https://orcid.org/0000-0002-5523-6120
https://orcid.org/0009-0006-3219-450X

УДК 004.93

С.В. ГОЛУБ[*], Р.Г. НЕМОВ[*]

ПАРАМЕТРИЧНА ОПТИМІЗАЦІЯ ФОРМУВАННЯ МАСИВУ ВХІДНИХ ДАНИХ НА ОСНОВІ СИМВОЛЬНИХ КРОС-ЗВ'ЯЗКІВ

[*]Черкаський державний технологічний університет, м. Черкаси, Україна

Анотація. *Розглянуто задачу параметричної оптимізації формування масиву вхідних даних (МВД) для атрибуції авторства програмних кодів, зокрема розрізнення кодів, написаних людиною та згенерованих великими мовними моделями. Розвинуто тип мовнонезалежних ознак — символьні крос-зв'язки між словами, що формуються як упорядковані пари суфіксів і префіксів слів у вікні. Завдяки врахуванню всіх пар слів, а не лише сусідніх, ознаки нечутливі до локальних переставлень фрагментів коду. Якість класифікації оцінюється на рівні вікон і на рівні авторів за мажоритарним голосуванням із використанням турніру 39 архітектур моделей машинного навчання. Проведено повнофакторний обчислювальний експеримент зі 168 дослідів для 12 авторів (10 людей і дві генеративні моделі) у чотирьох мовах програмування. Досліджено вплив розміру вікна, межі інформативної достатності, режиму фільтрації та рангу крос-зв'язків на частку правильно класифікованих об'єктів і розмір словника ознак. Установлено, що крос-зв'язки забезпечують приріст у 37 % конфігурацій, найбільший внесок (до +7,56 %) спостерігається для вікна 2000 символів — точки максимального внеску додаткових ознак, але не максимальної абсолютної якості. Парето-оптимальною є конфігурація з вікном 500 символів, межею три та усередненим режимом фільтрації, що досягає 99,37 %, коли розмір словника лише 310 ознак; режим максимуму підвищує якість ціною зростання словника майже у 20 разів. Визначено межі застосовності методу й умови, за яких крос-зв'язки знижують частку правильно класифікованих об'єктів, зокрема внаслідок ефекту стелі.*
Ключові слова: *атрибуція авторства, програмний код, рефакторинг коду, масив вхідних даних, символьні крос-зв'язки, межа інформативної достатності, програмування мікросервісів, параметрична оптимізація, моніторинг та аналіз даних, машинне навчання, розробка web-додатків на java, генеративні моделі, вебпрограмування, серверне програмування.*

Abstract. *The problem of parametric optimization of the input data array (IDA) construction process for source code authorship attribution is considered, in particular for distinguishing between human-written and large-language-model-generated code. A language-independent feature type has been further developed, namely character-level cross-links between words, represented as ordered pairs of word prefixes and suffixes within a sliding window. By considering all pairs of words, rather than only adjacent ones, these features are insensitive to local rearrangements of code fragments. The constructed input data array is evaluated using a tournament of 39 machine learning model architectures, and the classification quality is assessed both at the level of individual windows and at the author level by majority voting. A full-factorial computational experiment of 168 trials has been carried out for 12 authors (10 humans and 2 generative models) in four programming languages. The effects of window size, the informational sufficiency threshold, the filtering mode, and the cross-link rank on the proportion of correctly classified objects and the size of the feature dictionary have been studied. It has been established that cross-links improve performance in 37 % of the evaluated configurations. The greatest improvement (up to +7.56 %) is observed for a window size of 2,000 characters, which represents the point of maximum contribution of the additional features rather than the point of highest overall performance. The Pareto-optimal configuration uses a 500-character window, a threshold of three and the averaging filtering mode, achieving 99,37 % of correctly classified objects with a dictionary of only 310 features, whereas the maximum mode increases quality at the cost of a nearly twentyfold growth of the dictionary. The limits of applicability of the method and the conditions*

ISSN 1028-9763. Математичні машини і системи. 2026. № 3

under which cross-links reduce classification quality are determined, in particular for authors with a high baseline due to the ceiling effect.
Keywords: *attribution of authorship, program code, code refactoring, input data array, symbolic cross-links, informative sufficiency limit, microservices programming, parametric optimization, data monitoring and analysis, machine learning, web application development in Java, generative models, web programming, server programming.*

DOI: 10.34121/1028-9763-2026-3-72-81

1. Вступ

Поширення великих мовних моделей, здатних генерувати синтаксично коректний програмний код, загострило проблему атрибуції авторства програмних кодів — встановлення того, ким створено фрагмент коду: конкретною людиною чи генеративною моделлю штучного інтелекту. Ця задача є важливою для забезпечення академічної доброчесності, контролю якості програмного забезпечення, розслідування інцидентів інформаційної безпеки та захисту інтелектуальної власності. На відміну від традиційної атрибуції авторства, що спирається на лексичні та синтаксичні особливості мови програмування, потреба у мовнонезалежних ознаках зростає через багатомовність сучасних програмних проєктів та здатність генеративних моделей імітувати стиль різних мов.

