП за допомогою шини та механізму DMA.

Центральний процесор TM32 (рис. 1) перемикається від одного завдання до іншого наступним чином. Спочатку він декодує відеокادر, потім кілька пакетів аудіопотоку, повертається до відеоданих, керує портом PCI, залучає графічний співпроцесор для ресурсомісткої операції масштабування зображення тощо. За кожний такт ЦП може задіяти до 5 пристроїв одночасно, включаючи периферію та 27 конвеєризованих функціональних блоків.

Для успішного виконання функцій «диригента» центральний процесор звільнений від деяких операцій, характерних для мікропроцесорів загального призначення, наприклад, трансляції адрес або обчислень із подвійною точністю; з нього виключено планувальник команд і суперскалярну логіку та тим самим знижено апаратні витрати. Проте, набір команд TM32 включає усі стандартні операції. Внутрішня шина процесорів Nexperia також забезпечує доступ до регістрів стану та управління всіх пристроїв та зовнішньої периферії. Вона складається з двох окремих 32-розрядних шин адреси та даних. Для обміну використовується протокол групового пересилання.

Доступом до шини керує центральний арбітр, до якого підведено лінії запитів від кожного пристрою, що має можливість захоплення шини (bus master). Алгоритми роботи арбітра шини можуть залежати від виконуваної програми, як і смуга пропускання шини, яка виділяється пристроям. Кожен режим

роботи центрального арбітра гарантує мінімальну смугу пропускання та максимальний час очікування конкретного пристрою.

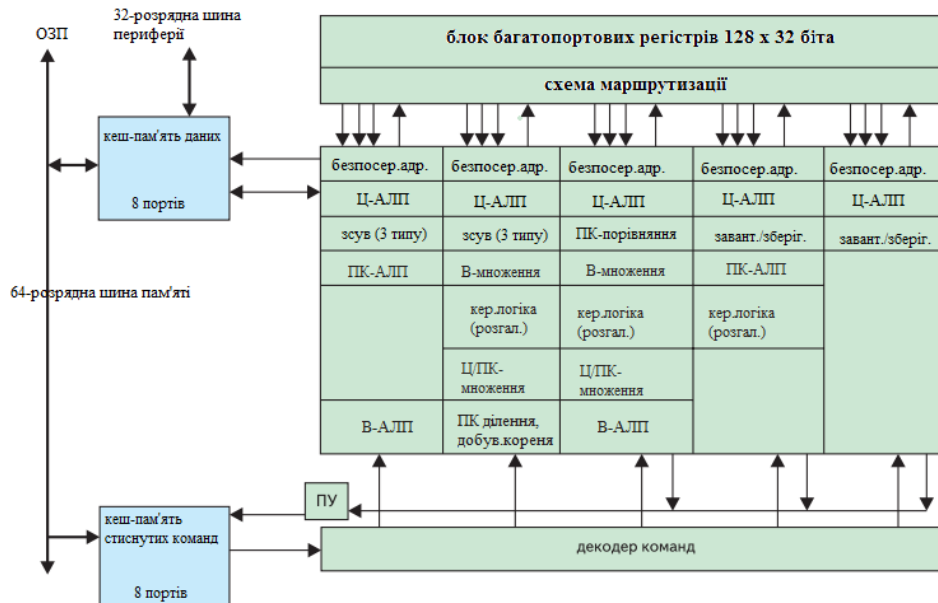


Рисунок 1 – Спрощена блок схема центрального процесора TriMedia (TM32). Умовні позначення: Ц – цілочисельний, ПК – плаваюча кома, В – векторний, ПУ – програмне управління

Описаний принцип роботи є однією з «родзинок» процесорів TriMedia/Nexperia, що забезпечують прискорення роботи мультимедійних програм. Для порівняння, перехресний комутатор (crossbar switch), що використовується у VLIW-процесорах компанії Texas Instruments, не підтримує гнучку систему пріоритетів пристроїв, його простий механізм вирішення конфліктів при запиті доступу до шини працює за жорстко обумовленою схемою. В цьому випадку необхідна швидкодія системи «добирається» необхідною кількістю АЛП, розвиненим кешуванням, потужністю співпроцесорів та високою тактовою частотою (до 1 ГГц).

Гнучкість процесорів Nexperia, їх унікальна здатність тонкого налаштування на виконуваний додаток, пред'являють особливі вимоги до інструментального програмного забезпечення. У структурі вищезгаданого процесора, планувальник команд інтегрований із розподільником регістрів, він підтримує захищене виконання команд, конвеєрну організацію та керування функціональними блоками ЦП та системи в цілому. Програмна реалізація багатопоточності дозволяє наочно продемонструвати ефективність розпаралелювання на рівні команд та зниження кількості звернень до регістрів та ОЗП.

Реалізуємо програмно кілька методів паралельного використання потоків через пасивне очікування, яке реалізується за допомогою операційної системи. Операційна система зберігає контекст потоку та вивантажує його, надаючи можливість виконуватись іншим потокам. Досліджено, що завантаження процесора при активному очікуванні в середньому дорівнює 92%. При пасивному очікуванні основний потік вивантажується і зайнятість ЦП в середньому дорівнює 50%.

Статичний метод дозволяє обробляти паралельно довільну кількість потоків незалежно один від одного (тут є суттєва різниця з статичним класом, доступ до якого необхідно організувати потокобезпечним). У наведеному фрагменті паралельно виконуються первинний та вторинний (дочірній) потоки:

// Статичний метод, який планується виконувати одночасно в головному (первинному) та у вторинних потоках.

```
static void WriteSecond()
{
    // CLR призначає кожному потоку свій стек із власними локальними змінними.
    // Окремий екземпляр змінної counter створюється в стеці кожного потоку,
    // тому для кожного потоку виводяться свої значення counter
    int counter = 0;

    while (counter < 10)
```

```

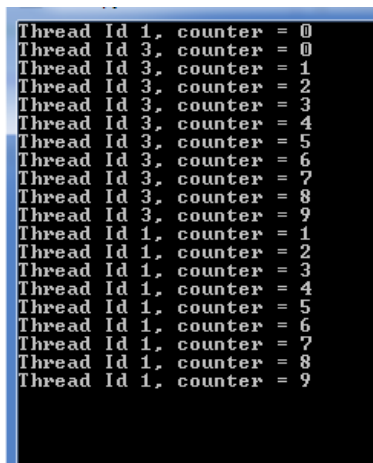
    {
        Console.WriteLine("Thread Id {0}, counter = {1}", Thread.CurrentThread.GetHashCode(), counter);
        counter++;
    }
}

static void Main()
{
    // Робота вторинного потоку.
    Thread thread = new Thread(WriteSecond);
    thread.Start();

    // Робота первинного потоку.
    WriteSecond();
}

```

Лістинг ефекту операцій за даним методом наведено на рисунку 2.



```

Thread Id 3, counter = 0
Thread Id 3, counter = 1
Thread Id 3, counter = 2
Thread Id 3, counter = 3
Thread Id 3, counter = 4
Thread Id 3, counter = 5
Thread Id 3, counter = 6
Thread Id 3, counter = 7
Thread Id 3, counter = 8
Thread Id 3, counter = 9
Thread Id 1, counter = 0
Thread Id 1, counter = 1
Thread Id 1, counter = 2
Thread Id 1, counter = 3
Thread Id 1, counter = 4
Thread Id 1, counter = 5
Thread Id 1, counter = 6
Thread Id 1, counter = 7
Thread Id 1, counter = 8
Thread Id 1, counter = 9

```

Рисунок 2 – Лістинг ефекту операцій за статичним методом

Наступний, *анонімний метод* створює безіменний блок коду, пов'язаний з конкретним екземпляром делегата. Для створення анонімного методу зазначено кодовий блок після ключового слова `delegate`. У наведеному прикладі анонімний метод служить виведенням повідомлень «№ counter = 1» від 1 до 3. Спочатку у програмі оголошується тип делегата `CountIt` без параметрів і з типом `void`. Далі у методі `Main()` створюється екземпляр `count` делегата, якому передається кодовий блок, наступний після ключового слова `delegate`. Саме цей кодовий блок і є анонімним методом, який виконуватиметься при зверненні до делегата `count`. В програмі використано скорочений синтаксис анонімного методу. Потік має властивість `IsAlive`, що повертає `true` після виклику `Start()` і до завершення потоку. Потік, який закінчив виконання, не може бути розпочато знову:

// Потоки. Анонімні методи.

```

namespace Threads
{
    class Program
    {
        static void Main()
        {
            int counter = 0;

            // Початок потоку
            Thread thread = new Thread(delegate() { Console.WriteLine("1. counter = {0}", ++counter); });
            thread.Start();

            Thread.Sleep(100);
            Console.WriteLine("2. counter = {0}", counter);
        }
    }
}

```

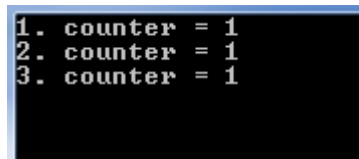
```

// Параметри
thread = new Thread((object argument) => { Console.WriteLine("3. counter = {0}", (int)argument); });
thread.Start(counter);

// Затримка
Console.ReadKey();
}
}
}

```

Результати тестування даного методу наведені на рисунку 3.



```

1. counter = 1
2. counter = 1
3. counter = 1

```

Рисунок 3 – Результати тестування анонімного методу

Наведені методи демонструють паралельне виконання коду через багатопоточність. Використання кількох потоків виконання – це один із способів забезпечити можливість реагування програми на дії користувача при одночасному використанні процесора для виконання завдань між появою або навіть під час появи подій користувача.

Висновки

Існує безліч засобів для оцінки продуктивності комп'ютерних систем. Деякі тести для такої оцінки вимірюють час виконання часто уживаних алгоритмів, інші оцінюють виконання завдань, специфічних для конкретної прикладної області, треті формують власний синтетичний показник, заснований на вимірі продуктивності різних комп'ютерних підсистем. Сучасні архітектури багатопроцесорних обчислювальних систем настільки різноманітні і складні, що при проектуванні нових алгоритмів необхідно враховувати значну кількість факторів. Ігнорування приватних архітектурних особливостей, закладених розробниками у обчислювальну систему, може призвести до значного падіння продуктивності та ефективності прикладної програми. Розглянуто теоретичну сутність мультипроцесорів Nexperia та виявлено, що мультипроцесорна архітектура забезпечує суттєве підвищення надійності системи, можливості розпаралелення інформаційних потоків. Програмні реалізації методів паралельних обчислень передбачають виконання їх на системі з двома процесорами, один з яких може працювати в режимі ядра, а інший – в режимі користувача. В подальшому ці алгоритми можна буде допрацювати, реалізувати на іншій мові програмування та застосувати для ряду інших прикладних задач галузі інформаційних технологій.

Список літератури

- [1] L. Wanga, "Research on the Performance of Robot Multiprocessor Control System Based on BS Structure Digital Media", *Microprocessors and Microsystems*. 2020. [Електронний ресурс]. Режим доступу: <https://www.sciencedirect.com/science/article/abs/pii/S0141933120300910#>. Дата звернення: 20 серпня 2022.
- [2] В. О. Денисюк, С. М. Цирульник, *Мікропроцесорні системи управління: навч. посіб.* Вінниця, Вінн. нац. аграр. ун-т: ТВОРИ, 2021, 204 с.
- [3] V. Padmajothi, J. MazherIqbal, V. Ponnusamy "Load - aware intelligent multiprocessor scheduler for time-critical cyber-physical system applications", *Computers & Electrical Engineering*. 2022. [Електронний ресурс]. Режим доступу: <https://www.sciencedirect.com/science/article/pii/S0045790621005462#>. Дата звернення: 20 серпня 2022.
- [4] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski and C. Kozyrakis, "Evaluating MapReduce for Multi-core and Multiprocessor Systems", 2007 IEEE 13th International Symposium on High Performance Computer Architecture, 2007, pp. 13-24.
- [5] Сайт компанії «Nexperia». [Електронний ресурс]. Режим доступу: <https://www.nexperia.com>. Дата звернення: 20 серпня 2022.

Стаття надійшла: 10.09.2022.

References

- [1] L. Wanga, "Research on the Performance of Robot Multiprocessor Control System Based on BS Structure Digital Media", *Microprocessors and Microsystems*. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0141933120300910#>. Accessed on: August 20, 2022.

- [2] V. O. Denisyuk, S. M. Tsirulnyk, *Microprocessor control systems: academic. Manual*. Vinnytsia, Vinn. national agrarian university: CREATIONS. 2021, 204 p.
- [3] V. Padmajothi, J. MazherIqbal, V. Ponnusamy "Load - aware intelligent multiprocessor scheduler for time-critical cyber-physical system applications", *Computers & Electrical Engineering*. 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790621005462#>. Accessed on: August 20, 2022.
- [4] C. Ranger, R. Raghuraman, A. Penmetsa, G. Bradski and C. Kozyrakis, "Evaluating MapReduce for Multi-core and Multiprocessor Systems", 2007 IEEE 13th International Symposium on High Performance Computer Architecture, 2007, pp. 13-24.
- [5] Nexperia website. [Online]. Available: <https://www.nexperia.com/>. Accessed on: August 20, 2022.

Відомості про авторів

Христинець Наталія Анатоліївна – кандидат технічних наук, старший викладач кафедри комп'ютерної інженерії та кібербезпеки.

N. A. Khrystynets

IMPLEMENTATION OF MULTI-THREADING ON THE ARCHITECTURE OF NEXPERIA MULTIMEDIA PROCESSORS

Lutsk National Technical University, Lutsk

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОБЧИСЛЮВАЛЬНІ МЕТОДИ

УДК 519.876.5

Д. І. Гришин, Т. М. Боровська

РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ ОПТИМАЛЬНОГО ВИРОБНИЦТВА ТА РОЗВИТКУ ВИРОБНИЧИХ СИСТЕМ З ДЕКОМПОЗИЦІЄЮ ПЛАНОВОГО ПРОЦЕСУ РОЗВИТКУ НА ІНТЕРВАЛИ

Вінницький національний технічний університет, Вінниця

Анотація. Розглянуто проблему побудови ефективних моделей оптимального розвитку та функціонування сучасних виробничих систем, що функціонують у активному оточенні. Після аналізу аналогів було виявлено, що для виробництв класу "виробництво-розвиток", які функціонують у активному оточенні – конкурентів, посередників, постачальників та споживачів – адекватні моделі відсутні. Базовий аналог – рішення варіаційної задачі розвитку і виробництва має обмежену область адекватності – це виробництва зі статичним зовнішнім оточенням. В розглянутому нами аналізі оптимальна стратегія розвитку створюється на весь плановий період, що зазвичай становить 2-10 років. На такий довгостроковий період неможливо передбачити стан ринків продукції, фінансів, технологій. Зазначене вище обумовлює актуальність даної роботи. В даній роботі використовується узагальнена модель оптимального розвитку на базі методології оптимального агрегування. Використання методології оптимального агрегування дозволяє перейти від багатовимірної задачі нелінійного програмування до системи одновимірних задач оптимізації. Обчислювальна складність при цьому зростає лінійно, що дозволяє використати цю методологію для виробничих систем з великою кількістю та нелінійністю зв'язків між елементами. В роботі виконується модифікація базової моделі оптимального розвитку з розбиттям процесу розвитку на інтервали. На початку кожного інтервалу оптимальна стратегія розвитку коригується з урахуванням уточнення інформації про майбутній стан активного середовища: дії конкурентів, споживачів, постачальників, посередників, світових ринків. Для визначення оптимального значення та оптимального розподілу ресурсів між підсистемами на кожному інтервалі визначаються максимуми критерію – параметризованої функції ефективності системи. Наведено приклади моделювання та тестування моделей.

Ключові слова: оптимальний розвиток, виробнича система, імітаційна модель, оптимальне управління, оптимальне агрегування, декомпозиція.

