КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (29), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.29.908
УДК 004.89:004.93'1

Пелещак Іван Романович
PhD зі спеціальності 124, доцент каф. Інформаційних систем та мереж
Національний університет «Львівська політехніка», Львів, Україна
ORCID ID: 0000–0002–7481–8628
ivan.r.peleshchak@lpnu.ua

Литвин Василь Володимирович
д.т.н., професор, професор каф. Інформаційних систем та мереж
Національний університет «Львівська політехніка», Львів, Україна
ORCID ID: 0000–0002–9676–0180
vasyl.v.lytvyn@lpnu.ua

Висоцька Вікторія Анатоліївна
д.т.н., доцент, професор каф. Інформаційних систем та мереж
Національний університет «Львівська політехніка», Львів, Україна
ORCID ID: 0000-0001-6417-3689
victoria.a.vysotska@lpnu.ua

Хобор Олексій Романович
аспірант каф. Інформаційних систем та мереж
Національний університет «Львівська політехніка», Львів, Україна
ORCID ID: 0000–0002–3210–036X
oleksii.r.khobor@lpnu.ua

TINYBERT З CROSS-ATTENTION + LORA + BI-GRU: КОМПАКТНА НЕЙРОМЕРЕЖЕВА АРХІТЕКТУРА ДЛЯ РОЗПІЗНАВАННЯ ФЕЙКОВИХ НОВИН

Анотація У роботі запропоновано компактну нейромережеву архітектуру TinyBERT з механізмом перехресної уваги (Cross-Attention), адаптацією низького рангу (LoRA) та двонапрямним рекурентним шаром Bi-GRU для задачі автоматичного розпізнавання фейкових новин. Розвиток цифрового інформаційного простору спричинив різке зростання кількості фейків, а класичні методи факт-чекінгу вже не здатні ефективно протистояти такій динаміці. Наявні нейромережеві підходи, що базуються на великих трансформерних моделях, часто непридатні до практичного застосування через надмірну ресурсозатратність та нездатність явно аналізувати семантичну невідповідність між заголовком і текстом новини. Запропонована модель усуває ці обмеження завдяки використанню дистильованого трансформера TinyBERT, що значно зменшує обчислювальні вимоги, та спеціалізованого модуля Cross-Attention, який забезпечує чутливість до невідповідностей у заголовку й тексті. Використання LoRA-доповнень мінімізує кількість параметрів, що тренуються, прискорюючи процес донавчання, а додавання Bi-GRU дозволяє зберігати контекстуальну інформацію послідовностей. Експериментальне дослідження, виконане на наборі Fake News Classification, продемонструвало, що ця модель перевершує класичні алгоритми машинного навчання, досягнувши точності 99%, F1-score 0,985 і ROC AUC 0,998. Завдяки низьким ресурсним витратам, швидкій адаптації до нових тематик і високій інтерпретованості рішень, модель є перспективним рішенням для інтеграції в сервіси факт-чекінгу в реальному часі.

Ключові слова: виявлення фейкових новин; TinyBERT; перехресна увага; LoRA; Bi-GRU; обробка природної мови; трансформерні моделі; AI-фактчекінг.

ВСТУП

Бурхливе зростання обсягу цифрового контенту зробило «AI-підсилену дезінформацію» однією з найсерйозніших глобальних загроз. Дезінформація здатна

підірвати вибори, поляризувати суспільства й коштує світовій економіці десятки мільярдів доларів щороку. Класичний, ручний факт-чекінг не встигає за швидкістю «цифрових пожеж»: потік новин поширюється зі швидкістю сотень тисяч публікацій на годину, тоді як повна верифікація однієї статті може забирати у журналістів від пів години до кількох діб.

Із появою великих мовних моделей (LLM) автоматизована перевірка стала технічно можливою, однак переважна більшість існуючих нейронних рішень має суттєві обмеження.

Постановка проблеми. Сучасні «важкі» трансформери (Longformer-Large, мультимодальні GNN-стеги) містять сотні мільйонів параметрів, що робить їх непридатними для розгортання на редакційних CMS-серверах, у браузерних розширеннях або на мобільних пристроях. Одногілкові моделі з CLS-пулінгом помітно гірше ловлять головний риторичний трюк фейків — інконгруентність між «гучним» заголовком і відносно невинним тілом новини. Дослідження показують, що ця невідповідність є однією з найінформативніших ознак медіаманіпуляції, але звичайні енкодери її не бачать — потрібна спеціальна взаємодія двох текстових потоків [1]. Крім того, повне донавчання великих трансформерних архітектур потребує значних обчислювальних ресурсів і часу, що обмежує можливість їх оперативного застосування у динамічних інформаційних середовищах.

Таким чином, існує потреба у створенні компактної, високоефективної та інтерпретованої нейромережевої моделі, здатної виявляти фейкові новини в режимі реального часу, зокрема за рахунок точного аналізу взаємозв'язку між заголовком і змістом, та придатної для швидкої адаптації до нових тематик і мовних контекстів без значних витрат ресурсів.

Для вирішення вищеописаних проблем у цій роботі пропонується нова компактна нейромережева архітектура. Її хребтом є TinyBERT-4L-312D — дистильований трансформер, який зберігає якість та функції BERT-Base, але у 7,5 разів менший [2].

Над його виходами будується двонапрямний cross-attention звязок між заголовком і тілом «новини», що робить модель чутливою до семантичних конфліктів. Щоб утримати параметри на рівні сотень тисяч, ми застосовуємо Low-Rank Adaptation (LoRA): замість повного fine-tune оновлюються лише низькорангові доповнення, що дає скорочення тренованих ваг у 32 рази без втрати якості [3]. Зокрема, порядковий контекст агрегується через вузький двонапрямний Bi-GRU (64 прихованих елементів).

Таким чином запропонована мережа:

- доступна — придатна для розгортання в реальному часі навіть на обмежених пристроях (мобільні платформи);
- спеціалізована на виявленні headline-body інконгруентності, яку ігнорують більш загальні LLM;
- інтерпретована — теплові карти cross-attention і Grad-CAM над GRU демонструють причини класифікації;
- гнучка у донавчанні — LoRA-адаптери достатньо «легкі», щоб підлаштувати модель під локальний медіапростір за лічені хвилини.

Аналіз останніх досліджень і публікацій. Перші ґрунтовні спроби автоматичного виявлення фейкових новин спиралися на рекурентні та згорткові мережі. У роботі [4] двошарова Bi-LSTM, доповнена лексичними й стилістичними ознаками, перевершила традиційні SVM і стала відправною точкою для глибинних підходів. Згодом, у роботі [5] застосували ієрархічну увагу, поділивши текст на слова й речення; така архітектура додала приблизно три відсоткових пункти до F-міри, що довело цінність багаторівневої моделі тексту. Автори роботи [6] показали, що шлях поширення новини у соцмережі не

менш інформативний за сам контент: їхня мережа ієрархічної пропагації помітно підвищила точність розпізнавання завдяки інтеграції часових та авторських зв'язків.

Наступний етап протидії дезінформації позначився бурхливим розвитком графових і трансформерних підходів. У роботі [7] запровадили гетерогенну граф-мережу з увагою до вузлів різного типу — авторів, тем і самого тексту — що ще більше наблизило моделі до реальної структури медіапростору. У той самий період автори роботи [8] показали, що глибокі CNN у комбінації з LSTM-стеками дають сталий приріст, проте їхня модель впритул наблизилася до обмежень споживаної відеопам'яті.

Поява дистильованих трансформерів [9] забезпечила додавання зовнішніх знань до ієрархічної уваги. Тим часом, автори роботи [10] довели, що поєднання CNN-фільтрів із HAN та простими стилістичними сигналами може підвищити точність без суттєвого роздування моделі.

Далі дослідники звернулися до робастності та мультимодальності. Автори роботи [11] уперше використали перехресну увагу між заголовком і текстом, продемонструвавши, що інконгруентність цих двох частин суттєво підвищує точність класифікації. У роботі [12] запропонували мережу з адверсаріальним навчанням, що виробляє стійкість до лексичних підмін, однак така модель стала ще важчою й вимогливішою до ресурсів. У роботі [13] об'єднали текст, зображення та граф користувачів, перетнувши межу найвищих показників, але потребуючи сотень мільйонів параметрів та доступу до соціальних API.