У межах методу формування масиву вхідних даних (МВД) багаторівневого інтелектуального моніторингу [1] кожен об’єкт класифікації описується масивом чисельних ознак, дібраних за критерієм інформативності. Якість класифікатора, що синтезується машинним навчанням на такому масиві, суттєво залежить від параметрів його формування. Тому актуальною є задача параметричної оптимізації процесу формування МВД для нового типу мовнонезалежних ознак — символьних крос-зв’язків між словами.

Практична значущість задачі зумовлена кількома чинниками. По-перше, зростає потреба в інструментах академічної доброчесності, здатних виявляти код, згенерований великими мовними моделями, у навчальних та кваліфікаційних роботах. По-друге, у промисловій розробці програмного забезпечення атрибуція авторства є складником контролю якості та аудиту коду. По-третє, мовнонезалежні ознаки, нечутливі до конкретного синтаксису, дозволяють будувати єдиний класифікатор для багатомовних проєктів без перенавчання під кожну мову програмування. Розв’язання задачі параметричної оптимізації наближає метод символьних крос-зв’язків до практичного застосування у зазначених сценаріях.

2. Аналіз останніх досліджень і публікацій

Класична атрибуція авторства програмних кодів започаткована роботами, в яких застосовано лексичні, синтаксичні та структурні ознаки у поєднанні з методами машинного навчання [2, 3]. У роботі [2] показано, що за стилем коду — відступами, способом іменування, частотами службових слів та структурою синтаксичного дерева — можливо деанонімізувати програміста з високою точністю навіть на великих вибірках. У дослідженні [3] цей підхід поширено на масштабні та мовнонезалежні постановки, де ознаки формуються без прив’язки до конкретної мови програмування, що наближує задачу до умов реальних багатомовних репозиторіїв.

Подальший розвиток пов’язаний із застосуванням глибоких нейронних мереж та методів підвищення стійкості ознак до навмисних спотворень стилю. У підході RoPGen [4] стійкість моделей атрибуції до цілеспрямованих змін стилю забезпечується поєднанням навчання на доповнених даних та градієнтних перетворень, що ускладнює зловмисне приховування авторства. Спільною рисою наведених підходів є залежність від структурних ознак конкретної мови, що обмежує їх застосовність за умов багатомовності та автоматичної генерації коду.

З поширенням генеративних моделей з'явилися дослідження, присвячені відрізненню згенерованого коду від написаного людиною [5–7]. У роботі [5] оцінено наявні детектори згенерованого коду й показано, що їх надійність суттєво залежить від мови та задачі. У дослідженні [6] запропоновано класифікатор GPTSniffer на основі моделі CodeBERT для виявлення коду, згенерованого ChatGPT. У роботі [7] проаналізовано стилістичні відмінності коду великих мовних моделей та запропоновано техніки виявлення джерела коду. Ці дослідження підтверджують актуальність задачі, проте здебільшого спираються на попередньо навчені мовні моделі коду, що потребують значних обчислювальних ресурсів і прив'язані до мови навчання.

Основою цього дослідження є методи багаторівневого інтелектуального моніторингу та формування МВД, розвинуті у працях наукової школи [1]. У дисертаційному дослідженні [1] обґрунтовано критерій інформативності ознаки як імовірність її використання у вікні тексту, запроваджено межу інформативної достатності та показано, що за оптимізації розміру вікна частка правильно класифікованих текстів сягає 98–100 % для повідомлень обсягом до 500 знаків. Принципово, що у [1] доведено: закономірності зміни кількості правильно класифікованих об'єктів за зміни межі інформативної достатності для вікон різної довжини є різними, а отже, під час побудови кожного наступного класифікатора потрібно розв'язувати задачу параметричної оптимізації. Синтез поліноміальних моделей у даному підході спирається на метод групового урахування аргументів [8].

У попередній роботі авторів [9] метод формування МВД уперше поширено з природномовних текстів на програмні коди та запропоновано тип ознак на основі символьних крос-зв'язків між словами, що показав відносний приріст кількості правильно класифікованих об'єктів. Проте параметри формування МВД у [9] було зафіксовано на основі обмеженого набору серій, без систематичного дослідження простору параметрів.