Abstract. The problem of developing effective models of optimal development and functioning of modern production systems functioning in an active environment is considered. Analyzing analogues showed that there are no adequate models for "production-development" industries that operate in an active environment – competitors, intermediaries, suppliers and consumers. The basic analogue is the solution to the variation problem of development and production has a limited area of adequacy – productions with a static environment. In the basic analogue, the optimal development strategy is created for the entire planning period, which is usually 2-10 years. For such a long-term period, it is impossible to predict the state of product, finance, and technology markets. The above confirms that this work is relevant. This work uses a generalized model of optimal development based on the methodology of optimal aggregation. Using the methodology of optimal aggregation allows us to move from a multidimensional problem of nonlinear programming to a system of one-dimensional optimization problems. At the same time, the computational complexity increases linearly, which allows us to use this methodology for production systems with a large number and nonlinearity of connections between elements. The work modifies the basic model of optimal development with the division of the development process into intervals. At the beginning of each interval, the optimal development strategy is adjusted taking into account the clarification of information about the future state of the active environment: the actions of competitors, consumers, suppliers, intermediaries, world markets. To determine the optimal value and the optimal distribution of resources between subsystems, the maxima of the – criterion the parameterized function of the system efficiency – are determined at each interval. Examples of modeling and testing models are given.

Keywords: optimal development, production systems, simulation model, optimal control, optimal aggregation, decomposition.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-65-73>.

Вступ

Глобалізація, швидкий темп науково-технічного прогресу та швидкий темп зміни моделей виробів, призвели до суттєвих структурних змін виробництва, як об'єкта управління [10]. Сучасні виробництва в більшості є інформаційно та ресурсно інтегрованими, вони характеризуються високою розмірністю, динамічністю і суттєвою нелінійністю характеристик, а технологічні процеси мають велику кількість параметрів та змінних. Це створює нові виклики у теорії і практиці управління виробничими системами.

Класичні методи оптимізації такі, як лінійне, випукле, цілочислове програмування, не можуть використовуватись при моделюванні технологічних процесів, через недостатність апріорної інформації про закономірності протікання процесів та складні нелінійні зв'язки між змінними. Метод динамічного програмування Р. Беллмана не знімає проблему високої розмірності, оскільки виконує декомпозицію процесу «в часі», а не «в просторі». Що робить метод непридатним для оптимізації великих індустріальних систем [2]. Статистичні методи стикаються з проблемою високої динамічності виробничих систем. Вінер писав, що його статистичні моделі і методи непридатні для індустріальних систем – там «занадто короткі статистичні ряди». А Форрестер зазначав, що лінійні методи статистики є непридатними для аналізу і прогнозування процесів виробничих систем через суттєву нелінійність цих процесів. Це стало причиною вибору безпошукових методів оптимального агрегування, які знімають проблему високої розмірності, та ряд математичних обмежень на вид вхідних функцій.

Інша проблема сучасних виробничих систем, це швидкоплинність і висока активність зовнішнього оточення. В зв'язку з чим виникає необхідність розробки нового підходу, який буде враховувати зміни стану оточення, учасників ринку, технологій, фінансів, тощо.

Мета

Метою роботи є розробка ефективних моделей процесів розвитку та виробництва для сучасних інтегрованих виробничих систем, що функціонують у активному оточенні, з застосування декомпозиції процесу розвитку на інтервали.

Задачі роботи – розробка параметризованого оператора переходу між інтервалами процесу розвитку; рішення задачі оптимального розвитку для структури «виробництво-розвиток» з використанням нового оператора; розробка та тестування робочих моделей в програмному пакеті Mathcad.

Об'єкт дослідження

Розглянемо предметно об'єкт дослідження. В статті розглядається інтегрована система «виробництво-розвиток». Змістовно це певне виробництво – цех, фабрика, що випускає високотехнологічну продукцію в умовах досить високих темпів змін технологій і продуктів виробництва. Саме в таких умовах зручно у виробничих приміщеннях або поряд розташувати лабораторії, ділянки роботи з обладнанням, дослідні і контрольні лабораторії, власні підрозділи монтажу і запуску нового обладнання. Таку систему вигідно об'єднати організаційно і фінансово. Природно виникає задача оптимального розподілу ресурсів між підсистемами «виробництво» і «розвиток» і зменшує втрати при відмовах обладнання і покупців.

Очевидно, що підсистема «розвиток» підвищує ефективність і виробничу потужність основного виробництва в математичних моделях. Це формулюється так: результат витрат ресурсів в підсистемі розвиток – зміна (покращення) виробничої функції основного виробництва. Тобто маємо неоднорідну структуру з параметричним зв'язком.

Базовий сценарій для моделі функціонування і розвитку: на вхід працюючої системи приходять «квант ресурсу». Його треба оптимально розподілити. Альтернативи розподілу знаходяться в інтервалі «все в розвиток» – «все у виробництво». Критерії оптимальності розподілу: прирощення випуску продукції, зважена сума прирощень випуску та виробничої потужності.

На рисунку 1 подана схема інтегрованої системи «виробництво-розвиток» [3]. На схемі зображені ресурсні та інформаційні зв'язки системи. На вхід системи подається ресурс, що може складатися з власного ресурсу (реалізація виробленого) та зовнішнього (кредити). Ресурс оптимально розподіляється між підсистемами виробництва та розвитку. Критерій оптимальності може відрізнитись в залежності від виробництва, в даній моделі критерій – максимальний випуск. Екстремальний регулятор розподіляє вхідний ресурс між виробництвом і розвитком, регулятор виробництва забезпечує необхідний темп випуску, регулятор розвитку забезпечує необхідний темп виконання процедур розвитку.

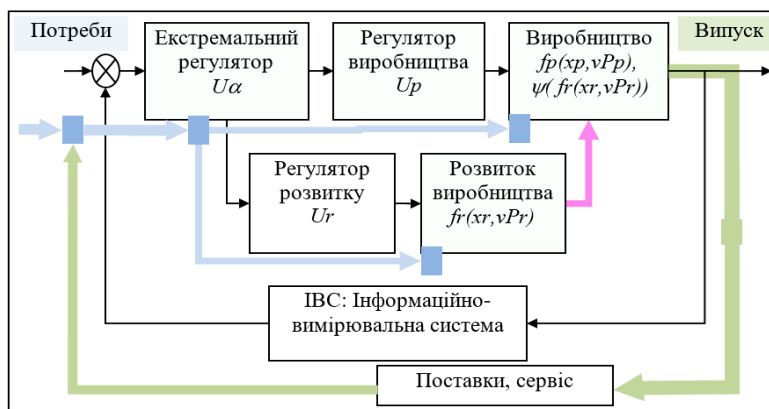


Рисунок 1 – Схема інтегрованої системи «виробництво-розвиток»

Блок «ІВС» виступає у якості зворотного зв'язку системи, відповідає за збір і обробку даних про стан системи. Блок «постачання, сервіс» (логістика) забезпечує потреби в продукції і так отримує кошти або бартер для себе, в американській практиці це називається «справедливий розподіл цінності продукту між виробником і користувачем», а «ціни, гроші, економіка і фінанси вважають головним джерелом криз» [1].

Сучасні індустріальні системи є інформаційно і ресурсно інтегрованими. Часто виконується і територіальна інтеграція в «мегазаводи», на яких інтегруються виробництво, розвиток, інновації, логістика, сервіста інші структурні одиниці для забезпечення випуску динамічної лінійки продуктів. Новіша тенденція у виробництві – подрібнення і територіальне розсіювання виробництва кінцевого продукту.

Соціальні та екологічні переваги нових технологій і виробництв – забезпечення якісними робочими місцями на малих територіальних утвореннях, екологізація виробництв. Такі структурні зміни виробничих систем вимагає зміни в методології і технології створення математичних моделей виробничих систем (ВС). Виникає потреба відображення складної інтегрованої структури виробництва у нових моделях розвитку та виробництва [3, 4].

По зазначеним причинам, в якості теоретичної основи статті була обрана методологія оптимального агрегування.

Методи дослідження

Методологія оптимального агрегування виробничих систем базується на інформаційній технології побудови «робочих моделей» – математичних моделей, реалізованих в середовищах математичних пакетів. Суть методології полягає в заміні багатовимірної задачі нелінійного програмування системою одновимірних задач оптимізації. Обчислювальна складність при цьому зростає приблизно лінійно в залежності від розмірності задачі. Однак, головною перевагою методології оптимального агрегування є рішення задач, для яких невідомі навіть постановки.

Конкретизуємо використану термінологію:

- метод оптимального агрегування – відображення структури виробництва та його ресурсних зв'язків в структуру бінарного дерева оптимального агрегування (БДОА);
- бінарний оператор оптимального агрегування – відображення зв'язку між парою елементів виробничої системи;
- бінарне дерево оптимального агрегування – це ресурсна структура виробничої системи представлена у вигляді бінарного дерева, яке потім агрегується в оптимально еквівалентну функцію виробництва (ОЕФВ).

Авторами було запропоновано оператори для різних видів ресурсних структур [8] – паралельної, послідовної, структури з ресурсним зворотнім зв'язком, структури «виробництво, розвиток» та ін.

Створення оператора – це не тривіальна задача, що має певні загальні правила розробки оператора, та потребує кваліфікації в області проектування робочих моделей. Методи оптимального агрегування являються комплексами задач і для осмислення потребують цілісного сприйняття. Згідно правилам моделювання, спочатку розробляється математична модель, потім створюється «алгоритм» для цієї моделі, потім цей алгоритм виконується в певному програмному оточенні (платформі), після чого модель перевіряється на адекватність і власне проводиться моделювання. Сучасні засоби моделювання поєднують ці кроки – робоча модель розробляється одразу в програмному оточенні, в вигляді інтерфейсів, графіків, інтерактивних стендів, тощо.

На рисунку 2 подано приклад рішення задачі методом оптимального агрегування. Ліворуч подано порядок кроків рішення задачі. Праворуч – графічне представлення рішення задачі оптимального агрегування для системи з трьох елементів. Подано: формулу (бінарне дерево оптимального агрегування), результат обчислення за цією формулою – матриця відповідної структури і розмірності, оптимальна еквівалентна функція виробництва системи (ОЕФВ) та її складові – функції виробництва (ФВ) підсистем, та графіки оптимального розподілу ресурсу між цими підсистемами.

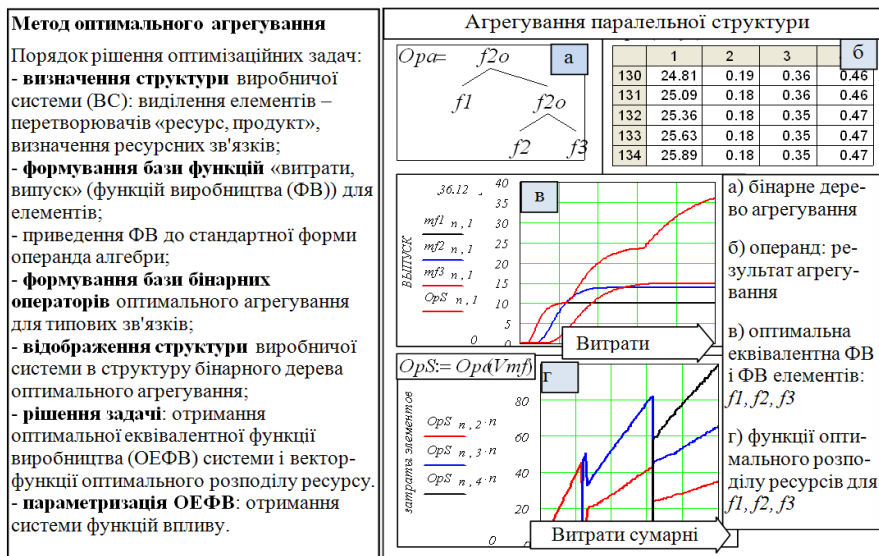


Рисунок 2 – Зразок рішення задачі методом оптимального агрегування

Аналіз прототипу

Проаналізуємо предметно базовий аналог – рішення задачі оптимального управління процесами виробництва та розвитку методом оптимального агрегування. Дана задача розв'язувалась в рамках попередніх досліджень авторів.

На рисунку 3 представлені структура і задачі базового аналога [8] – задачі оптимального розвитку. Першоджерелом базового аналогу є рішення варіаційної «задачі розподілу» Р. Беллманом [2], який досліджував цю задачу суто аналітичними методами, дав рішення в загальному виді для одновимірної задачі, двовимірної і шлях рішення для задач довільної розмірності. Однак обчислювальна складність рішень зростала комбінаторно із зростанням розмірності системи [9]. Р. Беллман сформулював головну задачу своїх досліджень як заміну задачі пошуку точки в багатовимірному просторі системою задач пошуку точки в фазових просторах меншої розмірності, бажано – одновимірних. В подальших дослідження Р. Беллман перейшов до дискретних динамічних моделей, запропонував концепцію динамічного програмування для рішення багатовимірних задач оптимізації.

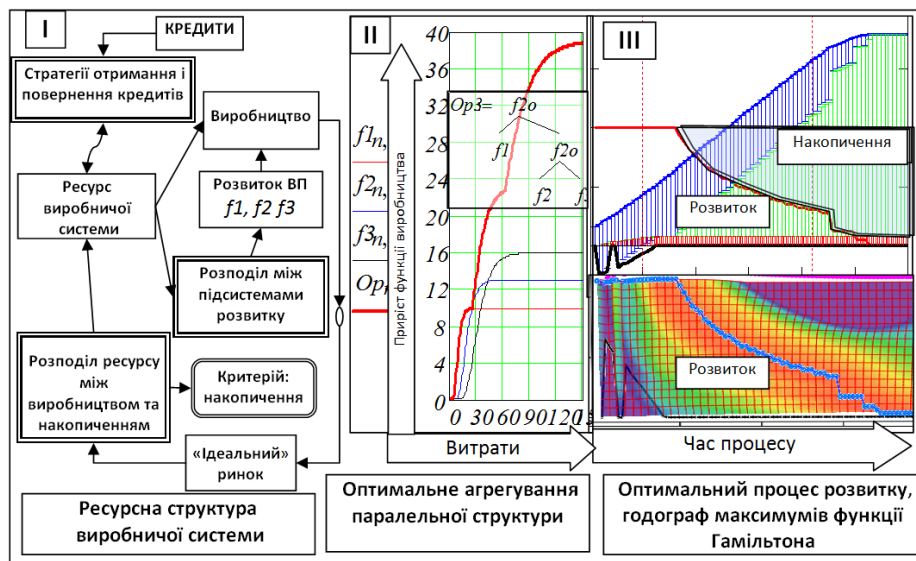


Рисунок 3 – Базовий аналог: ресурсна структура, оптимальне агрегування, оптимальна стратегія розвитку[8]

На рисунку 3 згруповано три компоненти аналогу, – три кроки рішення задачі розвитку на базі методів оптимального агрегування:

- формалізація і перевірка раціональності ресурсної структури виробництва;
- оптимальне агрегування, отримання оптимальних еквівалентних функцій виробництва і розвитку;
- рішення варіаційної задачі оптимального розвитку для еквівалентного одновимірного об'єкту.

Подивимось на частину I: ресурсна структура не елементарна; в частині II все очевидне – формула в структурному вигляді і графіки для складових формули: три функції виробництва замінені оптимальною еквівалентною функцією. Частина III – це вже динаміка, бачимо поточні значення сумарного виробництва і його розподіл: в розвиток, накопичення і повернення кредитів. В нижній частині – тривимірний графік функції Гамільтона, на якій подано годограф її максимумів. Цей годограф і є оптимальним розподілом ресурсу між розвитком і накопиченням. Це також приклад рішення варіаційної задачі розвитку методом принципу максимуму Понтрягіна [6].

Висновки з аналізу базового аналогу. Вибрана тема досліджень і розробки завжди була і залишається актуальною. Високі технології виробництва і глобалізація суттєво змінили властивості виробничих систем, від авіа індустрії до агровиробництва. Найважче наукове забезпечення виробництва не використовує в повній мірі переваги сучасних ІОС, ІУС, які, в свою чергу все далі відходять від сучасних методів моделювання і оптимізації при рішенні задач аналізу і синтезу виробничих систем.

Область адекватності моделі-аналога – виробництва зі статичним зовнішнім оточенням. На схемі це відображено у блоці «ідеальний ринок». В аналізі оптимальна стратегія розвитку створюється на весь плановий період (2-10 років). На такий період неможливо передбачити стан ринків продукції, фінансів, технологій. Тому виникає необхідність декомпозиції процесу розвитку на інтервали та корекції стратегії розвитку на початку кожного інтервалу. Для такої задачі прямих аналогів знайдено не було.

