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

УДК 004.93:004.85

С.В. ГОЛУБ[*], Р.Г. НЕМОВ[*]

ФОРМУВАННЯ МАСИВУ ЧИСЕЛЬНИХ ОЗНАК ДЛЯ КЛАСИФІКАЦІЇ АВТОРСТВА ПРОГРАМНИХ КОДІВ ІЗ ВИКОРИСТАННЯМ СИМВОЛЬНИХ КРОС-ЗВ'ЯЗКІВ МІЖ СЛОВАМИ

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

Анотація. *Розглянуто задачу автоматичної атрибуції авторства програмних кодів як складову інформаційної технології інтелектуального моніторингу. Існуючі підходи до атрибуції коду спираються переважно на синтаксичні ознаки конкретної мови програмування (абстрактні синтаксичні дерева, токени, лексичні конструкції) і тому не переносяться між мовами, тоді як реальні автори часто пишуть код різними мовами, зберігаючи характерний стиль. Як альтернативу використано методологію формування масиву вхідних даних (МВД) школи С.В. Голуба, розроблену в дисертаційному дослідженні М.С. Голуб для класифікації україномовних текстів, з імовірнісним критерієм інформативності та межею інформативної достатності (МІД). Запропоновано новий тип ознак — символьні крос-зв'язки між словами, які розширюють словник школи С.В. Голуба та характеризують парні комбінації суфіксів і префіксів ідентифікаторів коду в межах вікна фіксованої довжини. Формалізовано крос-зв'язки трьох рангів (1×1, 2×2, 3×3) як частоти появи впорядкованих пар k-символьних рядків при переборі всіх упорядкованих пар слів у вікні. Експериментально досліджено ефективність підходу на датасеті з 12 авторів (10 людей і дві генеративні моделі штучного інтелекту — ChatGPT та ClaudeCode) у чотирьох мовах програмування (Java, JavaScript, TypeScript, Python), 119 програмних класів і 641 вікно. У внутрішньомовному сценарії отримано 89,8–100 % правильно класифікованих вікон; у крос-мовному сценарії запропонований метод забезпечив 100% правильно класифікованих вікон при розмірі вікна 500 знаків, що відповідає перевазі методу до +1,30 %. Крос-зв'язки активно долають поріг МІД і становлять до 77 % обсягу адаптивного словника, що свідчить про їх високу інформативність як нового мовно-незалежного типу ознак для задач атрибуції авторства програмних кодів.*

Ключові слова: *атрибуція коду, класифікація текстів, словник ознак, масив вхідних даних, межа інформативної достатності, крос-зв'язки між словами, інтелектуальний моніторинг, МГУА.*

Abstract. *The paper addresses the problem of automated source-code authorship attribution as a component of the information technology for intellectual monitoring. Existing approaches to code attribution rely mainly on language-specific syntactic features — abstract syntax trees, tokens, or lexical constructs — and consequently do not generalize across programming languages, whereas real-world developers frequently write code in multiple languages while retaining a recognizable individual style. As an alternative, the methodology for constructing an input data array (IDA) within the S.V. Holub research school, developed in the dissertation research of M.S. Holub for the classification of Ukrainian-language texts, which employs a probabilistic feature-informativeness criterion and an information sufficiency threshold (IST), has been used. A new feature type — symbolic cross-links between words — that extends the Holub feature dictionary by capturing paired combinations of prefixes and suffixes of code identifiers within a fixed-length window, is used. Cross-links of three ranks (1×1, 2×2, 3×3) are formalized as occurrence frequencies of ordered k-character string pairs over all ordered word pairs within a window. The effectiveness of the proposed approach has been experimentally investigated on a dataset of 12 authors (10 human developers and 2 generative artificial-intelligence models — ChatGPT and Claude Code) across four programming languages (Java, JavaScript, TypeScript, and Python), comprising 119 source classes and 641 windows. In the within-language scenario, 89.8–100 % of windows have been*

ISSN 1028-9763. Математичні машини і системи. 2026. № 2

correctly classified. In the cross-language scenario, the proposed method achieves 100% accuracy in window classification at a window size of 500 characters, corresponding to a method advantage of up to +1.30 %. Cross-link features actively pass the IST and constitute up to 77 % of the adaptive dictionary volume, which demonstrates their high informativeness as a new, language-independent type of features for problems of software code authorship attribution.
Keywords: *code attribution, text classification, feature dictionary, input data array, information sufficiency threshold, cross-links between words, intellectual monitoring, GMDH.*