Огляд проведений у роботі [14] підсумував, що повне донавчання великих трансформерів більше не є практичним; натомість Low-Rank Adaptation (LoRA) дозволяє вносити зміни, оновлюючи лише мізерну частку ваг. Цю ж ідею розвинуто у [15], де продемонстровано, що невеликі LoRA-шари можна комбінувати у ансаблі з LLM і одержувати результати, що конкурують із флагманськими системами, але під високою апаратною ціною.

Попри численні успіхи, у поточних підходів залишилися суттєві прогалини. У більшості моделей зберігаються надмірні апаратні вимоги, що унеможливлює їх розгортання на ноутбуках редакцій або в браузерних плагінах. Значна частина рішень неспроможна явно оцінювати невідповідність між заголовком і основним текстом, тоді як саме цей дисбаланс є фундаментальною рисою пропагандистських матеріалів.

Мета статті. Метою даного дослідження є розробка та експериментальна перевірка компактної нейромережевої архітектури на основі дистильованого трансформера TinyBERT з механізмом перехресної уваги (Cross-Attention), адаптацією низького рангу (LoRA) та двонапрямним рекурентним шаром Bi-GRU для задачі автоматичного виявлення фейкових новин. Запропоноване рішення покликане забезпечити високу точність класифікації при значно знижених обчислювальних витратах, з можливістю явного аналізу семантичної відповідності між заголовком і текстом новини, швидкої адаптації до нових тематик та інтеграції у системи факт-чекінгу в режимі реального часу на пристроях з обмеженими апаратними ресурсами.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Схема передавання даних у моделі. Потік даних у моделі починається на рівні сирих записів CSV, де кожна новина подається як пара рядків — заголовок і основний текст. Ці два фрагменти проходять через блок попередньої обробки, що усуває HTML-теги, нормалізує кодування Unicode та приводить текст до нижнього регістру. Очищені

рядки потрапляють у WordPiece-токенізатор TinyBERT: заголовок перетворюється на послідовність із максимумом 64 токенів, а тіло — на послідовність довжиною до 448 токенів; за потреби зайві слова обрізаються, а короткі тексти доповнюються заповнювачами [PAD]. На обидві послідовності накладаються службові маркери [CLS] та [SEP], а паралельно формується двійкова *attention_mask*, що відрізняє реальні токени від падингів (рис. 1).

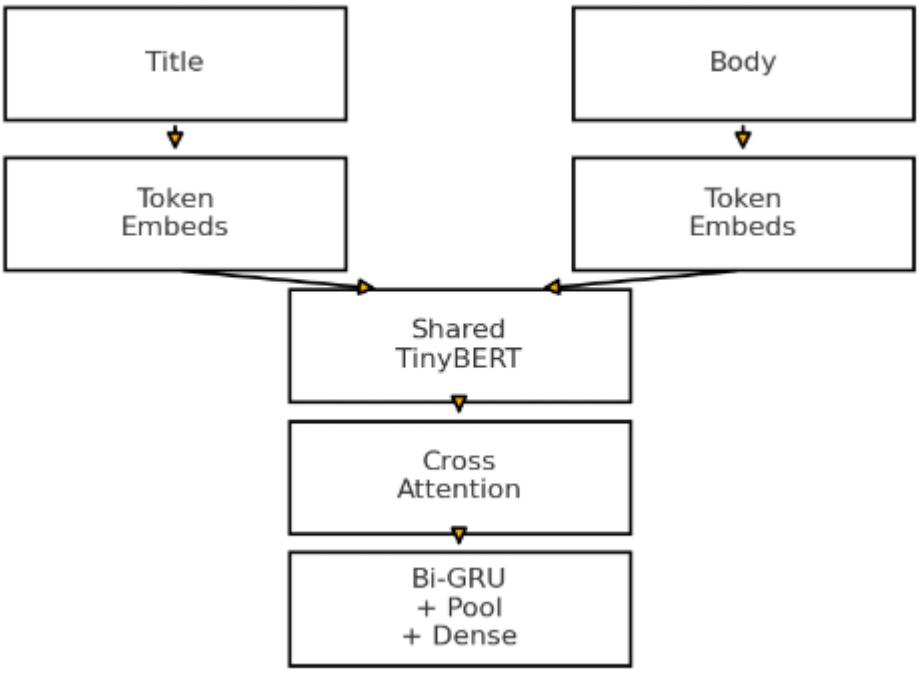


Рис. 1. Схема передавання даних у моделі

Далі обидві секції незалежно подаються до «спільного» TinyBERT-енкодера, який уже не змінює свої ваги на першому етапі навчання й виконує роль контекстуального перетворювача: кожен токен одержує 312-вимірний вектор, насичений локальною та глобальною семантикою. Виходи заголовка та тіла збігаються в модуль перехресної уваги: токени заголовка виступають запитами, а токени тіла — ключами й значеннями, і навпаки. Цей шар зіставляє два текстові потоки, посилюючи саме ті пари слів і речень, між якими виникає змістовний конфлікт або підтвердження. Всі великі матриці проєкцій у цьому блоці заморожені; адаптуються лише їхні низькорангові LoRA-доповнення, тож величина нового навчального простору залишається мікроскопічною.

Результатом перехресної уваги є єдина послідовність, що містить вже «зшиту» інформацію заголовка і тексту. Вона надходить до компактного двонапрямного GRU із 64 прихованими елементами. Рекурентний шар проходить по ній уперед і назад, акумулює порядковий контекст, а на виході до кожної позиції виробляє сукупний вектор важливості. Щоб згорнути часовий вимір у фіксований розмір, використовується глобальний макс-пулінг: з усіх токенів вибираються максимальні активації, що репрезентують найсильніші «аргументи» за або проти достовірності статті. Отриманий 128-вимірний вектор піддається дроп-ауту, а далі проходить крізь мінімальний лінійний шар, який видає один логіт — показник фейковості. Сигмоїда перетворює його у ймовірність, після чого рішення відсічки визначає, чи буде новина позначена як фейк.

Таким чином дані прямують від необробленого тексту до кінцевої класифікації через п’ять логічних вузлів — нормалізацію, токенізацію, контекстуалізацію трансформером, комбінування перехресною увагою та компактне рекурентне згортання — кожен із яких додає інформаційну цінність, але не перевищує апаратних можливостей навіть обмежених платформ.

Попередня обробка даних. У наборі Fake News Classification [16], що використовуємо для навчання та тестування цієї моделі є 37 106 Fake записів та 35 028 True записів, де кожен рядок містить чотири поля (Серійний номер (починаючи з 0); Заголовок (про текстовий заголовок новини); Текст (про зміст новини); та Мітка (1 = фейк та 0 = справжній).

Перш ніж подавати новинний контент до токенізатора, кожен заголовок і саме тіло «новини» проходять ланцюжок попередньої обробки, що усуває нерівномірності, які можуть збити статистику частот і спотворити простір ознак. Насамперед текст очищається від залишків HTML-розмітки: теги, службові атрибути та ентиті-коди переводяться у звичайні символи. Далі застосовується нормалізація Unicode у формі, аби всі «композитні» літери було зведено до єдиного коду, — це особливо важливо для слів із діакритиками, що інакше трактувалися б як різні токени.

Після уніфікації кодування текст зводиться до нижнього регістру, оскільки модель TinyBERT навчається в uncased-режимі й не розрізняє великі та малі літери. Наступний крок — гармонізація типографіки: лапки («…», „…“) і довгі чи короткі тире перетворюються на стандартні ASCII-аналоги. Така операція зменшує кількість унікальних токенів і дає змогу токенізатору WordPiece коректно поєднувати сегменти слів без появи зайвих.

Далі послідовність пропусків стискується до одного пробілу, а табуляції та зворотні переклади рядка замінюються пробілом, щоб токенізатор не створював порожні чи напівпорожні токени. Нарешті, із тексту видаляються непечатні керівні символи; вони часто зустрічаються в скопійованих фрагментах та можуть викликати помилкове відображення або вибух числа невідомих токенів.

У результаті цих кроків отримуємо чистий, стилістично однорідний рядок, який надалі безпечно передавати до WordPiece-токенізатора TinyBERT. Це забезпечує стабільний словник і загалом підвищує якість подальшого кодування тексту.