Постановка задачі оптимального агрегування з декомпозицією процесу розвитку на інтервали

Виробнича система (ВС) випускає декілька продуктів при необмежених потребах. Для кожного продукту відома функція розвитку (ФР) – залежність «витрати - приріст виробничих потужностей». Необмежені потреби існують на протязі «ринкового вікна», відповідно якому задається «плановий період» Tp . Ставиться мета так розподіляти поточні ресурси ВС між «накопиченням» та інвестиціями в розвиток виробництв продуктів, щоб максимізувати інтегральний критерій «сумарне накопичення».

Використаємо теорію оптимального агрегування, основне положення якої – можливість еквівалентної заміни виробничої системи оптимальним еквівалентним елементом з виробничою функцією. Рішенням задачі оптимального процесу розвитку і виробництва буде функція залежності оптимального розподілу ресурсу між виробництвом та розвитком від часу. Іншими словами це годограф максимумів поверхні Гамільтона.

Розглянемо оператор «функція Гамільтона». Носій – множина функцій $H(x(t))$, що складаються з спряжених функцій, які є результатом рішення системи нелінійних диференціальних рівнянь. Для реалістичних задач спряжені функції визначаються числовими методами в математичних пакетах.

$$Hka(x, \alpha, u) = xs \cdot (1 - \alpha) + \left(\sum_{j=1}^N fin(xs \cdot \alpha \cdot u_j) \right) \cdot (Tp - t) + zp(t) \cdot [1 + pr(Tp - t)] \quad (1)$$

$$Hka(x, \alpha) = xs \cdot (1 - \alpha) + fin(xs \cdot \alpha) \cdot (Tp - t) - zp(t) \cdot [1 + pr(Tp - t)] \quad (2)$$

$$fin(xs \cdot \alpha) = \begin{matrix} f2o \\ / \quad \backslash \\ flr \quad f2r \end{matrix} \quad (3)$$

Запишемо Гамільтоніан для оптимального агрегування елементів класу "виробництво-розвиток".

$$Hka(x, \alpha) = xs \cdot (1 - \alpha) + Fr(xs \cdot \alpha) \cdot (Tp - t) - zp(t) \cdot [1 + pr(Tp - t)] \quad (4)$$

$$Fr(xs \cdot \alpha) = \begin{matrix} f2o \\ / \quad \backslash \\ f2pr \quad f2pr \\ / \quad \backslash \quad / \quad \backslash \\ flp \quad flr \quad f2p \quad f2r \end{matrix} \quad (5)$$

де $fin(xs \cdot \alpha)$ – функція розвитку (ФР), що показує залежність прирощення параметрів функції виробництва від витрат, $xs(t)$ – сумарний обсяг ресурсу виробничої системи, $\alpha(t)$ – коефіцієнт розподілу ресурсів між підсистемами, тобто змінна управління, $zp(t)$ – функція що відображає використання зовнішніх ресурсів, Tp – «плановий період», pr – «вартість» зовнішніх ресурсів.

Таким чином, після багатьох складностей, ми отримали функцію Гамільтона як функцію двох аргументів – стану процесу та пропорції розподілу ресурсу між підсистемами розвитку та накопичення.

Таким чином, ми підходимо до аналогічної, як і при оптимальному агрегуванні, структури: годограф максимумів на множині альфа-функцій. Одна проекція – залежність максимуму функції Гамільтона від часу $maxH(x(t))$, інша проекція – оптимальна стратегія розвитку – залежність оптимальної пропорції від часу $op(x(t))$. Множина операцій – оператор переходу між станами.

При моделюванні процесу розвитку, поточна функція Гамільтона визначається рівнянням вигляду:

$$Hka(x_k, u_k) = x_k \cdot (1 - u_k) + fin(x_k \cdot u_k) \cdot (Tp - t) - ex(t) \cdot [1 + pr(Tp - t)] \quad (6)$$

Пропонується визначити оператор переходу між «станами функції Гамільтона»:

$$H^{(k)} = MH \left(H^{(k-1)} \right), H(x_k, u_k) = fH(x_{k-1}, u_{k-1}) \quad (7)$$

Це відкрита задача, для якої отримані рішення для простих моделей розвитку, але відсутні теоретичні дослідження. Саме в дослідженні і реалізації (7) полягає центральна задача даної статті.

На рисунку 4 подано схему нової, узагальненої моделі оптимального функціонування і розвитку. Ця схема виконана для тестової системи з чотирьох елементів. В класичних методах проблема розмірності

має таку закономірність: задача для системи з одного елементу – елементарна, для системи з двох елементів – легка, для системи з трьох і більше елементів – важка або можливо розв'язується штучними неймережами. В методології оптимального агрегування обчислювальні витрати залежить лінійно від розмірності системи і не створюються нові інтелектуальні проблеми [7]. Тому, якщо система з чотирьох елементів працездатна, то і система з 4000 елементів буде працездатною. Новизна і відсутність прямих аналогів обумовлюють необхідність вести розробку формалізованої (твердження, теореми, властивості) математичної моделі в комплексі з програмним модулем – тестовою робочою моделлю.

Схема на рисунку 4 складається з п'яти блоків:

- 1) структура системи;
- 2) оптимально агрегована система;
- 3) оптимальне агрегування інтервалу;
- 4) декомпозиція процесу розвитку;
- 5) оптимальне агрегування розвитку.

В даній статі розглядається задача декомпозиції процесу розвитку. Розробка оператора переходу між інтервалами процесу розвитку є новою розробкою на базі попередніх розробок авторів – рішення задачі розвитку методом оптимального агрегування.

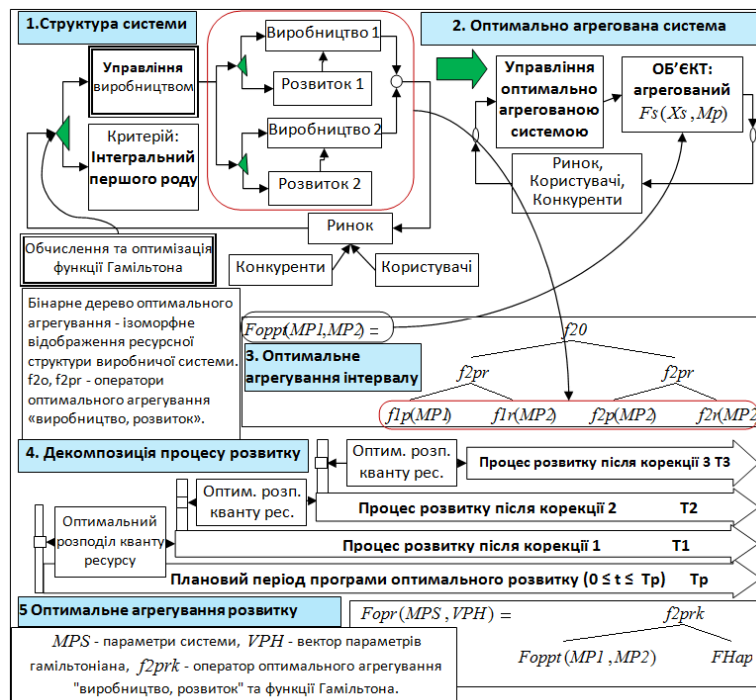


Рисунок 4 – Структура комплексу задач розробки. Нове рішення проблеми [8]

Проаналізуємо зв'язки між блоками: на базі аналізу структури системи отримуються всі функції виробництва і розвитку, виконується оптимальне агрегування параметризоване (блок 3). Результат оптимального агрегування – функція сумарного ресурсу і параметрів $F_s(X_s, M_p)$, яка використовується в блоці 2, який дозволяє оперувати з виробничою системою як з цілісним еквівалентним оптимальним об'єктом з ресурсним зворотним зв'язком «ринком».

Блоки 1, 2, 3 створюють необхідні дані для блоку 4 – «декомпозиція процесу розвитку», який є пунктом новизни даної статті. Функції цього блоку обумовлені тим, що оптимальна стратегія розвитку розраховується на певний «плановий період» T_p тривалістю 2-10 років в середньому. На такий період не можна точно передбачити стан технологій, ринків продукції і фінансів. Тому плановий період розбивається на інтервали. Оптимальна стратегія розраховується від початку чергового інтервалу до моменту T_p , з урахуванням уточнених прогнозів.

Останній блок – альтернатива функції Гамільтона. Функція Гамільтона – залежність прирощення інтегрального критерію розвитку від розподілу поточного ресурсу між витратами розвитку і прирощення накопичення. В даному випадку поточні ресурси розподіляються спочатку між розвитком і виробництвом (блок 3), а потім розподілом між цим результатом і накопиченням.

Таким чином, визначено п'ять задач, рішення, які необхідно для розробки узагальненої моделі оптимального розвитку. Задача декомпозиції процесу розвитку являється центральною частиною даної

статті і являється новим науковим результатом. Інші етапи рішення задачі розглядалися авторами у попередніх роботах [7, 8].

Розробка параметризованого оператора переходу між інтервалами розвитку

На рисунку 5 представлена деталізація завдання розробки параметризованого оператора переходу між інтервалами процесу розвитку. У верхній частині рисунку 5 – структура процесу оптимального функціонування і розвитку виробничої системи (ВС). У початковий момент, на базі початкового стану виробничої системи обчислюється оптимальна стратегія розвитку для «планового періоду» $(0, T_n)$, далі після закінчення деякого інтервалу ΔT обчислюється оптимальна стратегія для періоду $(\Delta T, T_n)$, з урахуванням реального поточного стану ВС, оточення і уточнених прогнозів на майбутнє. Управління на поточному інтервалі формується як деяка автономна задача, для якої задаються «квант ресурсу» ΔX_{sn} , початковий стан ВС і початкові значення параметрів підсистем для виконання оптимального агрегування «виробництво, розвиток» [5, 7]. Результат оптимального агрегування дає початкові дані для наступного інтервалу. Задане стратегічне управління і відпрацювання «квантів ресурсу» реалізуються на рівні оперативного управління. В рамках класичних методів стійке і якісне управління істотно нелінійної нестаціонарної стохастичною системою в загальному вигляді – майже нерозв'язна задача.

Однак, оптимальне агрегування дозволяє розподілити управління виробничою системою між елементами бінарного дерева оптимального агрегування. У нижній частині рисунку 4 представлений приклад моделі динаміки складної системи. Показаний вектор стану smt і оператор переходу між станами Ypr – програмний модуль структура якого узгоджена зі структурою стану. Такі моделі розроблені для динамічних систем класів «СМО», «оптимальний розвиток», «ринок лінійки продуктів», «управління запасами» [8].

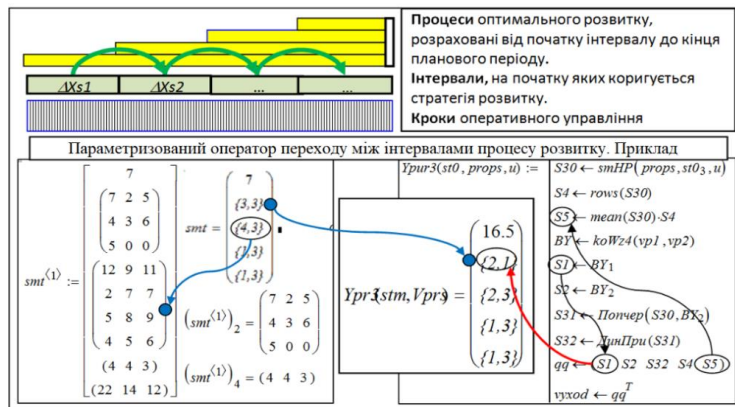


Рисунок 5 – Розробка параметризованого оператора переходу між інтервалами процесу розвитку [8]

Тестування оператора «перехід між інтервалами процесу розвитку»

На рисунку 6 подано тестування елементів динаміки оптимально агрегованих систем.

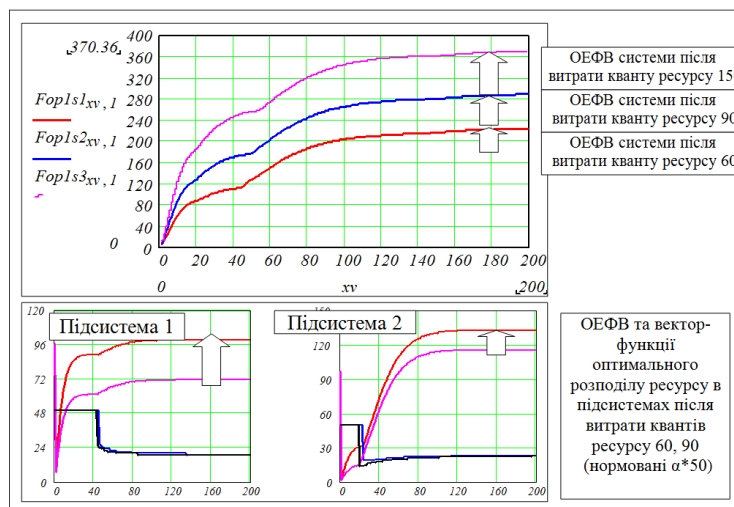


Рисунок 6 – Тестування нового оператора «перехід між інтервалами процесу розвитку». Тестування

В верхній частині подано зміну ОЕФВ тестової системи з чотирьох елементів. Після відпрацювання трьох інтервалів, для яких верхній рівень стратегічного управління виділяв послідовно 60, 90, 150 одиниць (нормованих) ресурсу. Бачимо зростання ефективності ОЕФВ.

В нижній частині тестування підсистем: для підсистеми 1 – 90 одиниць ресурсу, для підсистеми 2–60 одиниць. Це тестування синтаксичної коректності системи і підсистем. Бачимо: в підсистему 1 неефективно вкладати більше 40 одиниць ресурсу, а для підсистеми 2 більше 70 одиниць. Розроблено працездатний «інструмент», придатний для проведення досліджень реальних систем на віртуальній реальності.

Тестовий приклад підтверджує коректність функціонування складної для формалізації ресурсно-інформаційного ланцюга з параметричними зв'язками: витрати на підсистему «розвиток» змінюють параметри функції «витрати, випуск» підсистеми «виробництво», витрати в підсистему «виробництво» створюють матеріальний продукт. Розроблена модель не враховує інформаційні зворотні зв'язки: «виробництво» – джерело ідей і задач для «розвитку». Модель на базі методології оптимального агрегування відкрита для такого напрямку вдосконалень.

Висновки

В даній роботі розглядалася проблема побудови ефективних моделей оптимального розвитку та функціонування сучасних виробничих систем, що функціонують у активному оточенні. Проведено аналіз стану сучасних виробничих систем та встановлено, що сучасні виробництва є переважно інтегрованими, та мають велику кількість ресурсних та інформаційних зв'язків. Для моделювання було обрано інтегровану структуру «виробництво-розвиток», як найбільш типову структуру сучасного виробництва.

Виконано обґрунтування вибору методології оптимального агрегування. Метод оптимального агрегування знімає проблему великої розмірності об'єкту, оскільки дозволяє розбити багатовимірну задачу на послідовність одновимірних задач оптимізації.

Було проведено аналіз моделі базового прототипу та встановлено, що модель являється адекватною лише для статичного середовища. Запропоновано підхід декомпозиції процесу розвитку на інтервали, та корекції стратегії розвитку на початку кожного інтервалу.

Виконана розробка оператора переходу між інтервалами процесу розвитку та вирішена задача оптимального розвитку для структури «виробництво-розвиток» з використанням нового оператора. У результаті рішення отримали набір інтерактивних робочих моделей.

Актуальність у теоретичному плані – ефективна модель для рішення задачі розвитку виробничих систем у активному оточенні; у практичному – розроблено інструмент для тестування реальних виробничих систем у програмному середовищі.

