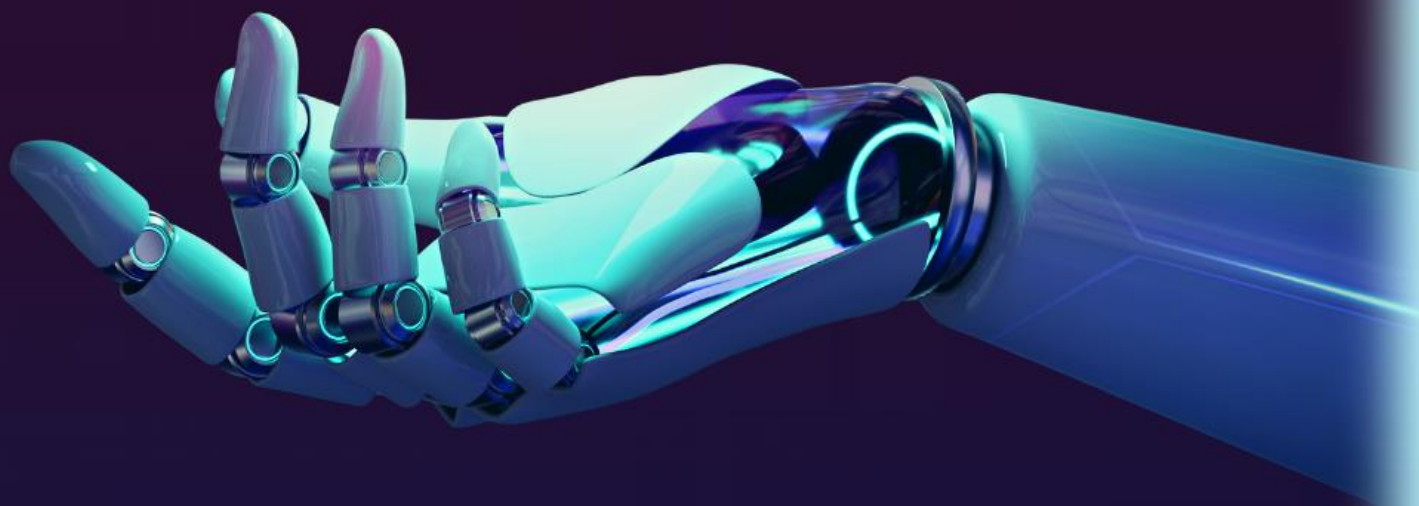


ISSN 2075-4272 (Print),
2786-9024 (Online)



**Наукові праці
Донецького національного технічного університету.
Серія: «Обчислювальна техніка та автоматизація»**



T.2, №2 (34), 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАУКОВІ ПРАЦІ
ДОНЕЦЬКОГО НАЦІОНАЛЬНОГО
ТЕХНІЧНОГО УНІВЕРСИТЕТУ**

*Серія: «Обчислювальна техніка
та автоматизація»*

Всеукраїнський науковий збірник

Заснований у липні 1998 року

Виходить 1-2 рази на рік

Т2. № 2(34)'2024

Дрогобич

2024

УДК 681.5: 658.5: 621.3

Друкується за рішенням Вченої ради Державного вищого навчального закладу «Донецький національний технічний університет» (протокол №10 від 31.10.2024 р.).

У збірнику опубліковано статті науковців, аспірантів, магістрів та інженерів провідних підприємств і закладів вищої освіти України, у яких наведено результати наукових досліджень та розробок, виконаних у 2023-2024 рр. відповідно до напрямків: автоматизація технологічних процесів, інформаційна безпека, інформаційно-вимірювальні системи, електронні та мікропроцесорні прилади, інформаційні технології, кібербезпека та захист критичної інфраструктури, математичне та комп'ютерне моделювання, телекомунікаційні системи та мережі.

Матеріали збірника призначено для викладачів, наукових співробітників, інженерно-технічних працівників, аспірантів і студентів, які досліджують питання інформаційної безпеки, розробки та впровадження інформаційних систем та технологій, розробки й використання автоматичних, інформаційних та електронних систем.

Засновник та видавець – Донецький національний технічний університет.

Редакційна колегія: Дорогий Я.Ю., д-р техн. наук, доц., головний редактор (Україна); Воропаєва В.Я., канд. техн. наук, доц., заст. головного редактора, відп. за випуск (Україна); Башков Є.О., д-р техн. наук, проф. (Україна); Лактіонов І.С., д-р техн. наук, доц. (Україна); Дмитрієва О.А., д-р техн. наук, проф. (Україна); Святний В.А., д-р техн. наук, проф. (Україна); Ямненко Ю.С., д-р техн. наук, проф. (Україна); Кучерук В.Ю., д-р техн. наук, проф. (Україна); Василець С.В., д-р техн. наук, проф. (Україна); Гільгурт С.Я., д-р техн. наук, ст. дослід. (Україна); Баркалов О.О., д-р техн. наук, проф. (Польща); Різун Н.О., д-р техн. наук, проф. (Польща); Цуркан В.В., канд. техн. наук, доц. (Україна); Бакалинський О.О., канд. техн. наук, ст. дослід. (Україна).