Токенізація та побудова послідовностей. Після нормалізації текст переходить на етап токенізації, де «сирі» символи перетворюються на числові індекси, зрозумілі нейронній мережі. Для цього застосовується WordPiece-токенізатор, навчений разом із TinyBERT. Його словник містить 30 522 одиниці — від цілих слів до характерних субслів. Робота WordPiece-токенізатора TinyBERT (словник $|V| = 30522$) описується формулою (1). Для рядка s він повертає послідовність індексів $\tau(s) = (x_1, ..., x_L)$.

$$\tau : str \to \{0, ..., |V| - 1\} \quad (1)$$

Така сегментація дозволяє зберігати рідковживані терміни й власні назви, розбиваючи їх на коротші, вже відомі лексеми, і водночас утримувати словник компактним. Токенізатор зчитує рядок зліва направо, вибираючи найдовший підрядок, який присутній у словнику; якщо слово не знайдено цілком, алгоритм рекурсивно ділить його навпіл, доки кожна частина не буде впізнана. Невідомі фрагменти замінюються на спеціальний токен [UNK], хоча після описаної раніше нормалізації такі випадки трапляються рідко.

У процесі формування батча заголовок і текст статті обробляються окремо. Для заголовка встановлено верхню межу у 64 токени разом зі службовими $L_t^{\max} = 64$, а для тексту — 448 токенів $L_b^{\max} = 448$. Якщо рядок коротший, він доповнюється заповнювачами [PAD], доки не досягне фіксованої довжини. Якщо довший, активується обрізання заголовка до потрібної довжини, а для тексту застосовується принцип «ковзаючого вікна» з кроком 128 токенів, щоб модель поступово побачила весь матеріал без катастрофічної втрати кінцівки.

Далі в обидві послідовності вписуються спеціальні маркери. На початку ставимо [CLS] — універсальний прапорець, чий вектор у класичних BERT-подібних архітектурах використовується для класифікації, але у нашому випадку він лишається в послідовності як звичайний токен. У кінці кожної частини ставимо [SEP], що сигналізує нейронній мережі про межу логічного блоку. У підсумку отримуємо два масиви input_ids: токени заголовка і токени тіла.

Паралельно створюються двійкові маски *attention_mask*: елементи, що відповідають реальним токенам, позначаються одиницею, а місця куди покладено [PAD] — нулем. Під час подальшої обробки у трансформері ці нулі заглушаються, тож мережа не витрачає обчислення на порожнечу й не допускає «витікання» градієнтів у падингові позиції.

Нарешті, для кожного токена формується позиційний індекс — його порядковий номер у межах послідовності. Цей номер мапується на вектор з окремої таблиці позиційних ембеддингів, і разом зі словниковим вектором вони складаються, утворюючи повний 312-вимірний представник токена. Після цієї операції матриці заголовка й тіла мають уніфікований формат і готові до передавання у спільний блок TinyBERT.

Блок TinyBERT-4L-312D. Основою усієї архітектури виступає дистильований трансформер TinyBERT, складений лише з чотирьох шарів «класичного» блоку увага + Feed-Forward. На вході він приймає вже готові матриці ембеддингів заголовка й тіла, у яких кожний токен має 312-вимірний вектор. Першою операцією є позиційне й токен-ембеддингове додавання: словниковий вектор, витягнутий із таблиці вбудов, підсумовується з позиційним вектором, що кодує порядковий номер.

Для кожного вхідного токена $x_i \in \{0,...,V-1\}$ формується вектор $\bar{e} = E_{x_i} + P_i$, де $E \in R^{V \times d}$ — словниковий елемент, $P \in R^{L_{\max} \times d}$ — позиційний внесок.

Таблиця 1

Кількість гіперпараметрів

Параметр	Значення
Кількість трансформер-шарів *M*	4
Розмір прихованого простору *d*	312
Кількість голів у Self-Attention *h*	12

Отримані матриці послідовно проходять через чотири трансформер-шари. Кожен шар складається зі дванадцяти голів самоуваги, які паралельно будують матриці запитів, ключів і значень. Після того, як вагові коефіцієнти уваги нормалізовано за софтмаксом, токени одержують контекстні вектори, багаті на інформацію про лексичне оточення, синтаксис і загальні мовні закономірності. Кожен такий блок завершується двошаровою feed-forward-мережею з нелінійністю GELU та двома резидуальними додаваннями з пост-нормалізацією LayerNorm. Цей внутрішній «ритм» увага → нормалізація → FFN →

нормалізація повторюється чотири рази, що дає достатню глибину для вивчення складної політичної та публіцистичної лексики, але не перевантажує пам'ять.

Ключовий момент полягає в тому, що одна й та сама копія TinyBERT обслуговує і заголовок, і основний текст (2). Через це ми фактично ділимо ваги між двома потоками і тим самим удвічі скорочуємо витрати пам'яті порівняно з підходами, де на заголовок і тіло тримають окремі енкодери.

$$H_t = TinyBERT\left(X^{title}, m^t\right), \quad H_b = TinyBERT\left(X^{body}, m^b\right), \tag{2}$$

де $H_t \in R^{L_t \times d}$, $H_b \in R^{L_b \times d}$ — приховані стани назви та тексту новини.

На першому етапі навчання всі параметри трансформера заморожені: він працює як готовий екстрактор мовних ознак, отриманих під час дистиляції із великої моделі BERT-Base. Це дозволяє швидко зійтися, адже мережа оптимізує лише легкі LoRA-доповнення в перехресній увазі та ваги Bi-GRU. Лише після того, як класифікатор стабілізувався, розморожуються два найвищі трансформер-шари й тонко налаштовуються з малою швидкістю навчання — так ми уникаємо катастрофічного забування загальномовних представлень і водночас даємо моделі змогу тонко підлаштуватися під риторику саме новинного жанру.

Дистильована природа TinyBERT означає, що він зберігає майже всю мовну компетентність «старшого» BERT-Base, утім містить у 7,5 разів менше параметрів і працює в рази швидше.

У підсумку спільний TinyBERT забезпечує потрібну «алфавітну» базу для всієї системи, дає компактне, але глибоке контекстуальне подання кожного токена і водночас зберігає ресурсну стриманість, без якої неможливо побудувати практичний, поширюваний інструмент автоматичної перевірки новин.

LoRA-адаптація. Після того як заголовок і основний текст пройшли через спільний TinyBERT, кожен їхній токен уже має повноцінний 312-вимірний контекстуальний вектор. Утім до цього моменту два фрагменти ще «не розмовляли» між собою: трансформер аналізував їх окремо. Щоб навчити модель бачити смисловий розрив або, навпаки, підтвердження між гучним заголовком і змістом новини, у мережу вводиться спеціальний шар перехресної уваги.

Ідея: береться матриця прихованих станів заголовка і трактуються як запити, тоді як матриця прихованих станів тіла новини розбивається на ключі та значення; за допомогою механізму scaled-dot-product обчислюються ваги, які показують, наскільки кожне слово заголовка «резонує» з кожним словом тіла. Потім ситуація дзеркально повторюється: уже токени тіла ставляться у позицію запитів, а заголовок — у позицію ключів і значень. Результатом є два потоки взаємної уваги, що буквально «зшивають» обидва тексти. На практиці цього двобічного зіставлення виявляється достатньо, щоб модель навчилася реагувати на класичний фейковий прийом: кричущий «Click-Bait» у заголовку, який ніяк не підтверджується далі у "новині". Двонаправлений механізм уваги описується:

Блок для напряму title → body

$$Q_t = H_t\left(W_Q^{tb} + \Delta W_Q^{tb}\right),$$

$$K_b = H_b\left(W_K^{tb} + \Delta W_K^{tb}\right),$$

$$U_b = H_b\left(W_U^{tb} + \Delta W_U^{tb}\right),$$

$$Attn_{t\to b} = softmax\left(\frac{Q_t K_b}{\sqrt{d_h}}\right) U_b \in R^{L_t \times d},$$

де $d_h = d/h = 156$, $h = 2$. У спеціалізованому блоці перехресної уваги між заголовком і тілом новини кількість голів самоуваги зменшено до 2 з метою зниження обчислювальних витрат. У цьому випадку знижується ризик перенасичення та градієнтного шуму. Зокрема, при $d_h = 156$ кожна голова самоуваги краще моделює семантику через збільшений d_h.

Напрям body → title оформлюється симетрично, після чого результати конкатенуються вздовж часової осі $F = \left[Attn_{t\to b}, Attn_{b\to t} \right] \in R^{(L_t+L_b)\times d}$.

Сирі параметри шарів *Q, K, U* та проєкції виходу залишаються замороженими — вони дісталися TinyBERT ще під час дистиляції й містять універсальні мовні закономірності. Підлаштовувати повну сотню тисяч ваг заради однієї локальної задачі недоцільно, тому використовується Low-Rank Adaptation. Сенс LoRA полягає в тому, що до кожної великої матриці додається крихітна добуткова матриця рангу 4; саме вона і стає єдиною «рухомою» частиною.