Список літератури

- [1] E. Jantsch, *Technological forecasting in perspective*. Paris: Organization for Economic Co-operation and Development, 1967.
 - [2] Р. Беллман, Р. Калаба, *Динамическое программирование и современная теория управления*. М.: Наука, 1969.
 - [3] Т. М. Боровська, *Математичні моделі функціонування і розвитку виробничих систем на базі методології оптимального агрегування*. Вінниця, Україна: ВНТУ, 2018.
 - [4] Т. М. Боровська, Д. І. Гришин, І. С. Колесник, В. А. Северілов, "Розробка моделей і методів оптимального управління системами проектів на базі методів оптимального агрегування", *Вісник Вінницького політехнічного інституту*, №1(148), с. 61-76. 2020.
 - [5] T. Borovska "Optimal aggregation of production systems with parametric connections", *Eastern-European Journal of Enterprise Technologies*, vol. 4, no. 11(70), pp. 9-19. 2014.
 - [6] N. Tauchnitz, "The Pontryagin maximum principle for nonlinear optimal control problems with infinite horizon", *Journal of Optimization Theory and Applications*, no.167(1), pp. 27-48. 2015.
 - [7] Т. Н. Боровская, И. С. Колесник, В. А. Северилов, И. В. Шульган "Оптимальное агрегирование интегрированных систем "производство-развитие"", *Інформаційні технології та комп'ютерна інженерія*, № 2(30), с. 18-28. 2014.
 - [8] T. M. Borovska, I. V. Vernigora, D. I. Grishin, V. A. Severilov, W. Wójcik, and M. Kalimoldayev, "Generalized model of optimal development of the production system based on optimal aggregation methodology", *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019*. 2019.
 - [9] C. Taylor. "Dynamic programming and the curses of dimensionality", *Applications of dynamic programming to agricultural decision problems*. CRC Press, pp 1-10. 2019.
 - [10] T. W. Leggatt, *The evolution of Industrial Systems*. London: Croom Helm, 1985.
- Стаття надійшла: 20.09.2022.

References

- [1] E. Jantsch, *Technological forecasting in perspective*. Paris: Organization for Economic Co-operation and Development, 1967.

- [2] R. Bellman, *Dynamic programming and modern control theory*. M.: Nauka, 1969 [in Russian].
- [3] T. Borovska, *Mathematical models of the functioning and development of production systems based on the methodology of optimal aggregation*. Vinnitsya, Ukraine: VNTU, 2018 [in Ukrainian].
- [4] T. Borovska, D. Hryshyn, I. Kolesnik, V. Severilov, "Development of models and methods of optimal management of project systems based on optimal aggregation methods", *Visnyk Vinnytskoho politekhnichnoho instytutu*, no. 1(148), 61-76. 2020 [in Ukrainian].
- [5] T. Borovska, "Optimal aggregation of production systems with parametric connections", *Eastern-European Journal of Enterprise Technologies*, 4(11(70)), pp 9-19. 2014. doi: 10.15587/1729-4061.2014.26306.
- [6] N. Tauchnitz, "The Pontryagin maximum principle for nonlinear optimal control problems with infinite horizon", *Journal of Optimization Theory and Applications*, no. 167(1), pp. 27-48. 2015.
- [7] T. Borovska, I. Kolesnik, V. Severilov, I. Shulhan, "Optimal aggregation of integrated systems "production-development"", *Informatsiini tekhnolohii ta kompiuterna inzheneriia*, no. 2(30), pp. 18-28. 2014 [in Ukrainian].
- [8] T. M. Borovska, I. V. Vernigora, D. I. Grishin, V. A. Severilov, W. Wójcik, & M. Kalimoldayev, "Generalized model of optimal development of the production system based on optimal aggregation methodology", *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2019*. 2019.
- [9] C. Taylor. "Dynamic programming and the curses of dimensionality", *Applications of dynamic programming to agricultural decision problems*. CRC Press, pp 1-10. 2019.
- [10] T. W. Leggatt, *The evolution of Industrial Systems*. London: Croom Helm, 1985.

Відомості про авторів

Гришин Дмитро Ігорович – аспірант групи 126-19а, кафедра комп'ютерних систем управління.

Боровська Таїса Миколаївна – доктор технічних наук, професор, професор кафедри комп'ютерних систем управління.

D. I. Hryshyn, T. M. Borovska

ELABORATION AND RESEARCH OF A MODEL OF OPTIMAL PRODUCTION AND DEVELOPMENT OF INDUSTRIAL SYSTEMS WITH DECOMPOSITION OF THE DEVELOPMENT PROCESS TO THE INTERVALS

Vinnitsya National Technical University, Vinnitsya

УДК 37.022.32: 681.3

С. А. Кирилащук, З. В. Бондаренко, В. І. Клочко, Т. Г. Кирилащук

ЧИСЛОВЕ МОДЕЛЮВАННЯ ЯК ОДНА ІЗ СКЛАДОВИХ ФУНДАМЕНТАЛЬНОЇ МАТЕМАТИЧНОЇ ПІДГОТОВКИ ІНЖЕНЕРІВ

Вінницький національний технічний університет, Вінниця

Анотація. У статті досліджено проблему оволодіння навичками числового моделювання студентами інженерних спеціальностей під час вивчення курсу вищої математики. Значна увага приділяється формуванню творчого мислення студентів у процесі навчання вищої математики. Основними компонентами методичної системи є використання систем комп'ютерної математики (СКМ) під час лекцій, практичних та індивідуальних занять. Використовуючи СКМ MathCAD, Maple та інші, студент може абстрагуватися від технічних деталей програмування, особливостей операційної системи та зосередити увагу на аналізі особливостей таких понять, як обумовленість задачі, стійкість методу, оцінювання результатів розрахунків. Професійна спрямованість підготовки майбутніх фахівців розглядається як творча, сповнена сучасних знань з математики та інформатики діяльність. Наведені приклади завдань можуть бути використані студентами з високим когнітивним рівнем для самостійного опрацювання. Пропонується формування змісту курсу вищої математики здійснювати у таких напрямках: підсилення ролі числових методів і їх реалізацію за допомогою СКМ; використання математичного моделювання під час розв'язування прикладних задач; оволодіння студентами математичними змістово-фаховими знаннями, необхідними для аналітичного і числового моделювання практичних інженерних задач, створення власних бібліотек користувача програмних продуктів. Виділено головні вміння й навички, які можуть бути сформовані у студентів під час інтегрованих занять з вищої математики і елементів числового моделювання: вміння використовувати необхідні програмні засоби в середовищі СКМ; вміння зіставляти результати в різних формах подання (аналітичне й графічне подання); вміння використовувати нові можливості, що надаються комп'ютерними технологіями, заснованими на використанні середовища СКМ; вміння проектувати в середовищі СКМ програмні засоби першого рівня складності для розв'язування навчальних і предметних завдань.

Ключові слова: курс вищої математики, поняття числового моделювання, стійкість, оцінювання результатів.

Abstract. The article investigates the problem of mastering the skills of numerical modeling by students of engineering specialties while studying the course of higher mathematics. Much attention is paid to the formation of creative thinking of students in the process of learning mathematics. The main components of the methodological system using SCM in lectures, practical and individual classes.

Using SCM MathCAD, Maple and others, the student can abstract from the technical details of programming, features of the operating system and focus on analyzing the features of such concepts as conditionality of the problem, stability of the method, evaluation of calculation results. Professional orientation of training future professionals with good personal qualities and is seen as creative, full of modern knowledge in mathematics and computer science. The given examples of tasks can be used for independent solution by students with a high cognitive level. Forming of maintenance of course of higher mathematics is offered to carry out in such directions: strengthening of role of numerical methods and their realization after dopomogo of SKM; the use of mathematical design is during untiing of the applied tasks; capture students mathematical semantically professional by knowledges, necessary for the analytical and numerical design of practical engineering tasks, creation of own libraries of user of software products. Main abilities and skills which can be formed for students during computer-integrated employments after higher mathematics and elements of numerical design are selected: ability to use necessary programmatic facilities in the environment of SKM; ability to compare results in the different forms of presentation (analytical and graphic presentation); ability to use new possibilities, which are given computer technologies, based on the use of environment of SKM; ability to design programmatic facilities of the first in the environment of SKM; ability to design in the environment of SKM programmatic facilities of the first level of complication for untiing of educational and subject tasks.

Key words: numerical modeling, higher mathematics, stability, evaluation of calculation results.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-74-84>.

Вступ

Використання математичних методів у різноманітних галузях інженерної діяльності потребує певного рівня математичної компетентності і високого рівня математичної підготовки. Рівень розвитку науки і техніки вимагає від фахівців, перш за все, самостійного поповнювання знань та оволодіння новими знаннями.

Інженерна практика наших днів все частіше зустрічається з математичними задачами, точний розв'язок яких досить складний, тому застосовують ті чи інші наближені методи обчислення, які набули за останні роки широкого розвитку.

У законодавчих документах про освіту передбачено спрямованість навчання у ЗВО в напрямку розвитку інтелектуальних здібностей студентів. Це потребує переорієнтації самостійної діяльності студентів переважно на вирішення навчальних професійно зорієнтованих завдань, що реалізується через засвоєння необхідного мінімуму інформації та використання довідкових навчальних засобів [4;8]. Засобами реалізації такого підходу на заняттях з вищої математики може бути математичне моделювання з використанням СКМ MathCAD, Maple, Maxima, Derive, Excel і інші. Це пов'язано з тим, що моделювання визнано універсальним методом пізнання, а, отже, ефективним інструментом навчальної діяльності [2;5;12].

Числове моделювання – це досить складний курс у циклі інформатичних дисциплін. У підготовці фахівців, він є міждисциплінарним курсом. Для успішного оволодіння ним потрібні найрізноманітніші знання. Для моделювання фізичних процесів потрібні знання в обраній предметній галузі, необхідно мати певний рівень знань законів фізики. Моделюючи економічні процеси – потрібні знання законів економіки тощо. Крім того, оскільки комп'ютерне моделювання використовує практично весь апарат сучасної математики, передбачається наявність знань з основних математичних дисциплін – алгебри, математичного

аналізу, теорії диференціальних рівнянь, математичної статистики, теорії імовірності. Для вирішення математичних завдань необхідно володіти в повному обсязі числовими методами розв'язування нелінійних рівнянь, систем лінійних рівнянь, диференціальних рівнянь, елементами функціонального аналізу, теорії наближення функції тощо.

Актуальність

Нині в теорії та практиці вищої освіти накопичено значні наукові напрацювання, що висвітлюють нові тенденції і потреби її розвитку в умовах суспільного реформування. Це роботи з питань інформатизації освіти взагалі й математичної освіти зокрема (М. І. Жалдак, В. І. Клочко, Н. В. Морзе, С. А. Раков, С. О. Семеріков, О. В. Співаківський, Ю. В. Триус та ін.); з питань впровадження компетентнісного та інтеграційного підходів у вищу освіту (М. І. Жалдак, І. М. Козловська, Ю. С. Рамський, Г. К. Селевко, Р. М. Собко, О. М. Спірін, Т. Д. Якимович та ін.); з питань комп'ютерного моделювання, зокрема використання систем комп'ютерної математики (СКМ), у процесі навчання математики (З. В. Бондаренко, С. А. Кирилашук, В. І. Клочко, Т. П. Кобильник, Н. М. Кузьміна, В. М. Михалевич і інші). Аналіз їхніх наукових здобутків показав, що впровадження у навчальний процес даного підходу є основою підготовки фахівців з вищою освітою, які здатні забезпечувати конкурентоспроможність України. Незважаючи на значні наукові здобутки в теорії педагогічної освіти, проблема математичної підготовки майбутніх фахівців, знаходження шляхів оновлення змісту математичної освіти на основі інтеграції математичних і спеціальних інформатичних дисциплін на методологічному, методичному та змістовному рівнях, не вирішена, тому у статті розглядаються порушені питання

Мета

Метою даної роботи є з'ясування можливостей в курсі вищої математики знайомити студентів із особливостями складових моделювання, зокрема числового моделювання.

Задачі

Перша. Розглянути деякі аспекти технології інтегративного навчання вищої математики з елементами числового моделювання.

Друга. Формування комплексу компетентнісно-орієнтованих завдань, вирішення яких вимагає встановлення різноманітних інтеграційних зв'язків інформатичних і математичних дисциплін.

Третя. Виділити головні вміння й навички, які можуть бути сформовані у студентів під час інтегрованих занять з вищої математики і елементів числового моделювання.

Розв'язання задач

Сьогодні не можна готувати фахівців завтрашнього дня, не включаючи в навчальні програми базової математичної підготовки нові розділи математики розроблені в останні десятиліття.

Залежно від аудиторії студентів та їх спеціалізації це можуть бути теорія вейвлетів, матричний аналіз або методи розв'язання нелінійних рівнянь і т.д. Нові математичні поняття, зокрема ті, що відносяться до числового моделювання, можуть бути впроваджені в навчальний процес: 1) частково за рахунок деякого ущільнення програм зі стандартного курсу вищої математики (це цілком можливо, оскільки з ключовими поняттями математичного аналізу похідної та визначеним інтегралом студенти знайомляться, хоча й недостатньо глибоко, ще в середній школі); 2) частково за рахунок необов'язкового, незатребуваного (фахівцями даного профілю) матеріалу. Тобто, формуються своєрідні "сплайн-розділи" курсу вищої математики.

Числові методи дозволяють звести розв'язування задачі до виконання скінченного числа арифметичних і логічних дій з числами. При цьому розв'язок визначається як набір чисел, які надалі можуть бути інтерпретовані різним способом (наприклад, подані у вигляді таблиць, графіків, анімації тощо). Їх перевагами є: абсолютна універсальність, тому що теоретично можуть бути застосовані для розв'язання переважної більшості задач. Таким чином, числові методи є основним апаратом розв'язання математичних задач, а їх значущість тільки збільшуватиметься у міру вдосконалення комп'ютерної техніки.

Головним завданням числових методів є знаходження розв'язку з необхідною точністю, що оцінюється. У знаннях, яких набуває студент під час вивчення числових методів, інтегруються теоретично-методичні знання та практичні уміння і навички. За рахунок конкретності характеру використання матеріалу, створюються умови його ґрунтовного засвоєння, формування математичних і професійно спрямованих компетентностей: оволодіння методами аналізу і синтезу, інтерпретування отриманого математичного результату, здатність застосовувати методи аналізу та синтезу і числового моделювання за допомогою СКМ.

Виділимо головні вміння й навички, формування яких поглиблюється у студентів під час вивчення дисципліни вищої математики: уміння використовувати необхідні програмні засоби в середовищі СКМ. Згідно з Россом Тернером [9], до набору компетентностей, які є фундаментальними для загальнокультурного розвитку людини, входять такі компетентності: уміння аналізувати результати у різних формах подання інформації (аналітичного й графічного подання); уміння створювати в середовищі СКМ програмні засоби доступного рівня складності для розв'язання навчальних і предметних завдань; уміння виявлення

помилку, не довіряючи результату обчислень; уміння вибирати з цілої низки рішень оптимальний. Студенти набувають досвіду подання завдання у вигляді програми, що була б простішою у використанні.

Побудова навчального процесу з урахуванням компетентностей XXI століття покликана забезпечити набуття всіма студентами знань і навичок, необхідних для того, щоб досягти успіху в світі, де зміни є постійними і навчання ніколи не зупиняється [3].

Слід зазначити, що під час вивчення курсу вищої математики або додаткових математичних дисциплін у студентів виникають значні труднощі, які пояснюються тим, що у них не сформувалася здатність аналізувати стан задачі, виділяти суттєві зв'язки між даними і компонентами системи, знаходити раціональний метод розв'язання, аналізувати результати. Для розвитку таких навичок доцільно застосовувати різні форми самостійної роботи студентів, оскільки самостійна робота над навчальним матеріалом – головне в освітній діяльності кожного студента. Активізація пізнавальної діяльності студентів, розвиток здатності самостійно вирішувати проблеми можуть бути досягнуті в результаті індивідуалізації навчання.

Етап формалізації математичної задачі (математична постановка) є першим етапом моделювання. На цьому етапі описується система, що досліджується: визначається її цільове призначення, характер діяльності, використовувані ресурси, описуються нормативні параметри. При постановці завдання відбувається вивчення об'єкта моделювання (системи, процесу), аналіз доступної інформації, визначення обмежень та припущень. Побудова математичної моделі починається з формулювання системи обмежень, у яких функціонуватиме дана модель, а також сукупності правил, що визначають допустимі значення проміжних результатів обчислення.

До одного із можливих завдань курсу вищої математики відноситься організація систематичного цілеспрямованого вивчення окремих складових технологій комп'ютерного та числового моделювання. Це повинно розв'язати суперечності між можливостями інформатизованої методичної системи розвитку інтелектуальних здібностей студентів у процесі застосування комп'ютерного моделювання у навчанні вищої математики і реальним навчальним процесом.

Для оцінки числових методів, тобто порівняння між собою методів для розв'язання однієї задачі, вводять такі їх основні характеристики: трудомісткість; порядок методу; збіжність; швидкість збіжності; стійкість до похибок обчислень; стійкість до похибок у початкових даних.