3. Формування завдання досліджень

Незважаючи на підтверджену працездатність символьних крос-зв'язків, залишається невирішеною задача систематичної параметричної оптимізації їх формування. Зокрема, не досліджено спільний вплив розміру вікна, межі інформативної достатності, режиму фільтрації ознак та рангу крос-зв'язків на частку правильно класифікованих об'єктів та на розмір словника ознак, що визначає обчислювальну складність. Не визначено також межі застосовності методу — умови, за яких символьні крос-зв'язки знижують частку правильно класифікованих об'єктів. Розв'язання цих питань потребує повнофакторного обчислювального експерименту.

На сьогодні залишається недостатньо вивченим процес досягнення компромісу між часткою правильно класифікованих об'єктів та розміром словника ознак. Збільшення кількості ознак потенційно підвищує селективність моделі, проте водночас зростають розмірність масиву вхідних даних, ризик перенавчання та час синтезу класифікаторів. Тому параметри формування масиву слід оцінювати не лише за часткою правильно класифікованих об'єктів, а й за відповідною обчислювальною ціною, тобто розв'язувати задачу багатокритеріальної оптимізації. Саме такий підхід застосовано у цьому дослідженні через побудову Парето-фронту у просторі «розмір словника — частка правильно класифікованих об'єктів».

Ця робота проводилась із метою підвищення кількості правильно класифікованих об'єктів у задачі атрибуції програмних кодів шляхом параметричної оптимізації формування масиву вхідних даних на основі символьних крос-зв'язків. Для досягнення мети поставлено такі завдання: формалізувати ознаку символьного крос-зв'язку та параметри її формування; провести повнофакторний обчислювальний експеримент; дослідити залежність якості класифікації та розміру словника від параметрів; визначити Парето-оптимальну конфігурацію та межі застосовності методу.

4. Формалізація методу

4.1. Формування ознак

Текст програмного коду розбивається на вікна по N символів без перекриття. У межах вікна виділяється множина слів за роздільниками, після чого будуються всі впорядковані пари різних слів. Для пари (w_i, w_j) символьний крос-зв'язок рангу k утворюється поєднанням суфікса слова w_i завдовжки k символів та префікса слова w_j завдовжки k символів. Частотою ознаки у вікні є кількість пар, що дали відповідний ключ. Оскільки враховуються всі пари слів у вікні, а не лише сусідні, ознаки нечутливі до локальних переставлень фрагментів коду (порядку імпортів, оголошень тощо), що є їх важливою властивістю стійкості.

Наприклад: для слів getValue та setName крос-зв'язок рангу 2 утворює ключ із суфікса «ue» першого слова та префікса «se» другого. Якщо у вікні міститься M слів, кількість упорядкованих пар дорівнює $M \cdot (M-1)$, що забезпечує значно щільніше покриття стилю автора порівняно з урахуванням лише сусідніх слів. Розширений словник формується для рангів $k = 1, 2, 3$ одночасно. За результатами експерименту середня кількість ознак крос-зв'язків у словнику розподілилася за рангами приблизно як 694, 2451 та 2740 для $k = 1, 2$ і 3 відповідно, тобто ознаки вищих рангів є численнішими, але водночас рідковживанішими.

4.2. Фільтрація за межею інформативної достатності

Межа інформативної достатності T задається кількісно як мінімальна частота появи ознаки у вікні, тобто кількість пар слів, що дали відповідний ключ; вона не є відсотковою величиною. Дібрана ознака зараховується до словника, якщо вона задовольняє межу T, обчислену виключно за вікнами «свого» автора. У режимі усереднення (AVG) ознака проходить, якщо її середня частота за всіма вікнами автора не менша за T; у режимі максимуму (MAX) — якщо хоча б в одному вікні частота не менша за T. Базовий словник (без крос-зв'язків) формується зі звичайних слів-токенів, а розширений додатково містить ознаки-крос-зв'язки рангів 1, 2 та 3. Порівняння класифікаторів, синтезованих за базовим і розширеним словниками, дозволяє кількісно оцінити внесок символьних крос-зв'язків у частку правильно класифікованих об'єктів.