DOI: 10.34121/1028-9763-2026-2-70-78

1. Вступ

Задача автоматичної атрибуції авторства програмного коду набуває дедалі більшого значення у зв'язку з розвитком інформаційних технологій у сферах кібербезпеки, виявлення плагіату у навчальному процесі, проведення судової програмно-технічної експертизи та аналізу загроз в екосистемах відкритого програмного забезпечення. Поширення великих мовних моделей та генеративних інструментів штучного інтелекту ускладнило цю задачу додатковим виміром — потребою відрізнити людського автора від автоматично згенерованого коду. Всі ці завдання потребують ефективних методів перетворення програмного коду до форми, придатної для машинного навчання моделей-класифікаторів.

Існуючі підходи до атрибуції коду здебільшого використовують ознаки, тісно пов'язані з конкретною мовою програмування: абстрактні синтаксичні дерева (AST), токени, лексичні конструкції. Це створює суттєве обмеження: моделі, навчені на одній мові програмування, не переносяться на інші. Водночас реальні автори часто пишуть програмний код різними мовами залежно від проєкту, зберігаючи при цьому власний стиль оформлення ідентифікаторів, коментарів, форматування. Це мотивує пошук мовно-незалежних ознак, які б характеризували саме автора, а не мову.

Альтернативний підхід запропоновано у науковій школі С.В. Голуба та розвинено у працях М.С. Голуб [1, 2] для задачі класифікації україномовних текстів у технології інтелектуального моніторингу. Підхід ґрунтується на перетворенні тексту до форми двовимірного масиву чисельних ознак — масиву вхідних даних (МВД) — із подальшим синтезом моделей-класифікаторів індуктивними методами, зокрема методом групового урахування аргументів (МГУА) [3]. Ключовою складовою методу є адаптивне формування словника ознак на основі критерію інформативності та межі інформативної достатності (МІД).

Поширення цього підходу на програмні коди досі не здійснювалось. Словник ознак, розроблений для україномовних текстів (довжини слів, n-грами символів, перші та останні літери слів), може виявитися недостатнім для характеризації стилю автора у програмних кодах, де структура ідентифікаторів, конвенції іменування та особливості використання службових символів несуть додаткову авторську інформацію. Виникає потреба в розробці нового типу ознак, що враховують зазначені особливості.

2. Аналіз останніх досліджень і публікацій

Сучасні підходи до атрибуції авторства програмного коду зазвичай спираються на лексичний, синтаксичний або семантичний аналіз. Роботи закордонних авторів демонструють високу точність на великих корпусах коду, проте майже всі вони мовно-залежні та потребують побудови AST для конкретної мови, що обмежує застосовність у практичних задачах крос-мовної ідентифікації.

Українська наукова школа С.В. Голуба розробила альтернативний методологічний апарат — інформаційну технологію багаторівневого інтелектуального моніторингу [4], в якій задача класифікації текстів розв'язується через агентний синтез моделей на основі

чисельних характеристик тексту. У монографічних працях та дисертаційному дослідженні М.С. Голуб [1, 2, 5] запропоновано ймовірнісний критерій інформативності ознаки, метод адаптивного формування індивідуального словника ознак для кожної задачі класифікації, дисперсний метод побудови точок спостереження. Експерименти на україномовних текстах показали емерджентний ефект — досягнення 98–100 % правильної класифікації при визначенні авторства, гендерної ідентифікації та географічної атрибуції за місцем проживання автора.

У працях [6, 7] авторами цієї статті започатковано перенесення методології школи С.В. Голуба на задачу аналізу програмних кодів. У попередніх дослідженнях показано технічну можливість перетворення коду до форми МВД засобами типових інструментів моніторингової інтелектуальної системи. Водночас питання розширення словника ознак новими типами, що враховують специфіку коду, до цього часу не розглядалося.

Отже, актуальним залишається питання формування нових типів ознак, які, з одного боку, зберігають мовно-незалежний характер методології школи С.В. Голуба, а з іншого – враховують структурну особливість програмного коду як тексту з високою щільністю ідентифікаторів.