Розгляд питання про способи зменшення обчислювальних похибок доцільно розпочати з конкретних прикладів типу: обчислення значення $\sin(x)$, користуючись розкладом функції за степенями x , використовуючи для цього на лекції відповідні СКМ. Скориставшись програмою і задавши значення аргументу

$$x = 0.5236 \quad (x \approx \frac{\pi}{6}) \quad \text{і} \quad x = 25.6564 \quad (x \approx 4 \cdot 2\pi + \frac{\pi}{6}) \quad \text{відповідно отримаємо} \quad 0.500001;$$

0.5000519569 . На прикладах згаданого типу і інших звертаємо увагу на ті проблеми, які виникають при розв'язуванні задач з використанням СКМ. Досвід показує, що такий підхід до вивчення цих питань дає позитивні наслідки не лише при вивченні розділу, а й курсу в цілому.

Творчий інженер не обмежується лише використанням алгоритмів. Він повинен знати, як це робиться. В іншому випадку він перестає бути інженером і стане заручником комп'ютера.

Зазвичай, на перших порах використання будь-якої із систем символьних обчислень, студент досить легко вирішує невеликі і нескладні навчальні приклади і задачі. Проте, під час розв'язання фахових задач, він стикається з низкою проблем: суттєві витрати часу на обчислення, не вистачає пам'яті, часто результат видається неправильний або не завжди зрозумілим. Тому, під час вивчення основного змісту курсу вищої математики варто знайомити студентів з контрприкладом та розв'язувати їх за допомогою СКМ. Це, до деякої міри, переконає студентів у тому, що не продумане застосування СКМ може нести в собі великі проблеми.

Досвід роботи з першокурсниками показує, що вони не завжди чітко усвідомлюють, що означає знайти корінь рівняння заданої точності. За означенням, x_i є наближенням кореня рівняння $f(x) = 0$ з точністю ε , якщо виконується умова $|f(x_i)| < \varepsilon$, де ξ – точний корінь рівняння даного рівняння.

Проблема в тім, що точний корінь ξ невідомий (та й він переважно і не буде знайденим), тому й саму різницю $|x_i - \xi| < \varepsilon$ порахувати неможливо. У студентів, які вільно оперують числами, виникають запитання: "Що ж порівнювати?", "Коли ж зупиняти процес знаходження кореня?", "Чому найчастіше користуються обмеженнями $y = (x + C)e^{-\sin x}$, $y = (x + 2)e^{-\sin x}$, або $|f(x_i)| < \varepsilon$ для забезпечення заданої точності?". Найкраще вирішити ці проблеми графічно, показавши відрізки $|\zeta, x_i|$, $|x_i, x_{i-1}|$, $|f(x_i)|$.

Така наочність допоможе визначити незрозумілі місця і, опріч цього, дасть змогу вибирати критерій зупинки процесу залежно від поведінки функції $f(x)$ (наприклад, для швидко зростаючої $f(x)$ умова

$|f(x)| < \varepsilon$ не годиться). Точні оцінки кореня для кожного методу, зокрема можна подати без виведення, або запропонувати звернутись до літератури [7; 15].

Деякі методи не передбачають задання початкових наближень кореня. Таким чином, процес формування компетентності включає знання про доступні допоміжні засоби та інструменти (методи), а також їх потенціал, обмеження та можливість використовувати їх вдумливо та ефективно.

Поряд із задачами розв'язування рівнянь, значна кількість інженерних задач зводиться з рештою до наближеного розв'язування як конкретних рівнянь так і систем рівнянь, що описують поведінку об'єкта дослідження.

Особливу увагу доцільно звернути на поняття стійкості, оскільки вивчення стійкості є одним з центральних питань чисельних методів. Звертаємо увагу студентів і на те, що необхідно розрізняти стійкість задачі і стійкість алгоритму її розв'язання. Доцільно не обмежуватись теоретичними питаннями, а навести конкретні приклади нестійких задач (задача диференціювання, приклад Уілкінсона і ін.) із нестійких алгоритмів.

До нестійких алгоритмів також відноситься пошук коренів многочленів великих степенів. У свій час сподівались розв'язати цю проблему за допомогою швидкодії обчислення. Прикладом проблеми є пошук коренів многочлена $p(x) = (x-1)(x-2)\dots(x-20)$, якщо розкрити дужки і до коефіцієнта $a_{19} = 210$ при x^{19} додати 2^{-23} , тобто, $\tilde{a}_{19} = 210 + 2^{-23}$. Очевидними коренями многочлена є $x_1 = 1, x_2 = 2, \dots, x_{20} = 20$. Але обчислення при $\tilde{a}_{19} = 210 + 2^{-23}$ за допомогою СКМ дає частину – 10 дійсних коренів $x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 3.91$, і частину – 10 комплексних коренів таких як $x_{19} = 21.129 - 1.996i, x_{20} = 21.672 - 0.552i$. На деяких СКМ можуть бути отримані комплексно-спряжені корені. Отриманий результат протирічить інтуїції і наявності очевидних коренів. Проте проблему нестійкості алгоритмів обчислення коренів многочлена математиками було розв'язано.

Необхідно оцінювання чутливості задачі відносно похибок. Запишемо поліном у вигляді

$$p(x, \tilde{a}_{19}) = x^{20} + \tilde{a}_{19}x^{19} + \dots$$

Далі обчислюється похідна
$$\frac{\partial x}{\partial \tilde{a}_{19}} = -\frac{\frac{\partial p}{\partial \tilde{a}_{19}}}{\frac{\partial p}{\partial x}}.$$

Міра чутливості для коефіцієнта \tilde{a}_{19} складає 3.1×10^8 , одна з найбільших (для $\tilde{a}_4 = 2.2 \times 10^{-3}$). Оскільки наближені результати розв'язування задачі без їх точності немає цінності, то вивчення необхідних правил, методів обов'язково повинно бути у навчанні студентів і не лише під час вивчення числових методів (здатність застосовувати аналітичні та числові методи вирішення завдань за допомогою СКМ).

Наприклад, для рівняння $\frac{1}{x} = 0$, яке, очевидно, не має дійсних коренів, але для довільної, як завгодно

малої точності, знайдеться x , що задовольняє критерію $|f(x_{k+1})| \leq \varepsilon : \left| \frac{1}{x_{k+1}} \right| \leq \varepsilon$, якщо $\varepsilon = 0.00001$, то

$x_{k+1} = 200000$, тобто знайдено число, що задовольняє заданий критерій точності пошуку розв'язку задачі.

Наведені приклади показують, що до результатів комп'ютерних обчислень необхідно завжди ставитися критично, аналізувати їх щодо правдоподібності. Щоб уникнути "підводних каменів" під час використання будь-якого стандартного пакета, що реалізує числові методи, потрібно мати уявлення про те, який саме числовий метод реалізований для вирішення тієї чи іншої задачі. Наприклад, коли відомий інтервал, в якому розміщений корінь, то можна скористатися іншими методами уточнення кореня.

Однією з проблем наближених методів розв'язування диференціальних рівнянь є аналіз похибок та стійкості. Тому особливу увагу доцільно на це звернути, оскільки вивчення стійкості є одним з центральних питань не лише в галузі звичайних диференціальних рівнянь, а й числових методів взагалі. На за-

няттях з вищої математики викладач звертає увагу студентів і на те, що необхідно розрізняти стійкість задачі і стійкість алгоритму її розв'язання. Доцільно не обмежуватись теоретичними питаннями, а навести конкретні приклади нестійких задач і нестійких алгоритмів. Зокрема, можна навести приклад диференціального рівняння $y'(t) = \lambda y(t)$, $\lambda \in \mathbb{C}$, $t \geq 0$, $y(0) = y_0$. Це рівняння є моделлю для прогнозування стійкості числових методів розв'язування нелінійних систем диференціальних рівнянь. Щодо нестійкості алгоритмів можна запропонувати студентам довести, що сума $y_{n+1} = -y_n + 2y_{n-1} + 3hf(x_n, y_n)$ двох стійких алгоритмів $y_{n+1} = y_{n-1} + 2hf(x_n, y_n)$ та $y_n = y_{n-1} + hf(x_n, y_n)$ не дає стійкого алгоритму.

Проаналізувати похибки, що породжуються алгоритмом, можна шляхом дослідження впливу похибок на заокруглення в арифметичних операціях. Однією з багатьох причин виникнення похибок у результатах обчислень є похибки заокруглення під час множення і ділення у скінченно розрядній арифметиці є втрата значущих цифр під час операції віднімання близьких чисел.

Студентам це можна показати на прикладі обчислення часткової суми (із використанням СКМ)

$$S_{1000000} = \sum_{n=1}^{1000000} \frac{1}{n^2} \text{ ряду } \frac{1}{n^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots + \frac{1}{n^2} + \dots \text{ за двома алгоритмами}$$

$$S_n = S_{n-1} + \frac{1}{n^2}; \quad n = 1, 2, \dots, 1000000, \quad S_0 = 0;$$

$$\sum_{n=1} = \sum_n + \frac{1}{n^2}; \quad n = 1000000, \dots, 2, 1; \quad \sum_{1000000} = 0$$

Студенти переконуються у тому, що похибка обчислення за другим алгоритмом на 20% менша. Досить часто втрата значущих цифр суттєво спотворює результат під час розв'язування СЛАР.

Ознайомимось також із іншими поняттями проблем числового моделювання та відповідним оцінюванням процесу і результатів моделювання.

Системи лінійних алгебраїчних рівнянь (СЛАР) $AX = B$ часто виникають із експериментів, то коефіцієнти системи отримуються з похибками, коли навіть їх задають формулами, до того ж, під час обчислення, виникають похибки округлення коефіцієнтів. У зв'язку з цим виникає важливе питання, як впливають похибки коефіцієнтів системи на розв'язок. Або іншими словами, як можна виміряти чутливість розв'язку X по відношенню до змінювання матриць A і B ?

Оцінювання відносної похибки правої частини системи викликають зміни у розв'язку більші у $C(A)$ разів. Де $C(A) = \|A\| \|A^{-1}\|$ міра (число) обумовленості матриці і використовується для грубішого оцінювання характеристик матриці, $\|A\|$ – норма матриці, що обчислюється за однією з формул $\|A\| = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$, число обумовленості матриці обчислюємо за такою формулою (чому?):

$$C(A) = \frac{\Delta X \cdot \|B\|}{\Delta B \cdot \|X\|}.$$

Приклад. Нехай систему $AX = B$ задано матрицями $A = \begin{pmatrix} 10 & 50 \\ 15 & 75.1 \end{pmatrix}$, $B = \begin{pmatrix} 110 \\ 165 \end{pmatrix}$. Розв'язком системи є матриця $X = \begin{pmatrix} 11 \\ 0 \end{pmatrix}$. Змінимо лише елементи матриці B , $B_1 = \begin{pmatrix} 110 \\ 165.04 \end{pmatrix}$, розв'язок змінюється

$$\text{суттєво } X_1 = \begin{pmatrix} 9 \\ 0.4 \end{pmatrix}.$$

Виконаємо відповідні обчислення норм матриць і число обумовленості матриці

$$\|B\| = 275, \quad \|X\| = 11, \quad \Delta B = \|B_1 - B\| = 0.04, \quad \Delta X = \|X_1 - X\| = 2.6, \quad C(A) = \frac{\Delta X \cdot \|B\|}{\Delta B \cdot \|X\|} = 1625.$$

Велике число обумовленості матриці пояснює чому малі зміни матриці B (відносна похибка 0.000145 (0.0145%)), викликала суттєві зміни розв'язку – 0.236 (23.6%).

В основному виникають такі проблеми під час розв'язування систем великої розмірності із застосуванням комп'ютерної техніки, вони пов'язані: з великим об'ємом оперативної пам'яті комп'ютера, необхідної при проведенні розрахунків із матрицею; великий об'єм самих розрахунків; матриця системи може бути виродженою; накопичення помилок обчислення. Великий об'єм самих розрахунків приводить до того, що для розв'язання задачі необхідно багато часу. Також при цьому відбувається накопичення похибок обчислення, що приводить до неправильного результату. Для вирішення цих проблем застосовують деякі прийоми та спеціальні чисельні методи.

Однією із проблем, що розв'язуються за допомогою сучасної обчислювальної математики, є наближення функцій однієї змінної та багатьох дійсних змінних іншими функціями більш простої, взагалі кажучи будови. Інша назва цієї задачі – апроксимування функції.

Із розвитком комп'ютерної техніки значно розвинулись і числові методи. При цьому важливу роль відіграє теорія наближення функції, оскільки на ній базуються алгоритми побудови апроксимаційних поліномів, що дає можливість, наприклад, будувати квадратурні формули. Для студентів цікавими виявилися завдання на побудову власних формул наближених обчислень визначених інтегралів. Широким полем для

навчально-дослідницької роботи може бути використання формули Гаусса
$$\int_a^b f(x)dx = \sum_{k=1}^n A_k f(k).$$

Власну формулу студент може вивести, вибравши довільні вузли x_k , або вузли спеціальних функцій, дослідити, отримати оцінку точності формули. Далі ці результати дослідження можуть застосовуватися під час розв'язування крайових задач диференціальних рівнянь.

Однією із вимог до числових методів раціональної апроксимації є надійність. Не завжди існує раціональна функція певного виду, що задовольняє накладеним умовам інтерполяції. Якщо метод апроксимації надійний, то програма має вказати, що задача не має розв'язку. Числовий алгоритм повинен розрізняти задачі що мають і не мають розв'язків з врахуванням помилок представлення та округлення. Аналіз цього питання приводить нас до поняття стійкості алгоритму, яке тісно пов'язане з поняттям надійності. Алгоритм стійкий, якщо малі зміни початкових даних приводять до невеликих змін результату. Хороший алгоритм раціональної інтерполяції повинен бути в змозі виділити ті випадки, коли початкові дані приводять до нестійкого результату.

Студентів слід ознайомити з постановкою задачі наближення функцій, сутністю методів наближення (інтерполювання, середнє квадратичне наближення, рівномірне наближення), оптимальним добором вузлів інтерполювання, найпростішими інтерполяційними методами для розв'язування рівнянь з одним невідомим.

До переліку фундаментальних компетентностей загальнокультурного розвитку людини відносять математичну компетентність, під якою розуміють перетворення реальної проблеми у математичну (математичне моделювання), тлумачення математичних об'єктів або інформації щодо змодельованих ситуацій. Одним із недоліків моделювання є те, що будь-який модельний аналіз звужує зміст можливих тлумачень результату. Моделювання висвітлює про об'єкт рівно стільки, скільки можна "втиснути" в рамки моделі, що було названо "прокрустовим ложем" моделювання.

Наведемо перелік компетентностей, що входять до складу математичної компетентності фахівців та тісно пов'язані із змістом статті:

- здатність здійснювати пошук, критичний аналіз та синтез інформації, застосовувати системний підхід для вирішення поставлених завдань;
- здатність застосовувати природничі та загальноінженерні знання, методи математичного аналізу та моделювання, теоретичного та експериментального дослідження у професійній діяльності;
- здатність представляти та захищати свої математичні розв'язки.

Прикладом формування наведених компетентностей може бути розв'язання задачі Коші,

$$y' + y \cos(x) = \exp(-\sin(x)), \quad y(0) = 2$$

$$y = (x + C) e^{-\sin x}, \quad y = (x + 2) e^{-\sin x}$$

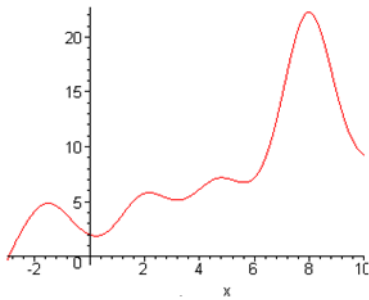


Рисунок 1 – Графік частинного розв'язку

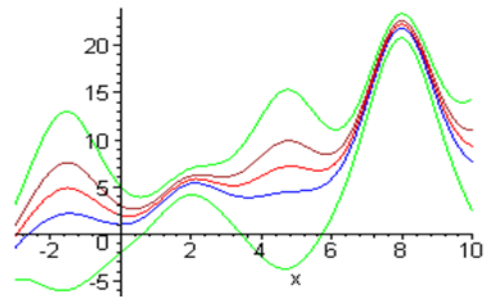


Рисунок 2 – Графік загального розв'язку

Студент доповідає власний аналіз відповідного процесу за результатом побудови фрагмента відповідної інтегральної кривої (рис. 1) та спробу передбачити, якими будуть інші інтегралі криві. Важливо проаналізувати зміст загальних та частинних розв'язків. Зміст загального розв'язку (звичайно, на множині побудованих графіків) дозволяє з'ясувати особливості динаміки процесу, виявити його поведінку в околі деяких точок. Аналіз таких ситуацій дозволяє усвідомити студентам значення математичного моделювання. Так, графік частинного розв'язку (рис. 1) вказує на наявність значної кількості екстремумів, і в той же час графіки загального розв'язку (рис. 2) інакше характеризують процес в околі деяких точок (точка *max* змінюється на точку *min*). Такий підхід до навчання математики формує здатність тлумачити інформацію, подану математичним способом, точно використовувати математику для передачі інформації та розв'язування задач, що розвине навички спілкування студентів мовою математики. Отже, вивчення математики в соціальному контексті дає змогу студентам посилити знання, необхідні в ситуаціях на робочому місці, і забезпечує кращу відповідність математики, необхідної в інженерній практиці.