4.3. Синтез класифікаторів

Сформований масив вхідних даних подається на турнір із 39 архітектур моделей машинного навчання з фіксованими гіперпараметрами (повноз'єднані мережі різної глибини, згорткові та рекурентні мережі, мережі з механізмом уваги, автокодувальники та поліноміальні моделі за методом групового урахування аргументів). Якість оцінюється двома показниками: часткою правильно класифікованих окремих вікон (на рівні вікон) та часткою правильно класифікованих авторів за мажоритарним голосуванням вікон (на рівні класів). Відносний приріст обчислюється як $\Delta P = (n\_крос - n\_база) / n\_база\ 100\ \%$, де n — кількість правильно класифікованих об'єктів.

5. Експериментальне дослідження

5.1. Умови експерименту

Проведено 168 дослідів: 12 авторів, кожен по черзі розглядався як «свій» проти решти 11, у 14 конфігураціях параметрів. Множину авторів утворюють 10 людей та дві генеративні моделі (ChatGPT та Claude Code); коди подано чотирма мовами програмування (Java, JavaScript, TypeScript, Python). Конфігурації охоплюють розмір вікна N {500, 1000, 2000} символів, межу T {3, 5, 8}, режими AVG і MAX та ранги крос-зв'язків. Загальний час обчислень склав близько 13 годин без помилок виконання.

5.2. Програмна реалізація

Обчислювальний експеримент виконано засобами гібридного програмного комплексу. Наукове ядро — препроцесинг коду, формування масиву вхідних даних, обчислення символьних крос-зв'язків та фільтрацію за межею інформативної достатності — реалізовано мовою Java, що забезпечує продуктивну обробку великих обсягів тексту. Етап машинного навчання моделей-класифікаторів реалізовано стандартними засобами на основі бібліотеки глибокого навчання. Обмін даними між компонентами відбувається через бінарний формат числових масивів: ядро серіалізує матрицю ознак розмірності «кількість вікон × кількість ознак» разом із вектором міток класів, а компонент навчання повертає показники якості класифікації. Запуск дослідів автоматизовано через інтерфейс командного рядка з параметрами розміру вікна, межі, режиму фільтрації, набору рангів та переліку каталогів «свого» й «чужих» авторів, що уможливило відтворюваність і пакетне виконання всіх 168 конфігурацій. Обчислення проводилися на центральному процесорі з обмеженою паралельністю, час одного запуску становив від трьох до десяти хвилин залежно від розміру словника.

5.3. Внесок крос-зв'язків

Символьні крос-зв'язки збільшили кількість правильно класифікованих об'єктів на рівні класів у 62 зі 168 конфігурацій (37 %); у 69 конфігураціях (41 %) показник не змінився, а у 37 (22 %) — знизився. Середній відносний приріст склав +0,40 %, максимальний +7,56 %, мінімальний — 4,20 %. Розподіл приросту є асиметричним: позитивні значення трапляються частіше та мають більшу амплітуду, ніж негативні, що свідчить про переважно конструктивний внесок ознак за умови вдалого вибору параметрів. Значна частка конфігурацій без зміни показника пояснюється ефектом стелі для авторів, чий базовий рівень класифікації вже близький до 100 %. На рівні окремих вікон, на відміну від рівня класів, приріст спостерігався частіше, проте мажоритарне голосування вікон згладжує цю різницю, тому ключовим показником для практичних висновків є якість на рівні класів. У табл. 1 наведено найкращу за приростом конфігурацію для кожного автора.

Таблиця 1 — Найкраща за відносним приростом конфігурація для кожного автора

Автор / модель	Конфігурація	База, %	Крос, %	ΔP, %
VladShatskiy	N2000_T5_AVG	92,44	100,00	+7,56
ChatGPT	N2000_T3_AVG	94,12	100,00	+5,88
JoshuaBloch	N1000_T8_MAX	93,28	98,32	+5,04
TarasMaichuk	N500_T5_AVG	91,60	96,64	+5,04
DmytroDanylyk	N2000_T3_AVG	94,96	99,16	+4,20
SergheiIakovlev	N2000_T3_AVG	94,12	98,32	+4,20
VladKravets	N2000_T5_AVG	95,80	100,00	+4,20
PaulMiller	N2000_T3_AVG	96,64	100,00	+3,36
DmytroSemenov	N2000_T8_AVG	96,64	99,16	+2,52
MaxKotliar	N1000_T3_AVG	96,64	99,16	+2,52
VolodymyrAgafonkin	N2000_T8_AVG	96,64	99,16	+2,52
ClaudeCode	N1000_T8_AVG	99,16	100,00	+0,84

Найбільший приріст отримано для автора з виразним індивідуальним стилем іменування (+7,56 %) та для моделі ChatGPT (+5,88 %), що свідчить про здатність крос-зв'язків відокремлювати згенерований код. Натомість для моделі Claude Code базовий показник уже становив близько 99 %, тому приріст обмежений ефектом стелі (+0,84 %).