3. Мета статті

Метою статті є розробка нового типу ознак — символьних крос-зв'язків між словами, які розширюють словник ознак методології школи С.В. Голуба та забезпечують підвищення селективності моделі-класифікатора при класифікації авторства програмних кодів, а також експериментальна перевірка запропонованого підходу у внутрішньомовному та крос-мовному сценаріях.

4. Формалізація методу

4.1. Постановка задачі класифікації

Задача класифікації авторства програмних кодів формалізована за аналогією до задачі класифікації текстів у [1]. Дано скінченну множину програмних кодів

$$T = \{t_1, t_2, ..., t_n\}, \tag{1}$$

які є навчальною вибіркою та експертно згруповані за авторством у m класів множини K:

$$K = \{k_1, k_2, ..., k_m\}, \tag{2}$$

де m — кількість авторів, за якими групуються програмні коди.

Необхідно побудувати модель-класифікатор f, що забезпечує відображення елементів множини $T^* = \{t_{n+1}, t_{n+2}, ..., t_{n+p}\}$ (тобто нових програмних кодів, отриманих після навчання моделі; $T^* \notin T$ на елементи множини K):

$$f : T^* \to K. \tag{3}$$

Властивості моделі залежать від вектора інформаційних ознак МВД x (що розраховуються у вікнах фіксованої довжини), вектора довжин вікон l та вектора алгоритмів синтезу моделей (АСМ) φ:

$$f = f(x, l, \varphi). \tag{4}$$

Адекватність класифікації оцінюється за кількістю правильно класифікованих точок спостереження (вікон) та кількістю правильно класифікованих програмних кодів у цілому після статистичної обробки результатів класифікації їх вікон.

4.2. Базовий словник ознак

Відповідно до методології [1, 2], вхідний текст програмного коду на першому етапі розбивається на вікна однакової довжини (параметр *N*, знаків). У межах кожного вікна обчислюються чисельні ознаки, що формують рядок МВД. Базовий словник ознак, успадкований із методології класифікації україномовних текстів, містить такі типи:

– базові метрики вікна: середня довжина слова, середня довжина рядка, кількість рядків, кількість порожніх рядків;

– частотні характеристики символів (n-грами): частоти появи окремих символів, пар суміжних символів (біграми) та трійок суміжних символів (триграми);

– позиційні ознаки слів: частоти появи перших та останніх 1, 2 або 3 символів у словах позначені у форматі «*a**» (початок слова) та «***s*» (кінець слова).

Для виділення слів використовується регулярний вираз \W+ (будь-яка послідовність символів, що не є буквою, цифрою або знаком підкреслення). Такий підхід коректно обробляє ідентифікатори у camelCase, snake_case та інші конвенції іменування, що зустрічаються у програмних кодах різних мов.

4.3. Символьні крос-зв'язки між словами

Основною новизною цієї роботи є введення нового типу ознак — символьних крос-зв'язків між словами, що характеризують парні комбінації символів на межі пар слів у вікні.

Нехай задано вікно тексту програмного коду, з якого виділено впорядковану послідовність слів $W = (w_1, w_2, ..., w_L)$. Символьним крос-зв'язком рангу $k \in \{1, 2, 3\}$ між словами w_i та $w_j (i \neq j)$ називається впорядкована пара k-символьних рядків:

$$c_k(w_i, w_j) = (suf_k(w_i), pre_k(w_j)), \tag{5}$$

де $suf_k(w)$ — останні k символів слова w, $pre_k(w)$ — перші k символів слова w. Крос-зв'язок визначений лише за умови $|w_i| \geq k$ та $|w_j| \geq k$. Для рангу $k = 1$ умова $|w| \geq 1$ виконується для будь-якого непорожнього слова.

Ознакою крос-зв'язку рангу k для заданої пари рядків $(s, p) \in \sum^k \times \sum^k$ у вікні W є частота появи цієї конкретної пари серед усіх упорядкованих пар слів:

$$f_{c_k(s,p)}(W) = |\{i, j\} : i \neq j, |w_i| \geq k, |w_j| \geq k, suf_k(w_i) = s, pre_k(w_j) = p\}|, \tag{6}$$

де Σ — алфавіт (у даному випадку — множина символів, що утворюють слова за регулярним виразом \W+). Ознаки крос-зв'язків позначаються у форматі «*XX YY*», де XX — останні k символів першого слова, YY — перші k символів другого слова.