Кожна «велика» матриця (наприклад W_Q^{tb}) залишається фіксованою, оптимізується лише низькорангова надбудова $\Delta W = \alpha AB$, де $A \in R^{d\times r}$, $B \in R^{r\times d_h}$, $r = 4$. Тобто ранг $r = 4$ дає $d \times r + r \times d_h = 1872$ ваг замість $d \times d_h = 48672$ (економія 96,2 %). Множник масштабування беремо $\alpha = 32$ (стандарт для LoRA), що дозволяє зберігати норму градієнтів. Навчаються у нашому випадку лише *A*, *B* та біаси LayerNorm, решта ваг заморожена.

Після обчислення уваги обидва отримані тензори об’єднуються в одну довгу послідовність: це вже суцільна мапа, у якій кожне слово заголовка «знає», на які речення воно опирається, і навпаки. Додатковий LayerNorm і dropout (0,1) стабілізують статистику перед передаванням далі в Bi-GRU, який тепер бачить не два незалежні фрагменти, а єдиний, смислово узгоджений «діалог» між ними.

Таким чином шар cross-attention виконує одразу дві критичні функції: робить модель чутливою до смислової інконгруентності, що є ядром більшості фейкових публікацій, і дає можливість інтерпретувати рішення — теплову карту уваги легко візуалізувати й показати редактору, які саме слова заголовка «скандували» з якими пасажами тексту. І все це досягається без вибуху параметрів завдяки точковому донавчанню LoRA.

Блок Bi-GRU-64. Коли перехресна увага завершила «зшивання» заголовка й основного тексту, ми отримуємо довгу послідовність токенів, у якій кожен вектор уже містить інформацію не лише про власний контекст, а й про смисловий зв’язок із другою частиною новини. Однак для класифікації потрібен єдиний фіксований представник документа. Звести сотні токенів до кількох чисел можна різними способами, але просте усереднення або вибір [CLS]-вектора часто стирає причинно-наслідкові відтінки дискурсу. Саме тому наступним кроком ставимо компактний двонапрямний GRU з 64 прихованими елементами у кожному напрямі $h_k^{bi} = \left[\vec{h}_k, \overleftarrow{h}_k \right] \in R^{128}$.

GRU працює як «навчаємий пулер»: проходячи послідовність зліва направо і справа наліво, він послідовно акумулює важливу інформацію, а завдяки вбудованим воротам reset і update сам вирішує, які частини контексту варто зберегти, а від яких — відмовитися. Кожен крок оновлення ґрунтується лише на двох наборах параметрів замість трьох, потрібних LSTM, тож модель обходиться без зайвих обчислень. Вибір саме 64 прихованих елементів — це компроміс: їх достатньо, щоб запам’ятати

довготривале посилання між фразами, але кількість ваг не перевищує рамок «легковісної» архітектури.

На фінальному кроці ми застосовуємо глобальний max-пулінг по часовій осі $g_j = \max_{k=1,\dots,L} \left(h_{k,j}^{bi}\right),\ g \in R^{128}$.

Замість того щоб покладатися на останній стан, цей прийом «вибирає» найсильніше збуджені ознаки з усіх позицій, тож жодне ключове речення не губиться, навіть якщо воно розташоване далеко від кінця тексту. Додатковий dropout 0,2, накладений на результат, відсікає випадкові піки й запобігає перенавчанню.

У такий спосіб Bi-GRU виконує одразу декілька завдань. Він вводить порядковий контекст, якого бракує трансформерним шарам після «плоского» self-attention, концентрує увагу на найзначущіших фразах і, головне, робить це з мінімальним обчислювальним слідом. У підсумку ми отримуємо 128-вимірний вектор, який містить і лінгвістичний, і структурний «профіль» новини — саме з ним працює класифікаційний шар.

Класифікаційний шар (класифікаційна «голова»). Після того як двонапрямний GRU стискає всю «новину» у компактний 128-вимірний вектор g, настає етап остаточного прийняття рішення: чи можна вважати публікацію фейковою, чи вона має ознаки достовірної. Цю функцію виконує мінімалістична, але достатня класифікаційна голова, використання якої підтримує загальну «легковісну» концепцію всієї моделі.

Спершу вектор g надходить на шар Dropout із ймовірністю 0,2. Під час тренування вона випадково зануляє п’яту частину компонентів, тим самим привчаючи мережу не покладатися на окремі «галасливі» ознаки й запобігаючи перенавчанню на одиничних тригерах — наприклад, надмірних знак оклику чи словах-кліше.

Далі працює єдиний лінійний шар, отримуємо логіт z, що віддзеркалює сумарну «вагу доказів» на користь фейковості.

Щоб логіт перетворився на ймовірність, застосовується сигмоїда $\hat{y} = \frac{1}{1+e^{-z}}$, яка стискає будь-яке дійсне значення у діапазон (0, 1). Під час інференсу саме цю величину бачить користувач: наприклад, 0,83 читається як 83 % впевненості, що стаття неправдива. За замовчанням поріг відсічення — 0,5, однак після навчання модель калібрується на валідаційному наборі, підбираючи поріг, що максимізує F-міру. Калібрування дає змогу підігнати баланс précision/recall до вимог конкретної редакції: комусь важливіше уникати хибних звинувачень, комусь — не пропустити жодного фейку.

Для всієї моделі використовується Focal Binary Cross-Entropy:

$$\Psi = -\left(\left(1-\hat{y}\right)^{\gamma} y \log \hat{y} + \hat{y}^{\gamma}\left(1-y\right)\log\left(1-\hat{y}\right)\right), \quad \gamma = 2.$$

Таким чином класифікаційна голова завершує роботу всієї мережі, підтримуючи три ключові принципи: мінімальний ресурсний слід, можливість детального пояснення кожного рішення і гнучкість, що дає легко адаптувати поріг або калібрувати ймовірності під нові домени.

МЕТОДИКА ДОСЛІДЖЕННЯ

Процес навчання починається з підготовки корпусу Fake News Classification. CSV-файл стратифіковано ділиться на тренувальний, валідаційний та тестовий піднабори у пропорції 70 : 15 : 15, причому баланс класів «фейк/реальна» зберігається в кожній

частині. Для перевірки стабільності метрик на тому самому розрізі даних проводиться п’ятифольдова перехресна перевірка. Тренування моделей відбувалось на GPU Tesla T4.

Оптимізація ведеться алгоритмом AdamW із класичними моментами (0,9; 0,999) та $\varepsilon = 1 \cdot 10^{-8}$. Усі «важкі» матриці Frozen-TinyBERT не беруть участі в оновленні — градієнти через них не проходять, тому пам’ять на зберігання градієнтів істотно менша, ніж у сценарії повного fine-tune. Навчання відбувається у два етапи. На першому, який охоплює дві епохи, трансформер залишається в режимі екстрактора ознак: відкритими лишаються тільки LoRA-адаптери перехресної уваги, тонка проєкція перед GRU, сам GRU та лінійний класифікатор. Для цих шарів використовується відносно висока швидкість навчання $2 \cdot 10^{-4}$; вона дає змогу моделі швидко «підлаштуватися» під риторику новин, не торкаючись глибинної мовної бази TinyBERT. Після того, як крива втрат стабілізується, починається другий етап: розморожуються два верхні трансформерні блоки, але з меншою швидкістю $1 \cdot 10^{-5}$, тоді як інші шари продовжують оновлюватися початковим кроком. Завдяки такому freeze-schedule великий корпус знань TinyBERT не руйнується, а модель отримує мінімально необхідну свободу «відшліфувати» контекст під специфіку дискурсу.

Швидкість навчання в обох групах зменшується за схемою «короткий лінійний прогрів — косинусний спад». Перші десять відсотків ітерацій виконують роль warm-up, утримуючи градієнти від нестабільних стрибків; далі крок поступово тане до нуля, що забезпечує м’яке збіження. На кожному кроці втрати обчислюються за фокальною бінарною крос-ентропією: підйом показника γ придушує вплив уже правильно класифікованих прикладів і концентрує градієнт на важких, прикордонних зразках; згладжування етикеток $\varepsilon = 0{,}05$ відсікає випадкові сплески, викликані шумом у даних.