Слід підкреслити загальну тенденцію розвитку математичних моделей реальних процесів, що намітилася останніми роками – їх суттєво нелінійний характер та підвищення порядку відповідних диференціальних рівнянь. Одним з механізмів підвищення порядку диференціальних рівнянь, що описують динамічні системи, є урахування зворотного впливу середовища на рух динамічної системи. Такий, наприклад, механізм появи похідних третього порядку у рівняннях електродинаміки. Тому актуальною є проблема створення та ознайомлення студентів з числово-аналітичними комп'ютерними методами дослідження і математичного моделювання цих процесів з урахуванням можливого змінювання їх параметрів.

Як відомо, немає досить ефективних та загальних методів аналітичного дослідження поведінки нелінійних систем, що описуються задачею Коші. Застосування методів якісної теорії диференціальних рівнянь вимагає, по-перше, автономності системи, а, по-друге, зі збільшенням числа ступенів свободи системи, складність дослідження за допомогою якісної теорії диференціальних рівнянь, а тим більше їх візуалізації, різко зростає. Фактично, єдиним надійним методом дослідження багатьох нелінійних систем є наближене розв'язування задачі Коші, що зводиться, зазвичай, до числового інтегрування нормальної системи ОДУ з відповідними початковими умовами. Але поряд з числовими методами застосовують наближені числово-аналітичні методи, що ґрунтуються, зокрема, на інтегральному перетворенні Лапласа або на сплайновій екстраполяції числових розв'язків задачі Коші [15].

Необхідність застосування досить складних числових методів, і пов'язана з цією обставиною необхідність професійного програмування, складність маніпуляцій з числовим розв'язанням, зокрема, складності візуалізації динаміки нелінійних динамічних систем є сукупним фактором, що різко обмежує область досліджуваних нелінійних систем, як математикам, так і фахівцям, які не є професійними програмістами.

Системи комп'ютерної математики, в принципі, помітно наближають таких фахівців до застосування методів комп'ютерного моделювання, але все ж таки тут стосовно дослідженню систем нелінійних ОДУ для таких фахівців зберігається помітна диспропорція між витраченими зусиллями та отриманням результату.

Крім того, оскільки комп'ютерне, числове моделювання використовує практично весь апарат сучасної математики, передбачається наявність знань з основних математичних дисциплін – алгебри, математичного аналізу, теорії диференціальних рівнянь, математичної статистики, теорії імовірності. Для вирішення математичних завдань необхідно володіти в повному обсязі числовими методами розв'язання нелінійних рівнянь, систем лінійних рівнянь, диференціальних рівнянь, елементами функціонального аналізу, теорії наближення функції тощо. І, звичайно ж, передбачається вільне володіння сучасними інформаційними технологіями, знаннями мов програмування й володіння навичками розробки прикладних програм.

Добір змісту понять числового моделювання, що розглядаються у курсі вищої математики ґрунтувався також на загально дидактичних принципах. Так, згідно принципу доступності передбачався змісто-

вий аналіз наявних і засвоєваних знань та умінь студентів, а також урахування "зони найближчого розвитку студента". Важливим фактором добору матеріалу було урахування рівня загальної підготовки студентів з курсу вищої математики, інформатики, фізики, оскільки це значно впливає на рівень опанування методологією моделювання.

Можна виділити наступні вимоги добору та подання теоретичного матеріалу курсу вищої математики: диференціація змісту; модульність; діалогова форма пояснення; надлишок інформації, необхідної для виконання завдань студентом; опосередковане керування роботою через текст (наголоси на основних положеннях теорії, її структурних зв'язках, формулювання правил тощо); використання наочності; виважена організація діяльності зі знаково-символьними засобами; висвітлення міжпредметних зв'язків; використання історичних відомостей щодо змісту тощо. На нашу думку, кожен наступний рівень розкриття змісту курсу вищої математики у поєднанні з елементами числового моделювання повинен мати вищий рівень науковості, абстрактності, проблемності.

На даний момент рівень математичної підготовки для багатьох спеціальностей інженерії потребує обов'язкових (глибоких) знань з таких розділів математики: основи матричного числення, основи інтегральних рівнянь, основи варіаційного числення, основи функціонального аналізу, основи математичних методів дослідження диференціальних рівнянь, основи математичного моделювання та багатьох інших. У зв'язку з цим потребує розширення наведених у статті понять числового моделювання та поглиблення їх змісту за рахунок введення в курс вищої математики понять наведених вище розділів.

Складнішими за технічною реалізацією і більш універсальними за можливостями є символічні, або аналітичні методи СКМ. Робота символічного процесора пов'язана з аналізом формули, що перетворюється, а це дозволяє, наприклад, під час інтегрування отримати відповідь у вигляді не тільки десяткового дробу, як у випадку застосування числового методу, а й у вигляді первісної. Крім того, символічні результати є абсолютно точними. Загалом, з упевненістю можна сказати, що за всіма параметрами аналітичний розв'язок в СКМ перевищує числові. Проте, на жаль, можливості використання символічних процесорів є дуже обмеженими, тому що для небагатьох задач можна отримати аналітичний розв'язок, а, крім того, не до всіх задач можна застосувати програми символічних перетворень.

Розглянемо деякі аспекти технології запропонованого інтегративного навчання вищої математики з елементами числового моделювання.

На лекції, на практичних заняттях викладач може використати побудовані математичні моделі для демонстрації явища, що розглядається. У процесі моделювання конкретного явища студент не тільки засвоює навчальний матеріал, але й набуває вміння ставити проблеми й завдання, передбачати результати дослідження, проводити оцінювання, виділяти головні й другорядні фактори для побудови моделей, вибрати аналогії й математичні формулювання, проводити аналіз обчислювальних експериментів.

Компетентністний підхід до навчання вищої математики, з урахуванням введених понять числового моделювання, визначається зв'язками між набутими математичними знаннями й вміннями та професійними компетентностями: здатність здійснювати пошук, критичний аналіз та синтез інформації, застосовувати системний підхід для вирішення поставлених завдань; здатність застосовувати природничі та загальноінженерні знання, методи математичного аналізу та моделювання, теоретичного та експериментального дослідження у професійній діяльності; здатність вирішувати стандартні завдання професійної діяльності на основі інформаційної та бібліографічної культури із застосуванням інформаційно-комунікаційних технологій.

Особливу значимість несе формування комплексу компетентісно-орієнтованих завдань у кожному модулі, вирішення яких вимагає встановлення різноманітних інтеграційних зв'язків інформатичних і математичних дисциплін. Кількість навчальних завдань, поставлених у кожному модулі, відповідає кількості елементів знань, які контролюються і оцінюються.

При доборі змісту враховуємо технологію професійної спрямованості вивчення курсу вищої математики, що реалізується у таких напрямках: методологічна насиченість змісту курсу вищої математики, що розвиває науковий світогляд студентів, методологічні знання, вміння реалізації функцій фахівця у професійній діяльності; міждисциплінарна взаємодія навчальних дисциплін, що дозволяє розкрити сутність вищої математики у якісному засвоєнні спеціальних інформатичних дисциплін, а також у майбутній професійній діяльності; використання комплексу навчально-професійних завдань, вирішення яких істотно впливає на мотивацію вивчення курсу вищої математики [4].

Побудова системи задач для інтегрованих занять ґрунтується також на принципах диференційованої реалізованості, систематичності і послідовності навчання. При цьому навчальна діяльність студентів спрямовується на засвоєння загальних способів дій і наукових понять під час розв'язання задач. У процесі засвоєння понять курсу вищої математики та числового моделювання викладач навчає студентів перетворенню логічної форми наукових знань у форму діяльності. Зокрема діяльність студентів спрямовується на здобуття системних знань щодо математики та числового і комп'ютерного моделювання в математиці, знань, що стосуються добору методів математичного моделювання, оцінювання похибок, стійкості методів, алгоритмів, застосування СКМ. До системи задач включались задачі, методи та алгоритми

розв'язування яких відомі студентіві, задачі проблемного характеру, що потребують розробки відповідних алгоритмів, проведення числового і комп'ютерного моделювання, аналізу отриманих результатів.

Оцінювання рівня досягнень результатів навчання здійснювалось за допомогою банку компетентнісно-орієнтованих тестових завдань, що дозволяє вести моніторинг сформованості професійних і спеціальних компетентностей майбутніх фахівців.

Крім того, оволодіння навичками уміння числового моделювання зорієнтовано на здоровий глузд студентів, їхній рівень мислення, кмітливість. Ми виходили з того, що під час знайомства з поняттями числового моделювання під час вивчення курсу вищої математики студенти можуть оволодіти простими питаннями комп'ютерного моделювання, що дозволить їм набути певного досвіду розв'язування конкретних прикладних задач, мати загальне уявлення та розвинути необхідну інтуїцію пошуку ефективних шляхів побудови результату числового моделювання. Мета та завдання практичних занять полягали у поглибленні та закріпленні теоретичних знань з основного курсу вищої математики, необхідних під час вивчення інформатики та комп'ютерної техніки, набуття навичок їх практичного застосування до розв'язування конкретних задач; набуття знань та вмінь наукового дослідження в рамках навчально-дослідного завдань; вдосконаленні вмінь та навичок побудови алгоритмів; формуванні вмінь професійного використання методів і засобів опрацювання інформації, СКМ (конкретно MathCAD, Maple), інноваційних методів та методик розв'язування задач.

Висновки

Висновок перший. Пропонується формування змісту курсу вищої математики здійснювати у таких напрямках: підсилення ролі числових методів і їх реалізацію за допомогою СКМ; використання математичного моделювання під час розв'язування прикладних задач; оволодіння студентами математичними змістово-фаховими знаннями, необхідними для аналітичного і числового моделювання практичних інженерних задач, створення власних бібліотек користувача програмних продуктів.

Висновок другий. Згідно із запропонованою методикою навчання, формування системи завдань повинно відповідати типу діяльності, визначеному серед інших – математичному моделюванню природничих, технічних, екологічних та суспільних явищ і процесів. Студенти поглиблюють навички створення математичної моделі, її дослідження за допомогою комп'ютерного моделювання. Комп'ютерне моделювання у математиці переконливо доводить студентам, що математичні конструкції – це моделі реальних відношень. А це змінює відношення студентів до математики, навчання стає осмисленим та продуктивнішим, у них формуються мотиваційні чинники щодо опанування та поглиблення знань із різних галузей науки, зокрема можливістю оволодіння навичками використання СКМ. Студенти поглиблюють уявлення про весь технологічний ланцюг розв'язання задачі. Що стосується прикладного дослідження в галузі математики, студенти навчаються вибору та використанню алгоритмів, методів, прийомів і способів розв'язування математичних задач з використанням комп'ютерних засобів. При цьому розглядаються задачі математичного моделювання за допомогою диференціальних рівнянь у частинних похідних, задачі на побудову регресії, вибору оптимальної регресії, задачі на оцінювання похибок у різних нормованих просторах, та інших типів задач.

Висновок третій. Виділено головні вміня й навички, які можуть бути сформовані у студентів під час інтегрованих занять з вищої математики і елементів числового моделювання: уміння використовувати необхідні програмні засоби в середовищі СКМ; уміння зіставляти результати в різних формах подання (аналітичне й графічне подання); уміння використовувати нові можливості, що надаються комп'ютерними технологіями, заснованими на використанні середовища СКМ; уміння проектувати в середовищі СКМ програмні засоби першого рівня складності для розв'язування навчальних і предметних завдань.

Список літератури

- [1] Ю. Горошко, "Система інформаційного моделювання у підготовці майбутніх учителів математики та інформатики," дис. д-ра пед. наук, Чернігівський нац. пед. ун-т імені Т. Г. Шевченка, Чернігів, 2013.
- [2] Т. П. Кобильник, "Вивчення елементів програмування у системі комп'ютерної математики Maxima," *Молодь і ринок: щомісячний науково-педагогічний журнал*, № 5-6 (28-29), с. 159-163. 2007.
- [3] С. А. Кирилашук, В. І. Клочко, З. В. Бондаренко, "Технологічний підхід до формування математичної компетентності фахівців галузі ІТ," *Modern education, training and upbringing: collective monograph*, International Science Group. Boston: Primedia eLaunch, с. 453-467. 2021. doi: 10.46299/ISG.2021.MONO.PED.I
- [4] З. В. Бондаренко, В. І. Клочко, С. А. Кирилашук, "Інтегративний підхід до формування професійних компетенцій майбутніх інженерів шляхом використання засобів математичного моделювання," *Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми. Зб. наук. пр.*, № 46, с. 114-117. 2016.

- [5] С. О. Семеріков, *Maxima 5.13: довідник користувача*. Ред. Академіка АПН М. І. Жалдака. Київ, 2007.
- [6] І. О. Теплицький, *Елементи комп'ютерного моделювання: навчальний посібник*. Кривий Ріг. КДПУ, 2005.
- [7] Developing 21st Century Competencies in the Mathematics Classroom Yearbook 2016, Association of Mathematics Educators, Edited by: Pee ChoonToh (NTU, Singapore), Berinderjeet Kaur (NTU, Singapore).
- [8] Ministry of Education, Singapore (2010). MOE to enhance learning of 21st century competencies and strengthen art, music and physical education. Available at: www.moe.gov.sg.
- [9] Turner, R., (2010) Exploring mathematical competencies. Available at: <http://research.acer.edu.au/resdev/vol24/iss24/5>.
- [10] І. В. Сітак, В. М. Давиденко, "Організація математичної підготовки майбутніх інженерів у вищому технічному навчальному закладі," *Scientific journal «Virtus»*, № 2, с. 139-142. 2015.
- [11] В. М. Михалевич, *Maple. Комп'ютерна підтримка курсу вищої математики в технічному вузі. Ч. 1. Лінійна й векторна алгебра. Аналітична геометрія*, Вінниця, Україна: ВНТУ, 2004.
- [12] B. A. Alpers, *Framework for Mathematics Curricula in Engineering Education*. A Report of the Mathematics Working Group. Brussels: European Society for Engineering Education, 2013.
- [13] Ю. С. Рамський, К. І. Рамська, "Про роль математики і деякі тенденції розвитку математичної освіти в інформаційному суспільстві," *Науковий часопис НПУ імені М. П. Драгоманова. Серія № 2. Комп'ютерно-орієнтовані системи навчання: зб. наукових праць*: НПУ, № 6 (13), с. 12-16. 2008.
- [14] К. Х. Зеленський, В. М. Ігнатенко, О. П. Коц, *Комп'ютерні методи прикладної математики*, 2002.
- [15] З. В. Бондаренко, С. А. Кирилашук, "Аспекти формування математичної та інформатичної компетентності у майбутніх фахівців з інформаційних технологій," *Міжнародна науково-методична Інтернет-конференція «Проблеми вищої математичної освіти: виклики сучасності*, Вінниця, 2018, с. 343-344.

Стаття надійшла: 18.07.2022.