Принциповою особливістю запропонованих ознак є використання всіх упорядкованих пар слів у вікні (не тільки сусідніх). Загальна кількість упорядкованих пар у вікні з L слів становить $L(L-1)$, з яких до крос-зв'язку рангу k потрапляють лише пари, в яких обидва слова мають довжину не меншу за k. Такий підхід робить запропоновані ознаки нечутливими до порядку слів у вікні та забезпечує стійкість до локальних переставлень, характерних для програмного коду (порядок рядків імпортів, послідовність оголошень полів класу тощо).

У цій роботі досліджуються три ранги крос-зв'язків:

– cross1 $(k = 1)$: останній символ першого слова + перший символ другого слова (формат «**a b**»);

– cross2 $(k = 2)$: два останні символи першого слова + два перші символи другого слова (формат «*ab cd*»);

– cross3 $(k=3)$: три останні символи першого слова + три перші символи другого слова (формат «*abc def*»).

4.4. Критерій інформативності та межа інформативної достатності

Для фільтрації ознак застосовується ймовірнісний критерій інформативності, обґрунтований у [1, 2]:

$$P_i = \frac{V_i}{n} \cdot 100\%, \tag{7}$$

де P_i — показник інформативності i-ї ознаки (ймовірність використання ознаки у вікні, виражена у відсотках), V_i — кількість випадків використання i-ї ознаки в окремому вікні, n — загальна кількість знаків у вікні.

Задається мінімальне значення критерію — межа інформативної достатності (МІД, параметр T, %). Ознака потрапляє до адаптивного словника, якщо її показник інформативності перевищує МІД. Застосовуються два режими фільтрації: MAX-режим (ознака лишається, якщо хоча б в одному вікні категорії «своїх» її значення перевищує поріг) та AVG-режим (ознака лишається, якщо середнє її значення по всіх вікнах категорії «своїх» перевищує поріг).

4.5. Формула переваги методу

Перевагу запропонованого методу з крос-зв'язками порівняно з базовим словником без них обчислено за формулою, запропонованою у [2]:

$$P = \frac{n_a - n_g}{n_g} \cdot 100\%, \tag{8}$$

де n_a — кількість правильно класифікованих точок спостереження зі словником, що містить крос-зв'язки; n_g — кількість правильно класифікованих точок спостереження з базовим словником (без крос-зв'язків).

5. Експериментальне дослідження

5.1. Датасет

Для експериментальної перевірки запропонованого методу сформовано датасет із 12 авторів програмних кодів. З них 10 — люди, авторство кодів яких підтверджено публічними репозиторіями, та 2 — генеративні моделі штучного інтелекту (ChatGPT, ClaudeCode), що дає можливість дослідити придатність методу для розрізнення людського та машинно-згенерованого коду. Характеристики датасету наведені в табл. 1.

Таблиця 1 — Характеристики датасету

Автор	Основна мова	Тип автора	Класів / вікон
DmytroDanylyk	Java	Людина	10 / 40
JoshuaBloch	Java	Людина	14 / 26
VladKravets	Java	Людина	10 / 41
DmytroSemenov	JavaScript	Людина	10 / 83
VolodymyrAgafonkin	JavaScript	Людина	10 / 97

Продовж. табл. 1

PaulMiller	**TypeScript**	**Людина**	**10 / 125**
VladShatskiy	TypeScript	Людина	10 / 87
TarasMaichuk	C# / TS	Людина	10 / 22
MaxKotliar	Python	Людина	10 / 68
SergheiIakovlev	Python	Людина	10 / 52
ChatGPT	Java	ШІ (OpenAI)	7 / 0
ClaudeCode	Java	ШІ (Anthropic)	8 / 0

5.2. Протокол експерименту

Для кожної досліджуваної серії обрано одного автора як позитивний клас («свій», код мітки +100), решту авторів — як негативний клас («чужі», код мітки — 100). Для кожної серії виконано два запуски класифікації: (i) з базовим словником ознак без крос-зв'язків; (ii) з розширеним словником, що включає крос-зв'язки рангів 1, 2 та 3. Параметри формування МВД у цій роботі фіксовано на значеннях N=500 знаків, T=10 %, режим фільтрації — AVG (середній). Задача параметричної оптимізації цих параметрів є предметом окремого дослідження.