5.4. Вплив розміру вікна

Залежність якості класифікації та розміру словника від розміру вікна наведено на рис. 1 та у табл. 2. За абсолютною часткою правильно класифікованих об'єктів найкращі результати досягаються за вікон 500–1000 символів, тоді як для вікна, що містить 2000 символів, усереднена частка правильно класифікованих об'єктів дещо знижується. Водночас саме за вікна 2000 символів спостерігається найбільший відносний приріст від крос-зв'язків (+1,14 %), оскільки збільшення вікна зменшує кількість спостережень і погіршує базовий показник, залишаючи більший простір для внеску додаткових ознак. Отже, вікно 2000 символів є точкою максимального внеску крос-зв'язків, але не максимальної абсолютної якості. Збільшення вікна супроводжується зростанням словника ознак на порядок.

Таблиця 2 — Усереднені показники залежно від розміру вікна (режим AVG, $k=1-3$)

Розмір вікна N	База P_a, %	Крос P_a, %	Сер. ΔP_max, %	Сер. |D|
500	93,45	94,36	+0,54	187
1000	93,61	94,51	+0,14	737
2000	92,83	93,88	+1,14	2420

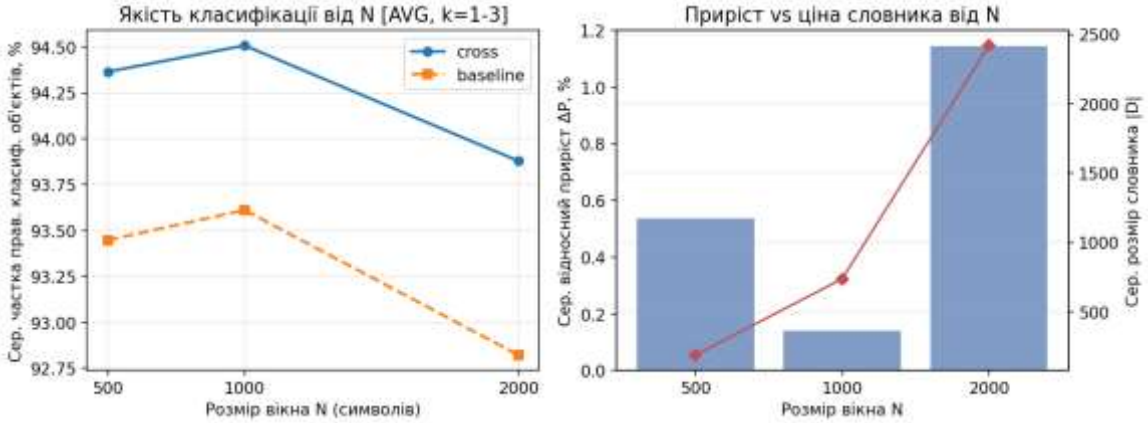

Рисунок 1 — Залежність якості класифікації та розміру словника від розміру вікна N (режим AVG, $k=1-3$)

5.5. Вплив межі інформативної достатності

Зі збільшенням межі T словник ознак скорочується, оскільки відсіюються рідковживані ознаки (рис. 2, табл. 3). Найвища усереднена частка правильно класифікованих об'єктів досягається за $T=3$, тоді як за $T=8$ спостерігається помірне зниження якості за суттєвого скорочення словника — приблизно втричі порівняно з $T=3$. Це підтверджує висновок [1] про необхідність окремої оптимізації межі інформативної достатності для кожної задачі та свідчить про наявність компромісу між повнотою словника й обчислювальною складністю.

Таблиця 3 — Усереднені показники залежно від межі інформативної достатності (режим AVG, $k=1-3$)

Межа T	База P_a, %	Крос P_a, %	Сер. ΔP_max, %	Сер. |D|
3	94,15	94,78	+0,61	1731
5	93,22	94,23	+0,75	1014
8	92,50	93,74	+0,47	600