References

- [1] Yu. Horoshko, "Systema informatsiinoho modeliuвання u pidhotovtsi maibutnix uchyteliv matematyky ta informatyky," *dys. d-ra ped. nauk*, Chernihivskiy nats. ped. un-t imeni T.H. Shevchenka, Chernihiv, 2013 [in Ukrainian].
- [2] T. P. Kobylnyk, "Vyvchennia elementiv prohramuvannia u systemi kompiuternoї matematyky Maxima," *Molod i ryнок: shchomisiachnyi naukoivo-pedahohichnyi zhurnal*, № 5-6 (28-29), s. 159-163. 2007 [in Ukrainian].
- [3] S. A. Kyrylashchuk, V. I. Klochko, Z.V. Bondarenko, "Tekhnolohichnyi pidkhid do formuvannia matematychnoi kompetentnosti fakhivtsiv haluzi IT," *Modern education, training and upbringing: collective monograph*, International Science Group. Boston: Primedia eLaunch, s. 453-467. 2021. doi: 10.46299/ISG.2021.MONO.PED.I [in Ukrainian].
- [4] Z. V. Bondarenko, V. I. Klochko, S. A. Kyrylashchuk, "Intehratyvnyi pidkhid do formuvannia profesiinykh kompetentsii maibutnix inzheneriv shliakhom vykorystannia zasobiv matematychnoho modeliuвання," *Suchasni informatsiini tekhnolohii ta innovatsiini metodyky navchannia u pidhotovtsi fakhivtsiv: metodolohiia, teoriia, dosvid, problemy*. Zb. nauk. pr., № 46, s. 114-117. 2016 [in Ukrainian].
- [5] S. O. Semerikov, *Maxima 5.13: dovidnyk korystuvacha*. Red. Akademika APN M. I. Zhaldaka. Kyiv, 2007 [in Ukrainian].
- [6] І. О. Теплицький, *Elementy kompiuternoho modeliuвання: navchalnyi posibnyk*. Kryvyi Rih. KDPU, 2005 [in Ukrainian].
- [7] Developing 21st Century Competencies in the Mathematics Classroom Yearbook 2016, Association of Mathematics Educators, Edited by: Pee ChoonToh (NTU, Singapore), Berinderjeet Kaur (NTU, Singapore).
- [8] Ministry of Education, Singapore (2010). MOE to enhance learning of 21st century competencies and strengthen art, music and physical education. Available at: www.moe.gov.sg.
- [9] Turner, R., (2010) Exploring mathematical competencies. Available at: <http://research.acer.edu.au/resdev/vol24/iss24/5>.
- [10] І. В. Сітак, В. М. Давиденко, "Orhanizatsiia matematychnoi pidhotovky maibutnix inzheneriv u vyshchomu tekhnichnomu navchalnomu zakladi," *Scientific journal «Virtus»*, № 2, s. 139-142. 2015 [in Ukrainian].
- [11] V. M. Mykhalevych, *Maple. Kompiuterna pidtrymka kursu vyshchoi matematyky v tekhnichnomu vuзі. Ch. 1. Liniina y vektorna alhebra. Analitychna heometriia*, Vinnytsia, Ukraina: VNTU, 2004 [in Ukrainian].

- [12] B. A. Alpers, Framework for Mathematics Curricula in Engineering Education. A Report of the Mathematics Working Group. Brussels: European Society for Engineering Education, 2013.
- [13] Yu. S. Ramskyi, K. I. Ramska, "Pro rol matematyky i deiaki tendentsii rozvytku matematychnoi osvity v informatsiinomu suspilstvi," *Naukovyi chasopys NPU imeni M. P. Drahomanova. Seriya № 2. Komp'uterno-oriientovani systemy navchannia: zb. naukovykh prats: NPU*, № 6 (13), s. 12–16. 2008 [in Ukrainian].
- [14] K. Kh. Zelenskyi, V. M. Ihnatenko, O. P. Kots, *Komp'uterni metody prykladnoi matematyky*, 2002 [in Ukrainian].
- [15] Z. V. Bondarenko, S. A. Kyrylashchuk, "Aspekty formuvannia matematychnoi ta informatychnoi kompetentnosti u maibutnikh fakhivtsiv z informatsiinykh tekhnolohii," *Mizhnarodna naukovo-metodychna Internet-konferentsiia «Problemy vyshchoi matematychnoi osvity: vyklyky suchasnosti, Vinnytsia*, 2018, s. 343-344 [in Ukrainian].

Відомості про авторів

Кириляшук Світлана Анатоліївна – кандидат педагогічних наук, доцент, декан факультету інформаційних технологій та комп'ютерної інженерії.

Бондаренко Злата Василівна – кандидат педагогічних наук, доцент, доцент кафедри вищої математики.

Клочко Віталій Іванович – доктор педагогічних наук, професор, професор кафедри вищої математики.

Кириляшук Тетяна Генадіївна – викладач кафедри комп'ютерних наук.

S. A. Kyrylashchuk, Z. V. Bondarenko, V. I. Klochko, T. G. Kyrylashchuk

NUMERICAL MODELING AS ONE OF THE COMPONENTS OF FUNDAMENTAL MATHEMATICAL TRAINING OF ENGINEERS

Vinnytsia National Technical University, Vinnytsia

УДК 004.4, 004.6

Д. М. Щитов, М. Ф. Мормуль, О. М. Щитов, Л. А. Романчук, Т. А. Чупілко

АУТСОРСИНГ ІТ-ПОСЛУГ В УКРАЇНІ

Університет митної справи та фінансів, Дніпро

Анотація. На основі статистичних даних проведено поточний аналіз ринку ІТ-послуг України, який показав, що існує великий потенціал для розвитку цієї інноваційної сфери діяльності в нашій країні. Здійснено прогнозування обсягів ринку ІТ України на 2022 рік з урахуванням умов нинішнього воєнного стану внаслідок російської агресії. Виявлено фактори, які найбільше впливають на обсяг ринку ІТ. Спрогнозовано позитивну тенденцію обсягів ринку ІТ в Україні у 2021-2022 роках, розглянуті та проаналізовані головні фактори, що впливають на розвиток ринку високотехнологічних послуг. Також сформульовані чинники, які можуть позитивно вплинути на швидше просування сфери ІТ-послуг, що особливо важливо у воєнний та післявоєнний часи, оскільки це єдина галузь, яка майже не зазнала втрат через високу мобільність спеціалістів та низьку капіталомісткість. До того ж, прибутки від розроблених програмних продуктів і сервісів значно перевищують витрати. Наукова новизна дослідження полягає в систематизації науково-практичної основи розвитку сфери високотехнологічних послуг у контексті глобальної економічної інтеграції. Практичне значення отриманих результатів та висновків бачиться у використанні вітчизняними та іншими науковцями, практиками та державними установами матеріалів цієї статті для аналізу розвитку й конкурентоспроможності високотехнологічних послуг України на світовому ринку. Аналіз факторів формування експортного потенціалу високотехнологічних послуг національних економік провідних країн є вкрай актуальним науковим завданням для подальшого дослідження шляхів розвитку аутсорсингу та ІТ-послуг як в Україні, так і у світі, оскільки вони тісно зв'язані між собою.

Ключові слова: офшоринг, ІТ-послуги, ІТ-компанії, аутсорсинг, айтівці, експорт високотехнологічних послуг, ФОП-модель, рейтинг, статистика, мобільність, чинники.

Abstract. On the basis of statistical data, a current analysis of the IT services market of Ukraine was conducted, which showed that there is a great potential for the development of this innovative field of activity in our country. Forecasting of the volumes of the IT market of Ukraine for 2022 was carried out, taking into account the conditions of the current state of war as a result of Russian aggression. The factors that have the greatest influence on the volume of the IT market have been identified. A positive trend in the volumes of the IT market in Ukraine in 2021-2022 is predicted, the main factors affecting the development of the high-tech services market are considered and analyzed. Factors that can have a positive effect on the faster advancement of the field of IT services are also formulated, which is especially important in the war and post-war times, since this is the only industry that almost did not suffer losses due to the high mobility of specialists and low capital intensity. In addition, the profits from the developed software products and services significantly exceed the costs. The scientific novelty of the study consists in the systematization of the scientific and practical basis of the development of the field of high-tech services in the context of global economic integration. The practical significance of the obtained results and conclusions can be seen in the use by domestic and other scientists, practitioners and state institutions of the materials of this article to analyze the development and competitiveness of high-tech services of Ukraine on the world market. The analysis of the factors forming the export potential of high-tech services of the national economies of leading countries is an extremely relevant scientific task for further research into the ways of development of outsourcing and IT services both in Ukraine and in the world, since they are closely interconnected.

Keywords: offshoring, IT services, IT companies, outsourcing, IT workers, export of high-tech services, FOP model, rating, statistics, mobility, factors.

DOI: <https://doi.org/10.31649/1999-9941-2022-55-3-85-90>.

Вступ

В умовах глобалізаційних процесів швидкий розвиток на міжнародному ринку високотехнологічних послуг набув закордонний аутсорсинг – «офшоринг». Світовий досвід використання аутсорсингу показує, що перевагою такої форми організації бізнесу є економія витрат на здійснення непрофільних видів діяльності, а саме: значна економія бюджету; зниження витрат часу; зменшення капітальних вкладень; оптимізація внутрішніх процесів; доступ до передових інновацій; залучення висококваліфікованого персоналу; удосконалення процесів керівництва та контролю, що разом особливо актуально у кризових соціально-економічних умовах. Найбільш перспективним напрямом у наданні ІТ послуг став аутсорсинг ІТ. На нього припадає понад 2/3 усіх контрактів, що укладаються на світовому ринку взагалі. Аутсорсинг ІТ-послуг є важливою частиною економіки України, що активно розвивається за рахунок наявності великої кількості кваліфікованих спеціалістів та низьких витрат на персонал.

Протягом останніх років ринок інформаційних технологій України зростає та розвивається неймовірними темпами. Нині Україна вважається однією з найкращих європейських країн для ведення технологічного бізнесу, адже надає безкрайні можливості для його розвитку. Щоб з'ясувати, з чим пов'язаний такий стрімкий прогрес ІТ-сфери в Україні, пропонується розглянути головні фактори, що впливають на нього, проаналізувати його поточний стан, а також зробити прогноз розвитку ринку ІТ на 2022 рік. Зробити це особливо важливо в умовах, коли на території країни йде війна, і купа інших галузей зазнають значні витрати та руйнацію. За таких умов роль ІТ-галузі, яка більш стабільна та незалежна від зовнішніх умов, неймовірно зростає. В запропонованій роботі використані дані Державної служби статистики України, журналу «Outsourcing Journal», платформи «Опендатабот», Національного банку України, Спільноти програмістів DOU. Проаналізовані статті відомих економістів та айтішників щодо перспектив розвитку ІТ-галузі України після війни.

Актуальність

ІТ-індустрія має стати провідною галуззю України після закінчення війни, бо безперечні переваги ІТ-сфери – висока мобільність спеціалістів та низька капіталомісткість, тобто прибутки від розроблених

продуктів і сервісів значно перевищують витрати. Саме ця галузь може дати швидкі прибутки країні у післявоєнний період, коли інші галузі зазнали значного спаду.

Європейський досвід показує, що для ефективного розвитку галузі аутсорсингу ІТ-послуг необхідною є значна державна підтримка, елементами якої можуть бути: створення програм підтримки малого та середнього бізнесу у сфері високих технологій; удосконалення венчурного інвестування; модернізація системи оподаткування та законодавчого регулювання сфери високих технологій; підвищення ефективності системи освіти ІТ-фахівців; створення програм щодо впровадження ІКТ в державному та приватному секторах; впровадження національної широкосмислової мережі; підтримка стартап-руху. Однією з вагомих конкурентних переваг для України на глобальному ринку є наявність великої кількості талановитих ІТ-фахівців. Основою для такої переваги стала дешевизна робочої сили, українська академічна школа, що дає потужну теоретичну базу підготовки. Через несприятливі умови підприємницького середовища в Україні існує проблема відтоку ІТ-фахівців за кордон. Державна підтримка може бути найбільш ефективною, якщо залучити кращих ІТ-фахівців до співпраці над створенням і впровадженням механізму управління розвитком аутсорсингу ІТ-послуг в Україні. Найбільш важливими, однак, досить нерозвиненими елементами для України є: створення внутрішнього попиту на ІТ-послуги та товари (обсяг закупівель бізнесу і державних закупівель); розвиток інноваційної інфраструктури (базової і високотехнологічної); створення стимулюючих умов для детінізації фінансового капіталу в високоефективній галузі національної економіки (венчурний капітал, кредити); покращення умов ведення бізнесу (податкова система, регулювання трудових відносин). Важливою складовою для розвитку ринку аутсорсингу в Україні є постійна наявність резерву кваліфікованих фахівців. Для постійного поповнення цього резерву слід розвивати і надавати фінансову підтримку системі освіти в Україні.

Мета

Метою роботи є аналіз ступеня зростання ІТ-галузі, виявлення чинників поліпшення аутсорсингу в Україні, визначення важливих складових, за рахунок яких галузь має кращі перспективи для розвитку, зазначення необхідних засобів з боку держави та ІТ-компаній для створення умов, які мають вивести галузь на перше місце у світі.

Задачі

1. Розглянути досягнення у сфері ІТ-послуг у минулому та нинішньому роках: обсяг експорту, податків тощо.
2. Порівняти вітчизняні показники з зарубіжними.
3. Зробити ІТ-прогноз на 2022 рік.
4. Зазначити необхідні засоби з боку держави та ІТ-компаній для створення умов, які мають вивести галузь на перше місце у світі.

Результати

Обсяг експорту в Україні в 2021 році збільшився на 38% та досяг рекордного числа – \$68,24 млрд. Обсяг імпорту також значно виріс: на 36%, до \$73,3 млрд. При цьому частка ІТ у структурі експорту послуг в Україні, згідно з оприлюдненими директором асоціації ІТ Україна Костянтином Васюком даними, за минулий рік зросла на 4,8% і досягла 37% структури експорту (це становить 2,7% у ВВП України), а обсяг українського ІТ-експорту у 2021 році досяг \$6,8 млрд (у 2020 році – \$5 млрд). Відповідно, у 2021 році ІТ-галузь України сплатила 23,5 млрд грн. податків і зборів, що на 36,6% більше, ніж у 2020 році, коли ця сума становила всього 17,2 млрд грн. [1], [2].

Для порівняння: у Польщі частка ІТ послуг в 2021 році становила 1,3% у ВВП країни, у Литві – 1,2%, Словаччині – 1,2%, Угорщині – 1,5%, Естонії – 2,4%, Сербії – 2,8%, Білорусі – 3,3% (див. рис. 1).

Отже, порівняно із сусідніми країнами Україна залишається привабливим хабом, проте кількість ІТ-фахівців на душу населення залишається вкрай низькою – у 2–4 рази меншою, ніж у країнах-сусідах: розмір ринку праці в ІТ в Україні минулого року становив 18,1 млн осіб, у Польщі – 17 млн осіб, Угорщині – 4,7 млн осіб, Білорусі – 4,5 млн осіб, Сербії – 3,2 млн осіб, Словаччині – 2,7 млн осіб, Литві – 1,5 млн осіб, Естонії – 0,7 млн осіб. Водночас кількість спеціалістів в Україні збільшилась з 244 000 до 285 000 у 2021 році, з яких трохи більше 55 тисяч наймані співробітники, а інші співпрацюють із компаніями за ФОП-моделлю.

Число активних ІТ-компаній на українському ринку зросло до 5 тисяч (у деяких по кілька юросіб). А найбільшими партнерами українських айтшників стали США, Британія, Швейцарія, Мальта, Ізраїль, Кіпр та ін.

Згідно з щорічним рейтингом найкращих аутсорсерів світу міжнародної профільної асоціації ІАОР, до цього списку увійшли 12 українських ІТ-компаній, а також міжнародних аутсорсерів із великими офісами розробки в Україні. Найкращі компанії обирались за чотирма критеріями: відгуки клієнтів, отримані компанією сертифікації та нагороди, інноваційність рішень та проекти корпоративної соціальної відповідальності. У порівнянні з минулим роком з рейтингу зникла компанія Program-Ace, проте з'явилися дві нові компанії – Softengi та Computools. Таким чином загалом кількість українських аутсор-

серів у рейтингу збільшилась. Повний список виглядає наступним чином [3]: Ciklum (понад 2800 спеціалістів в Україні, головний офіс у Києві); CompuTools (понад 800 спеціалістів, головні офіси у Києві, Запоріжжі, Кракові (Польща); Eleks (понад 1300 спеціалістів в Україні, головний офіс у Львові); Inforpulse (понад 1800 спеціалістів в Україні, головний офіс у Києві); Innovacs (понад 800 працівників в Україні, США та Європі, головний офіс у Києві); Intellias (понад 1200 спеціалістів в Україні, головний офіс у Львові); Intetics (понад 300 спеціалістів в Україні, головний офіс у США); Miratech (понад 1500 спеціалістів, головний офіс у Києві); N-iX (понад 1500 спеціалістів, головний офіс у Києві); NIX (група компаній з командою більше 2000 спеціалістів, головний офіс у Харкові); Sigma Software (понад 900 спеціалістів, головний офіс у Харкові); Softengi (понад 200 спеціалістів, головний офіс у Києві).