У турнірі моделей-класифікаторів беруть участь понад 30 архітектур нейронних мереж (сімейства Dense, CNN_1D, LSTM/GRU/BiLSTM, Attention, Autoencoder, TFHub-моделі NNLM та BERT) разом із реалізацією МГУА. Для кожної серії фіксується найкраща архітектура за точністю класифікації вікон.

Дослідження виконано у двох сценаріях: (А) внутрішньомовна атрибуція — коли «свій» та «чужі» автори пишуть однією мовою програмування; (В) крос-мовна атрибуція — коли «свій» автор пише однією мовою, а «чужі» — різними (до 4 мов одночасно). Ключовим питанням є ефективність крос-зв'язків у сценарії (В), де задача ускладнена необхідністю абстрагуватися від особливостей мови програмування.

5.3. Результати внутрішньомовного сценарію

Результати експерименту у внутрішньомовному сценарії (Java-автори проти Java-авторів) подані у табл. 2.

Таблиця 2 — Результати внутрішньомовної атрибуції (Java)

Серія	«Свій»	«Чужі»	Варіант	Вікна, %	Класи, %	Ознак
1.1	DmytroDanylyk	JoshuaBloch	Базовий	96.30	100,00	16
1.1	DmytroDanylyk	JoshuaBloch	З крос	96.30	100,00	47
1.2	JoshuaBloch	DmytroDanylyk	Базовий	100.00	96,77	17
1.2	JoshuaBloch	DmytroDanylyk	З крос	100.00	96,77	58
1.3	VladKravets	DmytroDanylyk	Базовий	100.00	100,00	20
1.3	VladKravets	DmytroDanylyk	З крос	100.00	100,00	39
1.5	ClaudeCode	DmytroDanylyk	Базовий	88.64	97,37	15
1.5	ClaudeCode	DmytroDanylyk	З крос	89.77	94,74	31

Аналіз результатів внутрішньомовного сценарію (табл. 2) свідчить про ефект насичення точності: у серіях 1.1–1.3 базовий варіант уже забезпечує від 96 до 100% правильно класифікованих вікон, що не залишає простору для подальшого покращення додаванням крос-зв'язків. Водночас у серії 1.5, де «своїм» є машинно-згенерований код (ClaudeCode), а «чужими» — людський код Java-автора, крос-зв'язки забезпечили приріст точності вікон на +1.13 п.п.

Принциповим спостереженням є стале зростання розміру адаптивного словника при додаванні крос-зв'язків: у 2,0–3,4 рази (з 15–20 ознак у базовому варіанті до 31–58 у розширеному). Це означає, що крос-зв'язки активно долають поріг МІД і складають істотну частку інформативних ознак, тобто несуть нову інформацію, а не дублюють класичні ознаки словника школи С.В. Голуба.

5.4. Результати крос-мовного сценарію

У крос-мовному сценарії кожен із трьох обраних авторів послідовно виступає у ролі «свого», а «чужими» є дев'ятеро інших авторів, що пишуть різними мовами програмування. Результати подані у табл. 3.

У крос-мовному сценарії отримано ключовий результат досліду — у серії 5.2 (DmytroSemenov, JavaScript проти дев'яти авторів-чужих) застосування крос-зв'язків підвищило точність класифікації вікон з 98,72 % до 100,00 %, що відповідає перевазі методу за формулою (8) P=+1,30 %. Одночасно розмір адаптивного словника збільшився з 19 до 83 ознак (у 4,37 рази). Це максимальне зростання серед усіх серій, що свідчить про високу інформативність крос-зв'язків саме у найбільш «шумному» сценарії — коли автор пише однією мовою, а більшість «чужих» — іншими.