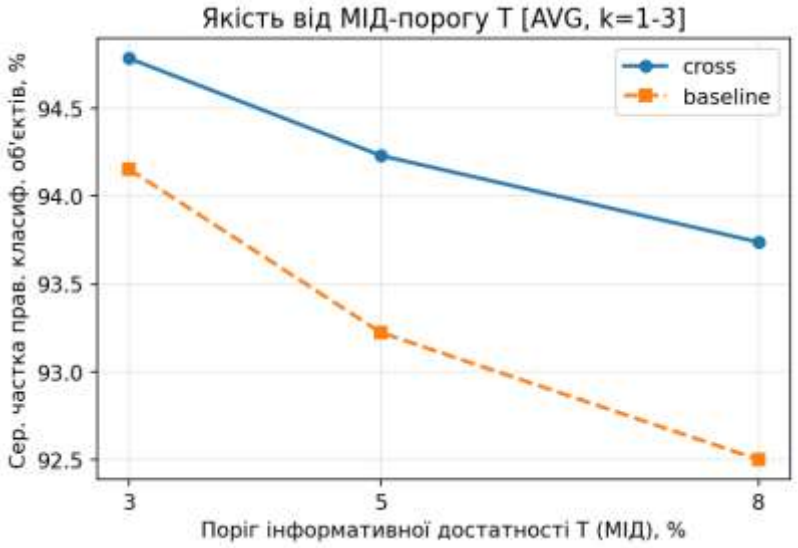

Рисунок 2 — Залежність якості класифікації від межі інформативної достатності T (режим AVG, $k = 1-3$)

5.6. Вплив режиму фільтрації

Порівняння режимів AVG і MAX (рис. 3, табл. 4) показує, що режим MAX забезпечує дещо вищу частку правильно класифікованих об’єктів, проте ціною зростання словника ознак приблизно у двадцять разів. Причина полягає в тому, що у режимі максимуму до словника потрапляє будь-яка ознака, що хоча б в одному вікні автора досягла межі, тоді як режим усереднення відсіює ознаки, нестабільні за всіма вікнами. Таке збільшення розмірності масиву вхідних даних суттєво підвищує обчислювальну складність синтезу моделей за майже незмінної якості на рівні класів. Отже, з погляду співвідношення якості та складності режим усереднення є практично доцільнішим.

Таблиця 4 — Порівняння режимів фільтрації AVG і MAX ($N = 1000$, $k = 1-3$)

Режим	База P_b, %	Крос P_b, %	Крос P_a, %	Сер. |D|
AVG (усереднення)	98,09	98,23	94,51	737
MAX (максимум)	99,25	99,16	95,46	21905

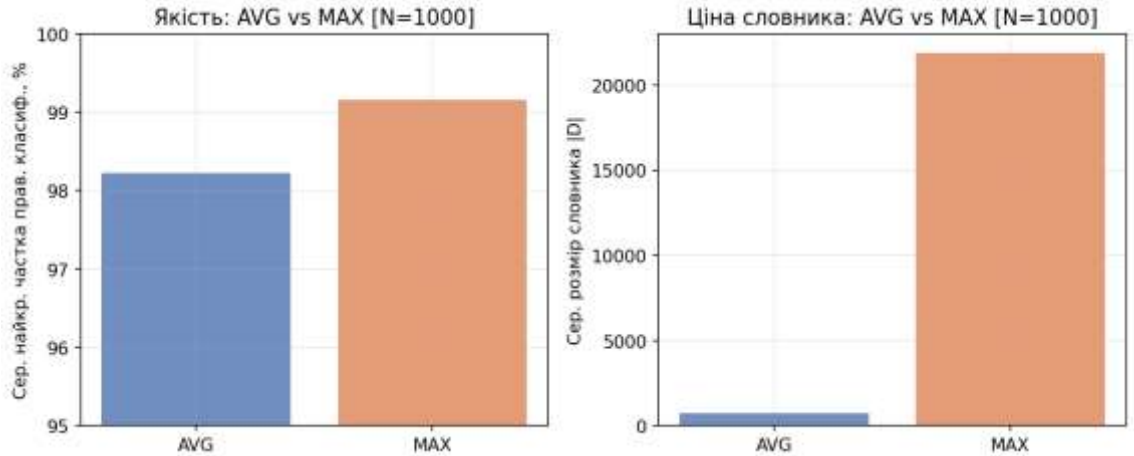

Рисунок 3 — Порівняння режимів фільтрації AVG і MAX за якістю та розміром словника ($N = 1000$, $k = 1-3$)

5.7. Вплив рангу крос-зв'язків

Для опорної конфігурації $N=1000$, $T=3$, режим AVG досліджено вплив рангу крос-зв'язків шляхом послідовного додавання ознак рангів 1, 2 та 3 (табл. 5). Розмір словника у цьому зрізі практично не змінюється, оскільки його переважно визначає базовий набір слів-токенів, тоді як ознаки крос-зв'язки різних рангів становлять його меншу частину. Найвищу якість на рівні класів забезпечує поєднання рангів 1 і 2, тоді як додавання рангу 3 не дає додаткового виграшу й навіть незначно знижує показник через зростання частки рідковживаних ознак. Відмінності між варіантами є невеликими, що не дозволяє зробити сильний висновок на основі однієї опорної точки; це питання потребує окремого дослідження на ширшому наборі конфігурацій.