Для боротьби з перенавчанням використано кілька додаткових заходів. Кожен вектор після GRU проходить через dropout із імовірністю 0,2; ваги самих GRU-воріт підлягають mix-out-регуляризації 0,1, що частково заморожує старі значення й не дає моделі необмежено підлаштовуватись під дрібні артефакти тренсету. Градієнти за потреби кліпуються до норми 1,0. Навчання переривається достроково, якщо за дві поспіль епохи macro-F1 на валідації не покращилась.

Після останньої епохи зберігається чекпоінт із найкращою F-мірою. Поверх десяти фінальних станів оптимізатора проводиться SWAG-усереднення: ваги згладжуються, що дає додаткові десяті частки відсотка до стабільності. Нарешті калібрується поріг прийняття рішення: на валідації підбирається значення, що максимізує F1, і ця константа записується в конфігураційний файл.

У такий спосіб повний цикл «дані → модель» лишається доступним навіть на споживчому обладнанні, забезпечує відтворювані результати та дозволяє за лічені хвилини донавчати LoRA-адаптери під нові теми або локальні інформаційні ландшафти — усе це без ризику втратити загальні знання, закладені у дистильований TinyBERT.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розглянемо результати експериментального порівняння запропонованої моделі TinyBERT з Cross-Attention + LoRA + Bi-GRU, із класичними «легкими» методами машинного навчання — логістичною регресією, лінійним SVM, наївним байєсівським класифікатором Бернуллі, алгоритмом k-ближчих сусідів (k = 7), випадковим лісом, градієнтним бустингом та XGBoost. Порівнювати нашу модель з повними fine-tuning великими трансформерами, коли їх час тренування іноді вимірюється в годинах, немає

сенсу. Експерименти проводилися на корпусі англомовних новин Fake News Classification [16], розділеному на тренувальний, валідаційний і тестовий піднабори, із використанням метрик точності, середньозваженого F1-score та площі під ROC-кривою. Метою дослідження було визначити не тільки абсолютну ефективність кожного підходу у розпізнаванні фейкових і правдивих новин, але й оцінити компроміс між якістю класифікації та обчислювальними ресурсами під час донавчання. Інтерпретація отриманих результатів дозволяє зробити висновок про доцільність інтеграції трансформерного ядра з lightweight-адаптацією та послідовної обробки контексту задля підвищення точності та стабільності моделі у задачі класифікації текстових даних.

У ході експериментального порівняння різних підходів до класифікації фейкових новин наша модель продемонструвала найвищі показники серед усіх проаналізованих алгоритмів: точність (Accuracy) склала 99 %, середньозважений показник F1-score досяг 0,985, а площа під ROC-кривою (ROC AUC) становила 0,998. Результати інших моделей наведені у табл. 2.

Таблиця 2

Результати комп'ютерного експерименту

Model	Precision		Recall		F1-score		Accuracy	F1 Score	ROC AUC	Training Time (s)
	Real	Fake	Real	Fake	Real	Fake				
TinyBERT + Cross-Attention + LoRA + Bi-GRU	0.98	**0.99**	**0.99**	0.98	**0.99**	**0.99**	**0.99**	**0.985**	**0.998**	778.1
Logistic Regression	0.95	0.96	0.96	0.95	0.96	0.95	0.96	0.954	0.9907	106.2
Linear SVM	**0.99**	0.87	0.86	**0.99**	0.92	0.93	0.93	0.9291	0.9955	104.9
Bernoulli NB	0.87	0.92	0.93	0.86	0.9	0.89	0.89	0.8892	0.9585	**88.3**
K-NN (k=7)	0.88	0.81	0.8	0.89	0.84	0.84	0.84	0.8446	0.9252	**88.3**
Random Forest	0.96	0.97	0.97	0.95	0.96	0.96	0.96	0.962	0.9947	1983.7
Gradient Boosting	0.93	0.96	0.96	0.92	0.95	0.94	0.94	0.9404	0.9851	728.4
XGBoost	0.9	0.54	0.21	0.98	0.34	0.7	0.58	0.6964	0.5919	114.8

Аналіз метрик клас-специфічної точності підтверджує високу надійність запропонованої архітектури в обох категоріях. Для класу «реальні новини» модель досягла precision = 0,98 і recall = 0,99, а для «фейкових» — precision = 0,99 і recall = 0,98, що гарантує збалансованість та мінімум хибно-позитивних чи хибно-негативних рішень.

З точки зору обчислювальних витрат, навчання LoRA-адаптера разом із донавчанням Bi-GRU у запропонованій моделі потребувало близько 778 секунд. Хоча це значно довше за час тренування лінійних моделей (Logistic Regression і Linear SVM ≈ 105 с; Bernoulli NB і K-NN ≈ 88 с), але наша модель забезпечує вищу якість класифікації, як видно з табл. 2. Порівняно з повним fine-tuning великих трансформерів, коли час тренування іноді вимірюється в годинах, вибір на користь TinyBERT дозволив суттєво зменшити навантаження та скоротити часові витрати (778 с.).

Щодо inference, TinyBERT із крос-увагою та Bi-GRU забезпечує близько 5 мс на GPU Tesla T4, що робить нашу модель прийнятною для інтеграції в сервіси реального часу.

ISSN 2663 - 4023

Такий компроміс між точністю та ресурсними витратами робить TinyBERT + Cross-Attention + LoRA + Bi-GRU привабливим рішенням для завдань факт-чекінгу в реальному часі.

У контексті матриць заплутаності помітно, як класичні алгоритми по-різному справляються з балансом між хибно-позитивними та хибно-негативними помилками. Так, у випадку логістичної регресії (рис. 2) показано симетричні метрики, однак точність розпізнавання, гірша приблизно на 3%, ніж у запропонованої нами моделі. З мінусів моделі можна виділити те, що вона працює лише, якщо розподіл даних близький до лінійно-роздільного, а також не вловлює складних взаємодій н-грам.

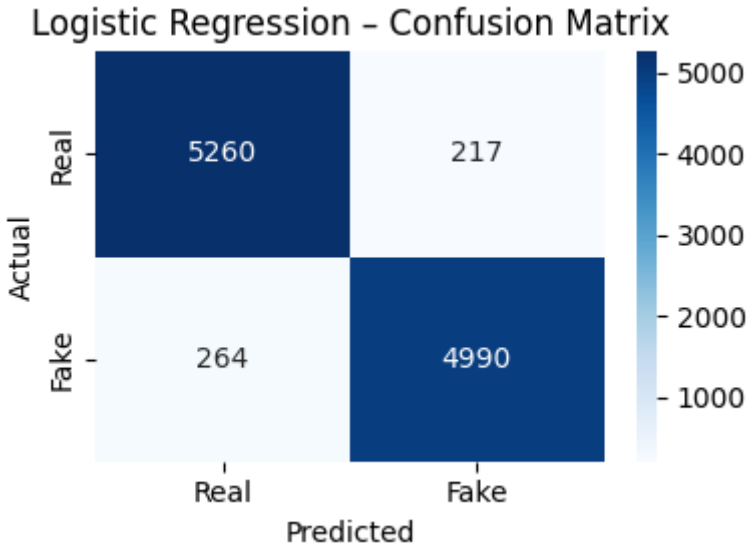
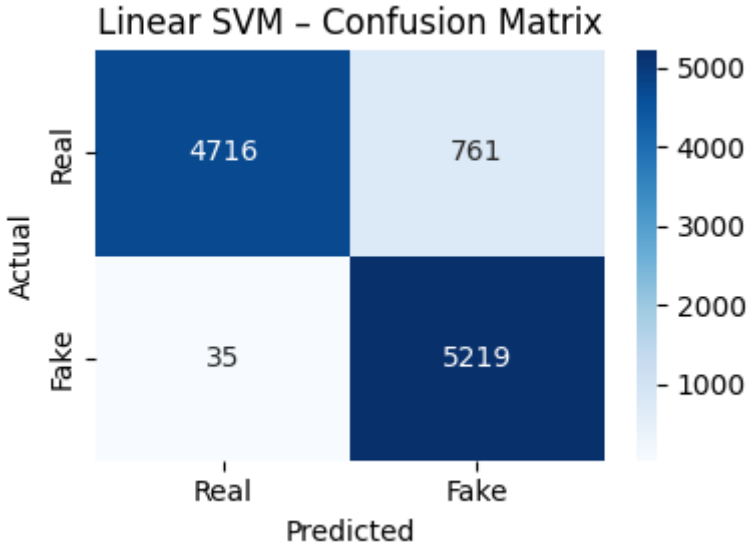


Рис. 2. Матриця заплутаності для логістичної регресії

Для Linear SVM (рис. 3) модель надійно не пропускає фейкові новини, але втрачає близько 14 % справжніх випадків, що відображено у значеннях хибно-негативних класифікацій у правому верхньому куті матриці.