У серії 5.3 (JoshuaBloch як «свій») спостерігається нетривіальна поведінка: на рівні окремих вікон крос-зв'язки дали незначне зниження точності (99,36 % → 98,72 %), однак після статистичної обробки результатів класифікації вікон (мажоритарне голосування) точність на рівні класів збільшилася з 98,94 % до 100,00 % (перевага P=+1,07 %). Це узгоджується з висновком М.С. Голуб [1, 2] про ефективність процедури мажоритарного голосування як засобу підвищення адекватності класифікації.

Таблиця 3 — Результати крос-мовної атрибуції

Серія	«Свій» (мова)	Варіант	Вікна, %	Класи, %	Ознак
5.1	DmytroDanylyk (Java)	Базовий	98,72	100,00	16
5.1	DmytroDanylyk (Java)	З крос	98,08	100,00	47
5.2	DmytroSemenov (JS)	Базовий	98,72	100,00	19
5.2	DmytroSemenov (JS)	З крос	100,00	100,00	83
5.3	JoshuaBloch (Java)	Базовий	99,36	98,94	17
5.3	JoshuaBloch (Java)	З крос	98,72	100,00	58

У серії 5.1 (DmytroDanylyk) обидва варіанти досягають 100 % точності на рівні класів, що унеможливлює демонстрацію переваги методу в цьому окремому випадку.

Архітектури моделей, що виявилися найкращими для варіанта з крос-зв’язками у крос-мовному сценарії, — TF_Dense_Ultra (серія 5.2) та TF_Dense_Pyramid (серія 5.3). Ці архітектури належать до сімейства повнозв'язних глибоких мереж і добре пристосовані до роботи з векторами чисельних ознак середньої розмірності (до 100 ознак).

6. Обговорення результатів

Отримані результати підтверджують робочу гіпотезу про те, що символьні крос-зв’язки між словами є інформативним типом ознак для класифікації авторства програмних кодів. Обидва сценарії експерименту (внутрішньомовний та крос-мовний) показали стале збільшення розміру адаптивного словника після додавання крос-зв’язків у 2,0–4,4 рази. Таке зростання не може бути пояснене випадковим потраплянням ознак у словник: воно означає, що значна частина крос-зв’язків долає поріг МІД за критерієм (7), тобто має ймовірність використання у вікні не нижчу від порогу T=10 %.

На відміну від класичних ознак (n-грами, позиційні ознаки слів), що фіксують локальні характеристики тексту, крос-зв’язки характеризують парні комбінації фрагментів різних слів у межах одного вікна. У програмному коді такі комбінації часто відображають авторські конвенції: типові поєднання імен типів і змінних, префікси-суфікси ідентифікаторів у межах одного класу чи модуля, особливості іменування в Android-, Flask- та React-екосистемах тощо. Принципово, що крос-зв’язки враховують не фіксовану послідовність слів, а саме факт їх співіснування у вікні — це робить ознаку стійкою до локальних переставлень (порядок імпортів, оголошень полів) і водночас чутливою до структурних особливостей стилю автора.

Ключовий експериментальний результат — досягнення 100 % точності класифікації вікон у крос-мовному сценарії (серія 5.2) — свідчить про мовну незалежність запропонованих ознак. Автор-JS-розробник коректно ідентифікується навіть за умов, коли його стиль порівнюється зі стилями дев'яти інших авторів, що пишуть чотирма різними мовами програмування. Такий результат узгоджується з загальною філософією методології школи С.В. Голуба — переходом на глибший за лексемний рівень декомпозиції тексту [8], що забезпечує стійкість до мовно-специфічних артефактів.

Обмеженням поточного дослідження є обсяг експериментальної верифікації: у внутрішньомовному сценарії виконано 4 серії, у крос-мовному — 3 із запланованих 10. Ефект насичення точності для базового варіанта у внутрішньомовному сценарії свідчить, що для більш контрастної демонстрації переваги методу потрібні складніші задачі: з більшою кількістю класів, менш однорідними «чужими» або з меншим розміром вікна N. Ці напрями заплановано в подальших дослідженнях.

7. Висновки

Запропоновано новий тип ознак — символьні крос-зв’язки між словами — для класифікації авторства програмних кодів у структурі інформаційної технології інтелектуального моніторингу. Крос-зв’язки рангів 1, 2, 3 формалізовано як частоти появи впорядкованих пар рядків, що утворюються суфіксами та префіксами слів у межах одного вікна, при переборі всіх упорядкованих пар слів.