Таблиця 5 — Вплив рангу крос-зв'язків ($N=1000$, $T=3$, режим AVG)

Ранги k	Крос P_b, %	Крос P_a, %	Сер. ΔP_max, %	Сер. |D|
k = 1	98,67	94,93	+0,07	1174
k = 1, 2	98,81	94,87	+0,28	1174
k = 1, 2, 3	98,53	94,88	+0,21	1174

6. Обговорення результатів

6.1. Парето-оптимальна конфігурація

Сукупний аналіз співвідношення якості класифікації та розміру словника наведено на рис. 4 у вигляді Парето-фронту. Фронт утворюють конфігурації з вікном 500 символів у режимі усереднення; усі конфігурації з вікном 2000 символів лежать під фронтом, тобто є строго гіршими за обома критеріями. Конфігурація з вікном 500 символів, межею три та режимом усереднення досягає 99,37 % правильно класифікованих об'єктів, коли розмір словника лише 310 ознак, що практично не поступається найкращій конфігурації режиму MAX (99,51 %) для словника обсягом понад 20 000 ознак. Отже, Парето-оптимальною є конфігурація $N=500$, $T=3$, режим AVG, ранги $k=1-3$.

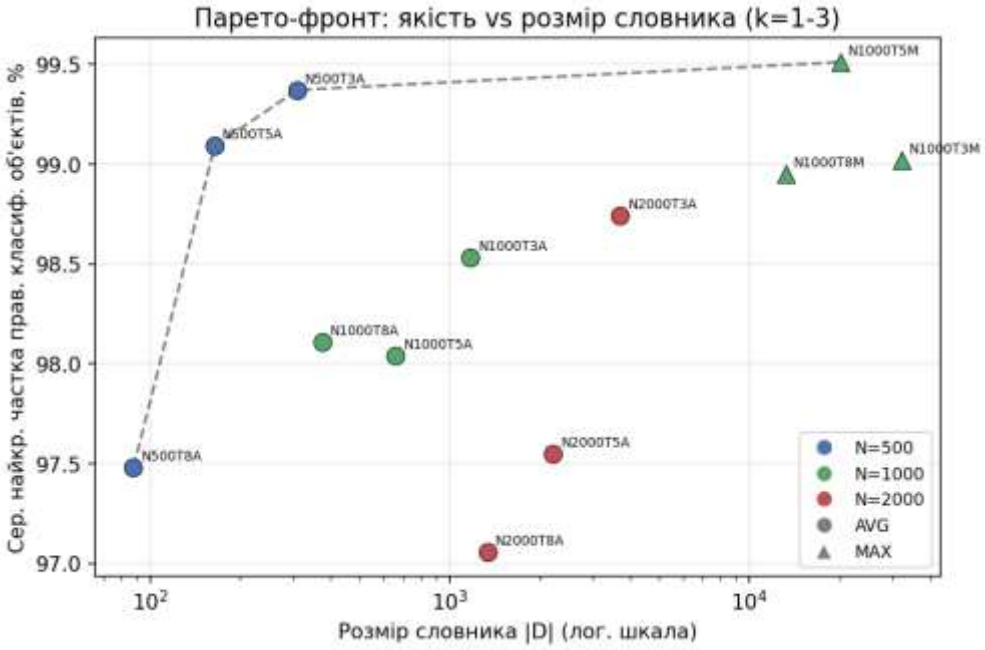

Рисунок 4 — Парето-фронт співвідношення якості класифікації та розміру словника ($k=1-3$)

6.2. Межі застосовності

Зниження якості класифікації від крос-зв'язків у 22 % конфігурацій спостерігається переважно для авторів із високим базовим показником (ефект стелі) та за великих значень межі T у поєднанні з малим вікном, коли словник крос-зв'язків стає надто розрідженим. Отже, символьні крос-зв'язки доцільно застосовувати як додатковий інструмент для авторів зі специфічним стилем іменування та для відрізнення згенерованого коду, а не як універсальний засіб підвищення якості.