ISSN 2663 - 4023

Рис. 3. Матриця заплутаності для Linear SVM

Bernoulli NB (рис. 4) демонструє майже рівномірний розподіл помилок, але загалом страждає від високо узагальнених й частково надмірних передбачень.

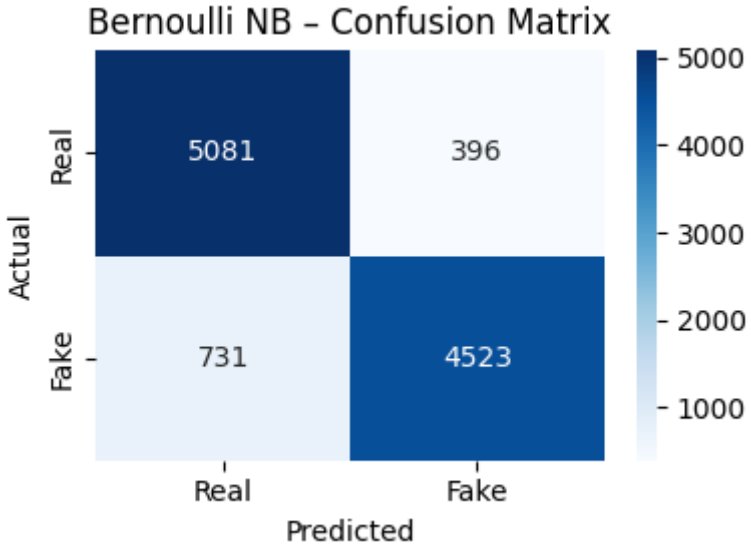


Рис. 4. Матриця заплутаності для Bernoulli NB

Тоді як K-NN із k=7 (рис. 5) часто плутає сусідні за ознаками приклади, що зумовлює розмиту діагональ.

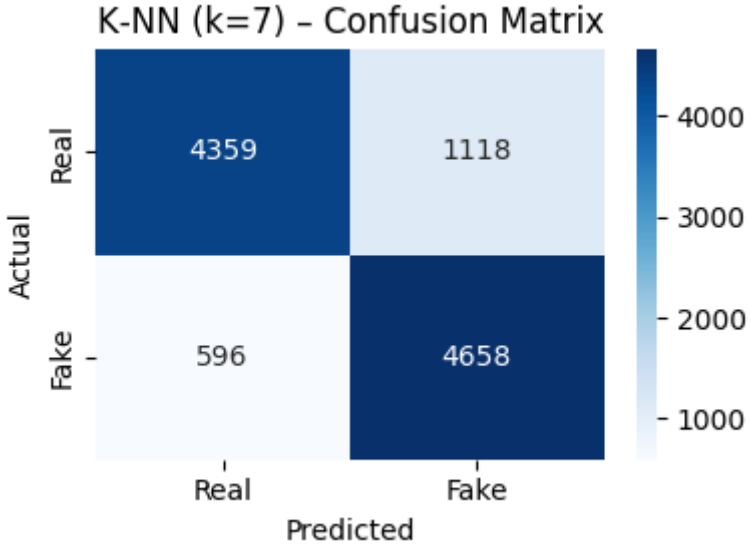


Рис. 5. Матриця заплутаності для K-NN

У випадку Random Forest (рис. 6) спостерігається значно чіткіша діагональ, проте окремі «справжні» пости іноді маркуються як фейкові через надмірну чутливість до кореляцій ознак, загалом модель показала точність на рівні моделі логістичної регресії 96%. Однак модель показала найдовший час навчання через велику кількість гіперпараметрів.

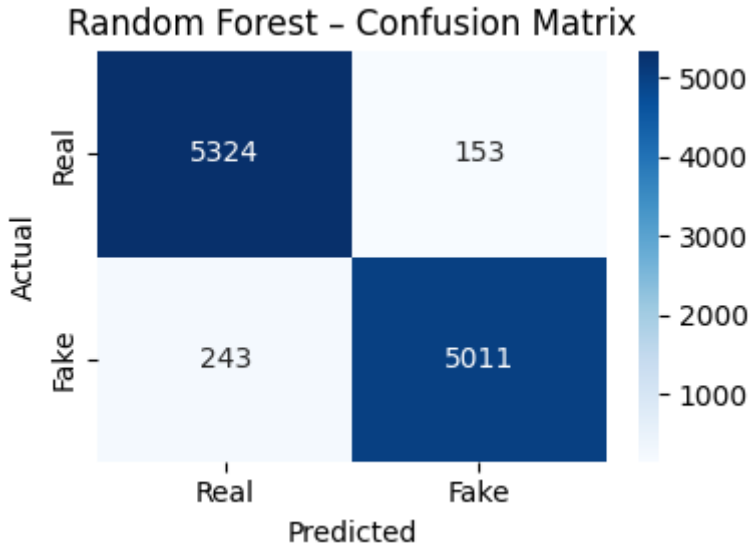


Рис. 6. Матриця заплутаності для Random Forest

Gradient Boosting (рис. 7) показує збалансовану поведінку, хоча модель не позбавлена хибно-позитивних відліків.

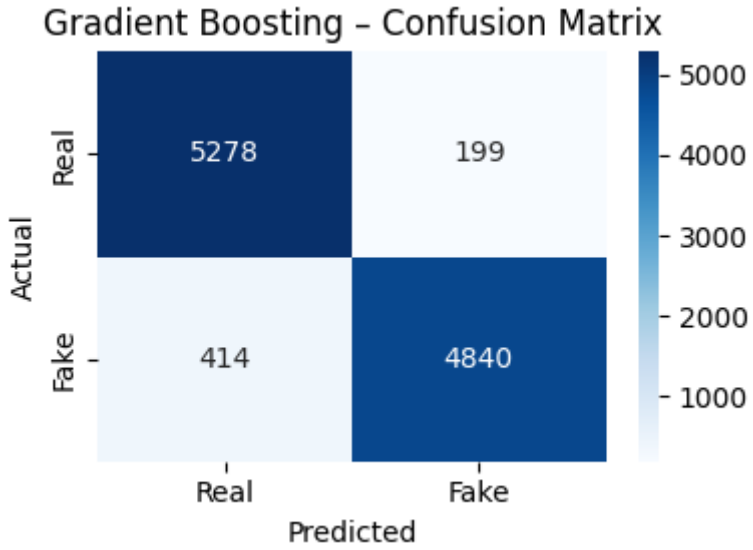


Рис. 7. Матриця заплутаності для Gradient Boosting

XGBoost (рис. 8) показала найгірший результат, приймає практично всі пости, як фейкові, але потрібно зазначити, що для тренувань використовувались базові гіперпараметри та модель не «тюнилась» під конкретну задачу.

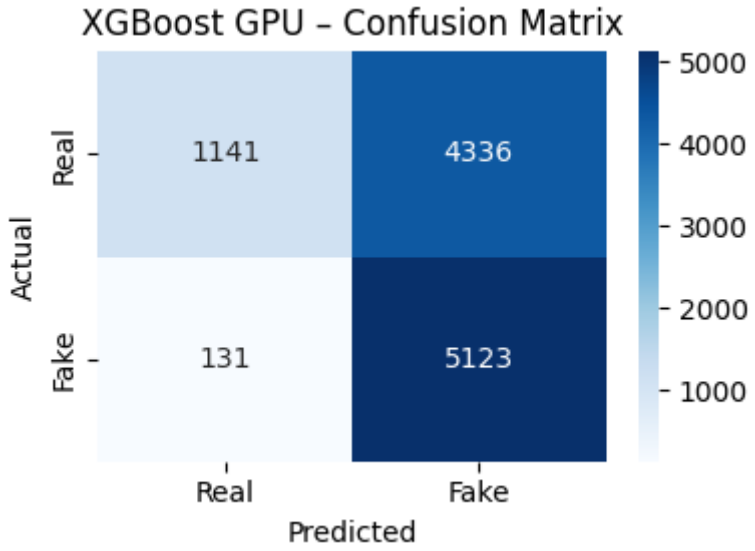


Рис. 8. Матриця заплутаності для XGBoost

Натомість матриця заплутаності для TinyBERT + Cross-Attention + LoRA + Bi-GRU (рис. 9) практично ідеальна: і справжні, і фейкові новини класифікуються правильно у 99 % випадків, з мінімальною кількістю хибно-позитивних та хибно-негативних рішень.