Методологію формування масиву вхідних даних із ймовірнісним критерієм інформативності та межею інформативної достатності, розроблену у працях школи С.В. Голуба для задач класифікації україномовних текстів, успішно перенесено на задачу класифікації авторства програмних кодів. Словник базових ознак розширено крос-зв’язками, що в експериментах становлять до 77 % обсягу адаптивного словника.

Експериментально на датасеті з 12 авторів (10 людей та дві моделі штучного інтелекту) у чотирьох мовах програмування (Java, JavaScript, TypeScript, Python)

підтверджено інформативність нового типу ознак. У крос-мовному сценарії класифікації досягнуто 100 % правильно класифікованих вікон тексту коду, перевага методу з крос-зв'язками на рівні вікон становить до +1,30 %, на рівні класів — до +1,07 %.

Перспективою подальших досліджень є розширення експериментальної бази до повного переліку з 10 серій крос-мовного сценарію, дослідження ефективності методу на задачах розрізнення людського та машинно-згенерованого коду, а також параметрична оптимізація процесу формування МВД за критеріями розміру вікна, межі інформативної достатності та режиму фільтрації ознак.

СПИСОК ДЖЕРЕЛ

1. Голуб М.С. Формування масиву вхідних даних при класифікації текстів у технології інформаційного моніторингу. *Математичні машини і системи*. 2018. № 1. С. 59–66.
2. Голуб М.С. Формування масиву чисельних ознак для класифікації україномовних текстів в інформаційній технології інтелектуального моніторингу: дис. канд. техн. наук: 05.13.06 / Черкаський державний технологічний університет. Черкаси, 2018. 137 с.
3. Ивахненко А.Г. Индуктивный метод самоорганизации моделей сложных систем. Киев: Наукова думка, 1981. 296 с.
4. Голуб С.В., Жирякова І.А., Куницька С.Ю., Авраменко В.П. Методи розвитку моніторингових інтелектуальних систем. *Інформація, комунікація, суспільство 2019*: матеріали 8-ї Міжнар. наук. конф. ICS-2019. Львів: Видавництво Львівської політехніки, 2019. С. 65–67.
5. Голуб М.С. Дисперсійний метод формування точок спостереження в інформаційній технології класифікації текстів. *Вісник інженерної академії України*. 2017. № 3. С. 38–42.
6. Немов Р.Г., Голуб С.В. Агентне програмування інтелектуального аналізу кодів програм. *13 міжнародна наукова конференція ІКС-2024*. Львів, 2024. С. 133–135.
7. Немов Р.Г., Голуб С.В., Немченко В.В. Структурна динаміка програмного агента інформаційного моніторингу. ІТСМ. Івано-Франківськ, 2023. С. 91–94.
8. Голуб М.С. Вибір ознак у процесі інтелектуальної обробки текстових повідомлень. *Інформація, комунікація, суспільство 2014*: матеріали 3-ї Міжнар. наук. конф. ICS-2014. Львів: Видавництво Львівської політехніки, 2014. С. 148–149.
9. Голуб С.В., Константиновська О.В., Голуб М.С. Формування показників масиву вхідних даних для ідентифікації авторства текстових повідомлень. *Системи обробки інформації*: зб. наук. праць. Харків: Харківський університет повітряних сил імені Івана Кожедуба, 2014. Вип. 2 (118). С. 89–92.
10. Голуб М.С. Формування словника ознак для класифікації україномовних текстів в інформаційній технології багаторівневого інтелектуального моніторингу. І*нформація, комунікація, суспільство 2019*: матеріали 8-ї Міжнар. наук. конф. ICS-2019. Львів: Видавництво Львівської політехніки, 2019. С. 68–70.
11. Голуб С.В., Мартинова Г.І., Голуб М.С. Моделювання діалектного тексту в технології багаторівневого інформаційного моніторингу. *Математичні машини і системи*. 2016. № 4. С. 76–83.
12. Немов Р.Г., Голуб С.В. Агентне програмування інтелектуального аналізу кодів програм. І Міжнародна науково-практична конференція. Харків-Яремче, 2025. С. 218–220.

Стаття надійшла до редакції 03.02.2026 / прийнята до друку 28.04.2026