KEY FACTS ABOUT OFFSHORE SOFTWARE DEVELOPMENT IN EASTERN EUROPE

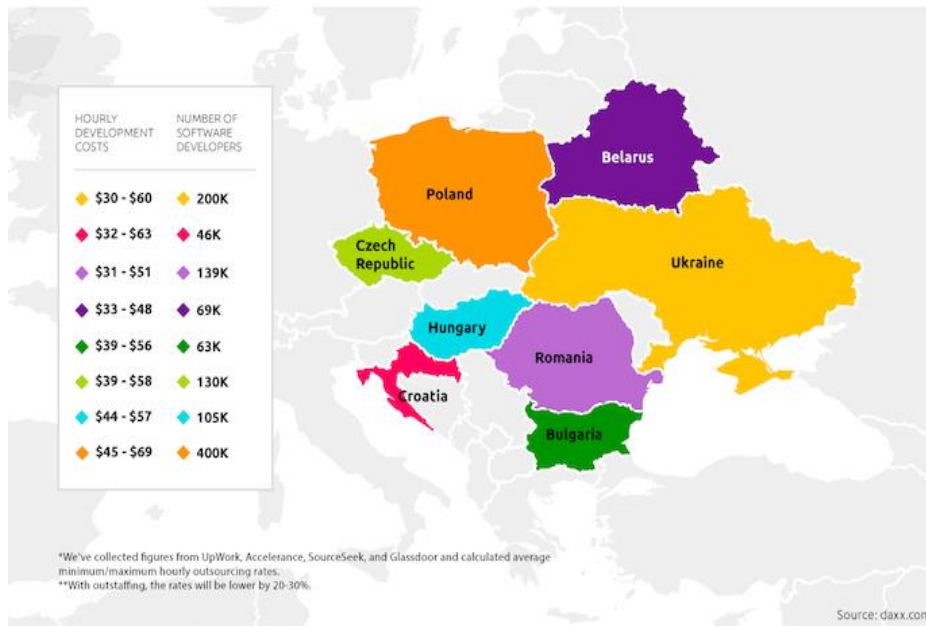


Рисунок 1 – Ключові показники про офшорну розробку програмного забезпечення в Східній Європі (Key facts about offshore software development in Eastern Europe)

У «The 2021 Global Outsourcing 100» входять не тільки постачальники послуг у галузі розробки, а й компанії, що надають аудиторські та консультативні послуги. У 2021 році до рейтинга увійшли три компанії-адвайзери з офісом в Україні: Deloitte (цей бренд об'єднує десятки тисяч висококласних фахівців, які працюють у незалежних фірмах у сфері аудиту, головний офіс у Києві); EY (глобальна організація входить до складу Ernst & Young Global Limited, головний офіс в Україні у Києві); KPMG (понад 240 тисяч спеціалістів у 145 країнах світу, головний офіс в Україні у Києві).

До рейтинга також потрапили компанії Artezio, Avenga та IBA Group, в яких є офіси в Україні. Крім того, Україна посіла перше місце серед країн Східної Європи із привабливості ІТ-аутсорсингу. Згідно звіту нідерландської компанії Daxx восени 2021 року з розробки програмного забезпечення та технологічного консалтингу: Україна посідає одне з перших місць серед таких країн як Хорватія, Румунія, Білорусь, Болгарія, Чехія, Угорщина та Польща за вартістю години розробки програмного забезпечення. В Україні вартість становить \$30 - \$60, у Польщі – \$45 - \$69, в Угорщині – \$44 - \$57, у Чеській Республіці – \$39 - \$58, у Болгарії – \$39 - \$56, у Білорусі – \$33 - \$48, у Румунії – \$31 - \$51, Хорватії – \$32 - \$63. А по кількості ІТ-фахівців серед тих же країн Україна займає друге місце. В Україні кількість ІТ-фахівців становить 200 тис., у Польщі – 400 тис., Угорщині – 105 тис., Чеській Республіці – 130 тис., Болгарії – 63 тис., Білорусі – 69 тис., Румунії – 139 тис., Хорватії – 46 тис. Згідно з даними журналу «Outsourcing Journal», Україна стала країною №1 з ІТ-аутсорсингу в Центральній та Східній Європі, а також увійшла до топ-20 країн у регіоні ЄБСА з офшорингу ІТ-розробки. Тепер Україну розглядають як перспективний технологічний центр із швидко зростаючою ІТ-індустрією, що відповідно привернуло увагу світової ІТ-спільноти, спонукаючи дедалі більше іноземних компаній наймати українських розробників.

Варто зазначити, що Україна відома не лише індустрією ІТ-аутсорсингу, а й наданням офшорних послуг із відкриття R&D-центрів. Значна кількість країн зацікавлена в отриманні цих послуг, але все ж

постійними клієнтами залишаються компанії зі США та Західної Європи. Серед головних факторів, які приваблюють іноземні фірми на український ІТ-ринок, можна виділити наступні:

- велика кількість талановитих ІТ-фахівців;
- вдале розташування;
- низькі податки (5%) й заробітна плата;
- якісна робота;
- схожість на культурному рівні [9, с.212-213].

Офшоринг в Україні є доволі цікавий для підприємств із США, ЄС та інших країн і в основному представлений ІТ-проектами. Офшоринг має наступні переваги: близькість до західних цінностей; висококваліфіковані трудові ресурси; дешеві трудові ресурси; невеликі мовні перепони; розвинута інфраструктура. Більшість фахівці на ринку ІТ-послуг працюють у великих компаніях, але ж малих та середніх компаній більше ніж великих. Основними центрами аутсорсингу в Україні є Київ, Харків, Львів, Дніпро, Одеса, де працює біля 85 % усіх програмістів. Більшість компаній мають офіси у Східній Європі. Доступність кваліфікованої робочої сили є однією з головних причин, чому іноземні компанії використовують аутсорсинг ІТ-послуг в Україні. Завдяки цьому європейські та американські компанії можуть заощаджувати від 40 % до 60 % від загального обсягу витрат на ІТ-послуги [1].

За оцінками експертів, якщо кількість іноземних компаній, що заходять на український ринок, продовжить зростати, тоді місцевий ІТ-ринок стабільно збільшуватиметься на 22-30% щороку, що також вплине і на збільшення кількості технічних фахівців, яких до 2024 року стане вдвічі більше [7].

Цікаві подвиги в галузі ІТ-послуг намітилися і в 2022 році. Попри війну, що розпочалася 24 лютого, ІТ-послуги стали єдиною сферою бізнесу, яка зросла у 2022 році у порівнянні з 2021 роком – на 27%: за 5 місяців цього року частка ІТ у всьому експорті послуг становила 46% проти 37% роком раніше. З початком війни українські айтівці через свою високу мобільність та, як правило, відсутність робочої прив'язки до місця перебування змогли зберегти високий рівень зайнятості в індустрії.

Як говориться у повідомленні платформи для роботи з відкритими даними «Опендатабот», з січня по травень 2022 року весь експорт послуг залишився на рівні відповідного періоду минулого року – 6,8 млрд доларів. Водночас експорт товарів впав на 20% – з 22,5 млрд до 17,9 млрд доларів [4].

Як свідчать дані Національного банку України, ІТ-індустрія України забезпечила рекордні \$2 млрд експортних надходжень в першому кварталі 2022 року, незважаючи на воєнний стан, мобілізацію, вимушену релокацію бізнесу та команд. Аналогічний показник у 2021 році становив \$1,44 млрд. Фактично обсяг ІТ-експорту збільшився на 28%.

Згідно з ними, стрімке зростання ІТ-індустрії продовжувалось до самої війни і в лютому 2022 року досягло найвищого за всю історію українського ІТ-ринку щомісячного показника експорту – \$839 млн, що на 43% більше за аналогічний період 2021 року (\$480 млн) [5].

У нових воєнних реаліях ІТ-індустрії вдалося консолідуватися та швидко переформатуватися завдяки антикризовим бізнес-планам. Більшість компаній зберегли клієнтів та обсяги своїх контрактів. Завдяки цьому галузь залишається фінансово стабільною, забезпечує регулярні валютні надходження до економіки України і державного бюджету та сплачує податки наперед. Так, у березні 2022 року українська ІТ-галузь зберегла 96% обсягу експорту комп'ютерних послуг (\$522 млн) у порівнянні з аналогічним періодом минулого року (\$546 млн). При цьому в березні валютна виручка через війну скоротилася лише на 38% у порівнянні з рекордним показником зростання у лютому. І це при тому, що інші експортні галузі впали на 50-60% [6].

За словами відомого українського економіста, ректора новоствореного Американського університету в Києві Романа Шеремета, ІТ-індустрія має стати провідною галуззю України після закінчення війни, бо безперечні переваги ІТ-сфери – висока мобільність спеціалістів та низька капіталомісткість, тобто прибутки від розроблених продуктів і сервісів значно перевищують витрати [8].

Висновки

1. Для удосконалення розвитку ІТ-сектору в Україні слід підвищувати внутрішній попит на аутсорсинг ІТ-послуг. Для цього необхідно запровадити проект впровадження аутсорсингу ІТ-послуг на українських підприємствах.

2. Можна запропонувати наступний алгоритм переходу підприємства на аутсорсинг: нормалізація бізнес-процесів; аналіз ресурсів, необхідних для аутсорсингу; аналіз ризиків; оцінка ефективності; підготовка критеріїв пошуку постачальників та договору; пошук постачальника через проведення конкурсу; підписання угоди; створення мотивації для постачальника та власного персоналу; поточний контроль за наданими послугами; аналіз ефективності аутсорсингової діяльності; продовження співробітництва або пошук нового постачальника.

3. Для країн з трансформаційною економікою, до яких належить Україна, доцільно долучатися до участі у міжнародних європейських мережевих програмах, формувати власну інституційну основу розвитку високотехнологічних послуг.

4. Для ефективного розвитку експортного потенціалу високотехнологічних послуг України слід долучити досвід країн ЄС, що є актуальним завданням для подальших досліджень.

Список літератури

- [1] "Зовнішня торгівля України послугами у 2021 році," *Державна служба статистики України*. [Електронний ресурс]. Режим доступу: www.ukrstat.gov.ua. Дата звернення: Серп. 12, 2022.
- [2] "Частка ІТ-послуг в експорті у 2021 році становить 27% – дослідження ІТ Ukraine," *Телеком*, с. 2, Янв. 20. 2022. [Електронний ресурс]. Режим доступу: <https://ua.interfax.com.ua/news/telecom/792926.html>. Дата звернення: Серп. 10, 2022.
- [3] В. Некрасов, "Україну назвали лучшей страной для ІТ-аутсорсинга в Восточной Европе," *Економічна правда*, с. 6, Листопад 4. 2021.
- [4] О. Павлиш, "За пять місяців експорт ІТ-послуг попри війну зріс на 27% – Оpendатобот," *Економічна правда*, с. 4, Лип. 6. 2022.
- [5] Л. Шкіль, *ІТ-послуги — єдина галузь бізнесу, що зросла у 2022 році*. [Електронний ресурс]. Режим доступу: <https://ain.ua/2022/07/06/it-poslugy-yedyna-galuz-biznesu-shho-zroslo-u-2022-roczii/>. Дата звернення: Серп. 11, 2022.
- [6] В. Даилченко, *Цього року експорт ІТ-послуг в Україні зріс на 27%*. [Електронний ресурс]. Режим доступу: www.unn.com.ua/uk/news/1984473-tsogo-roku-eksport-it-poslug-v-ukrayini-zris-na-27. Дата звернення: Серп. 1, 2022.
- [7] Д. Овчаренко, *Огляд ІТ-індустрії України: прогноз на 2022 рік*. [Електронний ресурс]. Режим доступу: <https://alcor-bpo.com/uk/your-own-rd-office-news/ukrainian-it-industry-market-overview-for-2022/>. Дата звернення: Черв. 30, 2022.
- [8] О. Стрельников, *Українське ІТ на воєнних рейках – що відбувається з галуззю просто зараз і чого чекати після перемоги*. [Електронний ресурс]. Режим доступу: <https://dou.ua/lenta/articles/ukrainian-it-during-the-war/>. Дата звернення: Черв. 28, 2022.
- [9] Д. Щитов, *Реалізація експортного потенціалу країн ЄС на світовому ринку високотехнологічного бізнесу*, дис. канд. економ. наук., фак-т міжнародних економічних відносин та туристичного бізнесу, нац. ун-т ім. В. Н. Каразіна, Харків, 2014.

Стаття надійшла: 10.09.2022.

References

- [1] "Zovnishnia torhivlia Ukrainy posluhamy u 2021 rotsi," *Derzhavna sluzhba statystyky Ukrainy*. [Online]. Available: www.ukrstat.gov.ua. Accessed on: August 12, 2022 [in Ukrainian].
- [2] "The share of IT services in exports in 2021 is 27% - research by IT Ukraine," *Telekom*, pp. 2, Jan. 20. 2022. [Online]. Available: <https://ua.interfax.com.ua/news/telecom/792926.html>. Accessed on: August 12, 2022 [in Ukrainian].
- [3] V. Nekrasov, "Ukraynu nazvaly luchshei stranoi dlia IT-aoutsorsynha v Vostochnoi Evrope," *Ekonomichna pravda*, pp. 6, Lystopad 4. 2021 [in Russian].
- [4] O. Pavlysh, "Za piat misiatsiv eksport IT-posluh popry viinu zris na 27% – Opendatobot," *Ekonomichna pravda*, pp. 4, Lyp. 6. 2022 [in Ukrainian].
- [5] L. Shkil, *IT-posluhy – yedyna haluz biznesu, shcho zroslo u 2022 rotsi*. [Online]. Available: <https://ain.ua/2022/07/06/it-poslugy-yedyna-galuz-biznesu-shho-zroslo-u-2022-roczii/>. Accessed on: August 12, 2022 [in Ukrainian].
- [6] V. Daylchenko, *Tsoho roku eksport IT-posluh v Ukraini zris na 27%*. [Online]. Available: www.unn.com.ua/uk/news/1984473-tsogo-roku-eksport-it-poslug-v-ukrayini-zris-na-27. Accessed on: August 12, 2022 [in Ukrainian].
- [7] D. Ovcharenko, *Ohliad IT-industrii Ukrainy: prohnaz na 2022 rik*. [Online]. Available: <https://alcor-bpo.com/uk/your-own-rd-office-news/ukrainian-it-industry-market-overview-for-2022/>. Accessed on: August 12, 2022 [in Ukrainian].
- [8] O. Strelnikov, *Ukrainske IT na voiennykh reikakh — shcho vidbuvaietsia z haluzziu prosto zaraz I choho chekaty pislia peremohy*. [Online]. Available: <https://dou.ua/lenta/articles/ukrainian-it-during-the-war/>. Accessed on: August 12, 2022 [in Ukrainian].
- [9] D. Shchytov, *Realizatsiia eksportnoho potentsialu krain YeS na svitovomu rynku vysokotekhnolohichnykh posluh*, dys. kand. ekonom. nauk., fak-t mizhnarodnykh ekonomichnykh vidnosyn ta turystychnoho biznesu, nats. un-t im. V. N. Karazina, Kharkiv, Ukraine, 2014 [in Ukrainian].

Відомості про авторів

Щитов Дмитро Миколайович – кандидат економічних наук.

Мормуль Микола Федорович – кандидат технічних наук, доцент.

Щитов Олександр Миколайович – кандидат фізико-математичних наук, доцент.

Романчук Любов Анатоліївна – кандидат філологічних наук.

Чупілко Тетяна Анатоліївна – кандидат технічних наук, доцент.

D. M. Shchytyov, M. F. Mormul, O. M. Shchytyov, L. A. Romanchuk, T. A. Chupilko

IT SERVICES OUTSOURCING IN UKRAINE

University of Customs and Finance, Dnipro

ДО ВІДОМА АВТОРІВ

Найновіші правила оформлення і подання статей знаходяться на сайті журналу

<http://itce.vntu.edu.ua/>