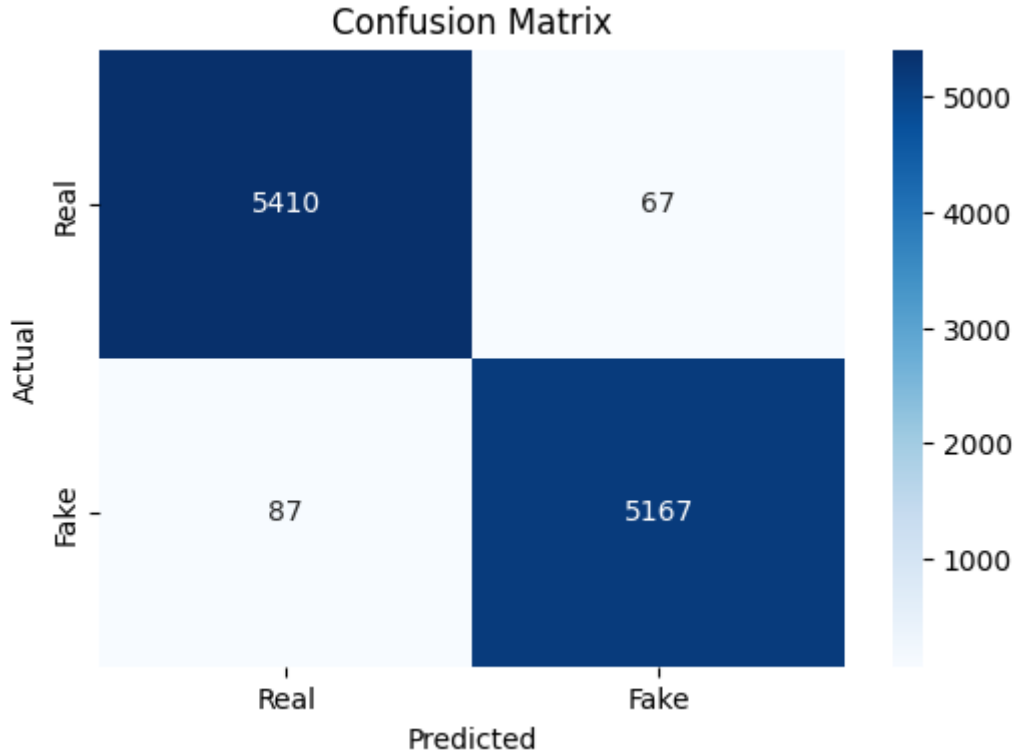


Рис. 9. Матриця заплутаності для TinyBERT + Cross-Attention + LoRA + Bi-GRU

Така картина свідчить про здатність комплексного підходу коректно інтерпретувати семантичні нюанси та послідовний контекст тексту, забезпечуючи майже повну відсутність втрачених або неправильно позначених прикладів.

Обмеження дослідження. Незважаючи на високі показники класифікації, дослідження має низку обмежень, які варто враховувати при інтерпретації результатів та подальшому розгортанні системи. По-перше, експериментальна оцінка базувалася винятково на англомовному корпусі Fake News Classification, тому переносимість моделі на тексти інших мов або культурних контекстів залишається невизначеною. По-друге, у задачі класифікації враховувалися лише текстові ознаки: мультимодальні дані, такі як зображення, відео чи графічні метадані, могли б додати критично важливі сигнали для відокремлення фейку від правди, але не були задіяні в поточному підході, хоча архітектура TinyBERT із cross-attention має високий потенціал для роботи з мультимодальними даними — об'єднання тексту з зображеннями, відео чи графічними метаданими могло б суттєво покращити виявлення фейку, однак це збільшить і ресурсозатратність. Також потрібно зазначити, що використовувався один GPU T4, що ускладнює масштабування дослідів на більших моделях або більшому обсязі даних.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

1. Запропонована компактна нейромережева архітектура TinyBERT з Cross-Attention, LoRA та Bi-GRU продемонструвала високу ефективність у задачі автоматичного розпізнавання фейкових новин. Отримані результати значно перевершують класичні моделі машинного навчання (логістичну регресію, SVM, наївний Байєсівський класифікатор Бернуллі, k-NN, випадковий ліс, градієнтний бустинг і XGBoost), досягнувши точності 99%, середньозваженого F1-score 0,985 та ROC AUC 0,998.
2. Основною перевагою розробленої моделі є здатність явно виявляти невідповідність між заголовком і текстом новини завдяки використанню механізму перехресної уваги (cross-attention). Такий підхід суттєво підвищує точність класифікації, особливо в контексті поширених клікбейтних заголовків, які традиційні нейромережеві моделі не завжди здатні коректно інтерпретувати.
3. Використання Low-Rank Adaptation (LoRA) дозволило значно скоротити обчислювальні ресурси, необхідні для донавчання моделі, зберігаючи при цьому її якість, що є важливою перевагою для практичного застосування моделі на пристроях з обмеженими обчислювальними потужностями.
4. Додавання двонапрямного рекурентного шару Bi-GRU до трансформерної основи забезпечує збереження порядкового контексту та ефективну агрегацію інформації про важливі семантичні особливості тексту, що позитивно впливає на якість класифікації та інтерпретованість результатів.
5. Практичні переваги запропонованої моделі включають її придатність для інтеграції у реальні сервіси факт-чекінгу завдяки низькому часу інференсу (5 мс на GPU Tesla T4), можливість швидкого донавчання під нові тематики та мовні контексти, а також високий рівень інтерпретованості рішень через теплові карти уваги та Grad-CAM.
6. Порівняння матриць заплутаності показує, що запропонована модель ефективно балансувала між точністю та повнотою класифікації, демонструючи мінімальну кількість помилкових рішень у порівнянні з іншими алгоритмами.

Таким чином, запропонована модель TinyBERT з Cross-Attention + LoRA + Bi-GRU є високоефективним та ресурсно-оптимізованим рішенням, придатним для інтеграції в системи автоматичного виявлення фейкових новин у реальному часі, із значним потенціалом для подальшого розвитку та адаптації до нових умов застосування.

Подальший розвиток запропонованої архітектури передбачає розширення її можливостей та підвищення ефективності в умовах реальних сценаріїв застосування. Перспективним напрямком є проведення експериментів з мультимодальними даними, що включатимуть об'єднання текстової інформації з додатковими джерелами, такими як зображення, відео та графічні метадані. Інтеграція цих модальностей у модель (наприклад, через механізми cross-modal attention) дозволить підвищити точність класифікації та забезпечить глибший аналіз змісту публікацій, зокрема у випадках, коли фейковий контент супроводжується маніпулятивними візуальними матеріалами.

Ще одним важливим напрямком є оптимізація роботи моделі для портативних пристроїв та вбудованих систем. Планується дослідити застосування різних технологій квантування для зменшення обчислювальних витрат і скорочення часу інференсу без суттєвої втрати точності. Зокрема, передбачається тестування post-training INT8-квантування для всієї моделі, а також використання низькорозрядних форматів FP8 або INT4 для обчислювально-інтенсивних компонентів. Реалізація цих підходів із залученням платформно-специфічних інструментів — TensorFlow Lite з NNAPI для Android, Core ML для iOS та ONNX Runtime Mobile для ARM-пристроїв — дозволить досягти прискорення роботи моделі та зменшення споживання пам'яті.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yoon, S., et al. (2021). Learning to detect incongruence in news headline and body text via a graph neural network. *IEEE Access, 9,* 36195–36206. https://doi.org/10.1109/access.2021.3062029
2. Jiao, X., et al. (2020). TinyBERT: Distilling BERT for natural language understanding. *Findings of the Association for Computational Linguistics: EMNLP 2020.* https://doi.org/10.18653/v1/2020.findings-emnlp.372
3. Ye, H., et al. (2025). LoRA-Adv: Boosting text classification in large language models through adversarial low-rank adaptations. *IEEE Access,* 1. https://doi.org/10.1109/access.2025.3579539
4. Taher, Y., Moussaoui, A., & Moussaoui, F. (2022). Automatic fake news detection based on deep learning, FastText and news title. *International Journal of Advanced Computer Science and Applications, 13*(1). https://doi.org/10.14569/ijacsa.2022.0130118
5. Mishra, R., & Setty, V. (2019). SADHAN: Hierarchical attention networks to learn latent aspect embeddings for fake news detection. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '19)* (pp. 197–204). ACM. https://doi.org/10.1145/3341981.3344229
6. Shu, K., et al. (2020). Hierarchical propagation networks for fake news detection: Investigation and exploitation. *Proceedings of the International AAAI Conference on Web and Social Media, 14,* 626–637. https://doi.org/10.1609/icwsm.v14i1.7329
7. Chang, Q., Li, X., & Duan, Z. (2024). Graph global attention network with memory: A deep learning approach for fake news detection. *Neural Networks,* 106115. https://doi.org/10.1016/j.neunet.2024.106115
8. Qawasmeh, E., Tawalbeh, M., & Abdullah, M. (2019). Automatic identification of fake news using deep learning. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 1–7). IEEE. https://doi.org/10.1109/snams.2019.8931873
9. Dun, Y., et al. (2021). KAN: Knowledge-aware attention network for fake news detection. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(1), 81–89. https://doi.org/10.1609/aaai.v35i1.16080
10. Alghamdi, J., Lin, Y., & Luo, S. (2024). Enhancing hierarchical attention networks with CNN and stylistic features for fake news detection. *Expert Systems with Applications,* 125024. https://doi.org/10.1016/j.eswa.2024.125024
11. Zhang, W., et al. (2024). Cross-attention multi-perspective fusion network-based fake news censorship. *Neurocomputing,* 128695. https://doi.org/10.1016/j.neucom.2024.128695
12. Maham, S., et al. (2024). ANN: Adversarial news net for robust fake news classification. *Scientific Reports, 14*(1). https://doi.org/10.1038/s41598-024-56567-4