6.3. Відрізнення згенерованого коду

Окремий інтерес становить поведінка методу на кодах генеративних моделей. Для двох моделей середній відносний приріст склав +0,63 % за середньої частки правильно класифікованих об'єктів 99,02 %, тоді як для людей — +0,39 % за 98,34 %. Найвиразніший ефект отримано для моделі ChatGPT (+5,88 %): її код добре відокремлюється крос-зв'язками, що свідчить про наявність стійких символьних закономірностей у згенерованому тексті. Натомість для моделі Claude Code базовий показник уже сягав близько 99 %, тому простір для приросту обмежений ефектом стелі. Це підтверджує, що символьні крос-зв'язки є інформативними для виявлення згенерованого коду, а їх внесок залежить від того, наскільки виразним є стиль конкретного джерела.

7. Висновки

Проведено параметричну оптимізацію формування масиву вхідних даних на основі символьних крос-зв'язків для атрибуції авторства програмних кодів за результатами повнофакторного обчислювального експерименту зі 168 дослідів. Установлено, що символьні крос-зв'язки збільшують кількість правильно класифікованих об'єктів у 37 % конфігурацій, причому найбільший внесок припадає на розмір вікна 2000 символів, який є точкою максимального внеску додаткових ознак, але не максимальної абсолютної якості класифікації.

За критерієм співвідношення якості класифікації та обчислювальної складності Парето-оптимальною визначено конфігурацію з вікном 500 символів, межею інформативної достатності три та режимом усереднення, яка досягає 99,37 % правильно класифікованих об'єктів, коли розмір словника лише 310 ознак. Підтверджено висновок попередніх досліджень школи С.В. Голуба про необхідність окремої параметричної оптимізації для кожної задачі та визначено межі застосовності методу.

Зведено вплив кожного параметра формування масиву вхідних даних. Зменшення розміру вікна до 500–1000 символів підвищує абсолютну частку правильно класифікованих об'єктів та скорочує словник; зменшення межі інформативної достатності до трьох підвищує якість ціною збільшення словника; режим усереднення забезпечує практично таку саму якість на рівні класів, що й режим максимуму, але при цьому майже у 20 разів зменшуючи обсяг словника. Поєднання рангів крос-зв'язків 1 і 2 є достатнім, а додавання рангу 3 не дає виграшу. Найбільший внесок крос-зв'язків зафіксовано для авторів зі специфічним стилем іменування та для коду, згенерованого моделлю ChatGPT, що підтверджує придатність методу для відрізнення згенерованого коду.

Перспективами подальших досліджень є розширення множини авторів та генеративних моделей, дослідження крос-зв'язків вищих рангів, а також автоматизація вибору параметрів формування масиву вхідних даних за властивостями класів.

СПИСОК ДЖЕРЕЛ

1. Голуб М.С. Формування масиву чисельних ознак для класифікації україномовних текстів в інформаційній технології інтелектуального моніторингу: дис. канд. техн. наук: 05.13.06. Черкаси: ЧДТУ, 2018. 157 с.

2. Caliskan-Islam A., Harang R., Liu A. et al. De-anonymizing programmers via code stylometry. Proceedings of the 24th USENIX Security Symposium. Washington, 2015. P. 255–270.
3. Abuhamad M., AbuHmed T., Mohaisen A., Nyang D. Large-scale and language-oblivious code authorship identification. *Proc. of the 2018 ACM SIGSAC. Conference on Computer and Communications Security*. Toronto, 2018. P. 101–114.
4. Li Z., Chen G.Q., Chen C. et al. Towards robustness of deep program processing models — detection, estimation and enhancement (RoPGen). *Proc. of the 44th International Conference on Software Engineering (ICSE)*. Pittsburgh, 2022. P. 1097–1108.
5. Pan W.H., Chok M.J., Wong J.L.S. et al. Assessing AI detectors in identifying AI-generated code. *Proc. of the 46th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEET)*. Lisbon, 2024. P. 1–12.
6. Nguyen P.T., Di Rocco J., Di Ruscio D. et al. GPTSniffer: A CodeBERT-based classifier to detect source code written by ChatGPT. *Journal of Systems and Software*. 2024. Vol. 214. P. 112059.
7. Oedingen M., Engelhardt R.C., Denz R. et al. ChatGPT code detection: Techniques for uncovering the source of code. *AI*. 2024. Vol. 5, N 3. P. 1066–1094.
8. Ивахненко А.Г. Индуктивный метод самоорганизации моделей сложных систем. Киев: Наукова думка, 1981. 296 с.
9. Голуб С.В., Немов Р.Г. Формування масиву чисельних ознак для класифікації авторства програмних кодів із використанням символьних крос-зв’язків між словами. *Математичні машини і системи*. 2026. № 2. С. 70–78. DOI: https://doi.org/10.34121/1028-9763-2026-2-70-78.

Стаття надійшла до редакції 02.06.2026 / прийнята до друку 13.08.2026