13. Zhou, D., et al. (2024). GS2F: Multimodal fake news detection utilizing graph structure and guided semantic fusion. *ACM Transactions on Asian and Low-Resource Language Information Processing*. https://doi.org/10.1145/3708536
14. Kuntur, S., et al. (2024). Under the influence: A survey of large language models in fake news detection. *IEEE Transactions on Artificial Intelligence*, 1–21. https://doi.org/10.1109/tai.2024.3471735
15. Kabir, M. M., et al. (2025). CUET-NLP_MP@DravidianLangTech 2025: A transformer and LLM-based ensemble approach for fake news detection in Dravidian. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages* (pp. 420–426). ACL. https://doi.org/10.18653/v1/2025.dravidianlangtech-1.75
16. Kaggle. (2023, October 8). *Fake news classification.*

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (29), 2025
ISSN 2663 - 4023

Ivan Peleshchak
PhD in Information Technologies, Associate Professor of the Department of Information Systems and Networks
Lviv Polytechnic National University, Lviv, Ukraine
ORCID ID: 0000–0002–7481–8628
ivan.r.peleshchak@lpnu.ua

Vasyl Lytvyn
Doctor of Technical Sciences, Professor, Professor of the Department of Information Systems and Networks
Lviv Polytechnic National University, Lviv, Ukraine
ORCID ID: 0000–0002–9676–0180
vasyl.v.lytvyn@lpnu.ua

Victoria Vysotska
Doctor of Technical Sciences, Associate Professor, Professor of the Department of Information Systems and Networks
Lviv Polytechnic National University, Lviv, Ukraine
ORCID ID: 0000-0001-6417-3689
victoria.a.vysotska@lpnu.ua

Oleksii Khobor
Postgraduate Student of the Department of Information Systems and Networks
Lviv Polytechnic National University, Lviv, Ukraine
ORCID ID: 0000–0002–3210–036X
oleksii.r.khobor@lpnu.ua

TINYBERT WITH CROSS-ATTENTION + LORA + BI-GRU: A COMPACT NEURAL NETWORK ARCHITECTURE FOR FAKE NEWS DETECTION

Abstract. This paper proposes a compact neural network architecture based on TinyBERT with a Cross-Attention mechanism, Low-Rank Adaptation (LoRA), and a bidirectional recurrent Bi-GRU layer for automatic fake news detection. The rapid development of the digital information space has led to a sharp increase in the amount of disinformation, while traditional fact-checking methods can no longer effectively counter this trend. Existing neural network approaches based on large transformer models are often unsuitable for practical use due to their high computational cost and inability to explicitly analyze semantic inconsistencies between a news headline and its body text. The proposed model addresses these limitations by employing a distilled TinyBERT transformer, which significantly reduces computational requirements, and a specialized Cross-Attention module that provides sensitivity to headline–body discrepancies. The use of LoRA adapters minimizes the number of trainable parameters, speeding up the fine-tuning process, while the integration of Bi-GRU enables the preservation of contextual information in sequences. Experimental evaluation on the Fake News Classification dataset demonstrated that the model outperforms classical machine learning algorithms, achieving an accuracy of 99%, an F1-score of 0.985, and a ROC AUC of 0.998. Due to its low resource consumption, rapid adaptability to new topics, and high interpretability of decisions, the model shows strong potential for integration into real-time fact-checking services.

Keywords: fake news detection; TinyBERT; cross-attention; LoRA; Bi-GRU; natural language processing; transformer models; AI fact-checking.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Yoon, S., et al. (2021). Learning to detect incongruence in news headline and body text via a graph neural network. *IEEE Access, 9,* 36195–36206. https://doi.org/10.1109/access.2021.3062029
2. Jiao, X., et al. (2020). TinyBERT: Distilling BERT for natural language understanding. *Findings of the Association for Computational Linguistics: EMNLP 2020.* https://doi.org/10.18653/v1/2020.findings-emnlp.372
3. Ye, H., et al. (2025). LoRA-Adv: Boosting text classification in large language models through adversarial low-rank adaptations. *IEEE Access,* 1. https://doi.org/10.1109/access.2025.3579539

4. Taher, Y., Moussaoui, A., & Moussaoui, F. (2022). Automatic fake news detection based on deep learning, FastText and news title. *International Journal of Advanced Computer Science and Applications, 13*(1). https://doi.org/10.14569/ijacsa.2022.0130118
5. Mishra, R., & Setty, V. (2019). SADHAN: Hierarchical attention networks to learn latent aspect embeddings for fake news detection. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '19)* (pp. 197–204). ACM. https://doi.org/10.1145/3341981.3344229
6. Shu, K., et al. (2020). Hierarchical propagation networks for fake news detection: Investigation and exploitation. *Proceedings of the International AAAI Conference on Web and Social Media, 14*, 626–637. https://doi.org/10.1609/icwsm.v14i1.7329
7. Chang, Q., Li, X., & Duan, Z. (2024). Graph global attention network with memory: A deep learning approach for fake news detection. *Neural Networks*, 106115. https://doi.org/10.1016/j.neunet.2024.106115
8. Qawasmeh, E., Tawalbeh, M., & Abdullah, M. (2019). Automatic identification of fake news using deep learning. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 1–7). IEEE. https://doi.org/10.1109/snams.2019.8931873
9. Dun, Y., et al. (2021). KAN: Knowledge-aware attention network for fake news detection. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(1), 81–89. https://doi.org/10.1609/aaai.v35i1.16080
10. Alghamdi, J., Lin, Y., & Luo, S. (2024). Enhancing hierarchical attention networks with CNN and stylistic features for fake news detection. *Expert Systems with Applications*, 125024. https://doi.org/10.1016/j.eswa.2024.125024
11. Zhang, W., et al. (2024). Cross-attention multi-perspective fusion network-based fake news censorship. *Neurocomputing*, 128695. https://doi.org/10.1016/j.neucom.2024.128695
12. Maham, S., et al. (2024). ANN: Adversarial news net for robust fake news classification. *Scientific Reports, 14*(1). https://doi.org/10.1038/s41598-024-56567-4
13. Zhou, D., et al. (2024). GS2F: Multimodal fake news detection utilizing graph structure and guided semantic fusion. *ACM Transactions on Asian and Low-Resource Language Information Processing*. https://doi.org/10.1145/3708536
14. Kuntur, S., et al. (2024). Under the influence: A survey of large language models in fake news detection. *IEEE Transactions on Artificial Intelligence*, 1–21. https://doi.org/10.1109/tai.2024.3471735
15. Kabir, M. M., et al. (2025). CUET-NLP_MP@DravidianLangTech 2025: A transformer and LLM-based ensemble approach for fake news detection in Dravidian. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages* (pp. 420–426). ACL. https://doi.org/10.18653/v1/2025.dravidianlangtech-1.75
16. Kaggle. (2023, October 8). *Fake news classification.*