Ідентифікатор медіа R30-02474 відповідно з додатком до Рішення НРУ з питань телебачення і радіомовлення №139 від 18.01.2024 р.

Збірник включено до списку друкованих (електронних) періодичних наукових фахових видань України, у яких можуть публікуватися результати дисертаційних робіт на здобуття наукових ступенів доктора й кандидата наук (спеціальності 151, 152, наказ Міністерства освіти і науки України № 886 від 02.07.2020 р.).

ISSN 2075-4272 (Print),

ISSN 2786-9024 (Online)

© Донецький національний технічний університет, 2024

ЗМІСТ

РОЗДІЛ 1. ТЕЛЕКОМУНІКАЦІЙНІ СИСТЕМИ ТА МЕРЕЖІ

Чистик І.М., Воропаєва В.Я. Дослідження ефективності роботи мережевих систем управління 4-13

РОЗДІЛ 2. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Байтєльман Я.Л., Поцєнаєв В.В. Розробка системи управління контентом та її розгортання на хмарному середовищі 14-31

Рожко А.М., Дорогий Я.Ю., Тєличко Г.О. Інформаційна система обліку відвідуваності студентів 32-38

РОЗДІЛ 3. ШТУЧНИЙ ІНТЕЛЕКТ

Ковалєнко В.М., Дорогий Я.Ю., Дорошенко К.С. Компаративний аналіз моделей для виявлення та класифікації фейкових новин з використанням GRU 39-57

РОЗДІЛ 4. КІБЕРБЕЗПЕКА ТА ЗАХИСТ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

Гільгурт С.Я. Оцінка апаратних витрат для фільтра Блума з лічильниками в системах захисту інформації на ПЛІС 58-69

CONTENT

CHAPTER 1. TELECOMMUNICATION SYSTEMS AND NETWORKS

Ivan Chystyk, Viktoriya Voropayeva. Research of the efficiency of networked control systems 4-13

CHAPTER 2. INFORMATION TECHNOLOGY

Yakiv (Jacob) Baytelman, Valerii Potsepaiev. Development of a content management system and its cloud deployment 14-31

Alina Rozhko, Iaroslav Dorohyi, Hanna Telychko. Information system for recording student attendance 32-38

CHAPTER 3. ARTIFICIAL INTELLIGENCE

Vitalii Kovalenko, Iaroslav Dorohyi, Kateryna Doroshenko. Comparative analysis of models for fake news detection and classification using GRU 39-57

CHAPTER 4. CYBERSECURITY AND CRITICAL INFRASTRUCTURE PROTECTION

Serhii Hilgurt. Estimation of counting Bloom filter hardware costs for FPGA-based cybersecurity systems 58-69

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОБОТИ МЕРЕЖЕВИХ СИСТЕМ УПРАВЛІННЯ

І.М. Чистик¹, В.Я. Воропаєва¹

¹ Department of Automation and Telecommunications, Donetsk National Technical University, Luts'k, Ukraine

E-mail: ivan.chystyk@donntu.edu.ua

Отримано 15.04.2024

Прийнято до публікації 01.08.2024

Опубліковано 01.11.2024

АНОТАЦІЯ

Мета даної роботи визначається необхідністю у детальному вивченні та аналізі можливих проблем і недоліків, що виникають, або можуть виникнути при створенні мережесистем управління (NCS). Кожен із визначених негативних чинників має свій вплив і виникає в різних ситуаціях, має свої особливості і різний характер поведінки і походження. Оскільки детальне вивчення роботи мережесистем контролю зазвичай супроводжується процесом моделювання такої системи, то для адекватності і точності процесу моделювання необхідно створити правильний апарат, який враховує ці показники.

Проаналізована інформація із різних джерел та розглянуті матеріали дозволили створити уявлення про основні види і типи негативних факторів і недоліків, що супроводжують роботу мережесистем управління. Такий підхід дозволяє швидше і більш направлено реагувати на причину виникнення можливих затримок у роботі мережесистем управління, які викликані визначеними обмеженнями. Враховуючи, що показники швидкості і точності при роботі систем управління – одні з ключових, то справедливо буде вважати, що наміри по покращенню ситуації з негативним впливом таких чинників є доречним і необхідним шляхом. Оскільки NCS – це складна система, що містить у собі елементи різних рівнів, то визначено також і саму сутність визначених проблем, тобто з'ясовано не тільки причини появи і виникнення цих факторів, але і що вони собою представляють, на що впливають і до яких наслідків це може привести у тих чи інших ситуаціях і за різних умов.

Враховуючи можливу наявність таких негативних чинників у мережесистемах управління, логічним виглядає спроба визначити також і підходи і методи, що можуть бути застосовані при зменшенні або нейтралізації того чи іншого обмеження в різних можливих умовах. Розуміючи можливість наявності негативних чинників та недосконалостей при роботі мережесистем управління, ми маємо можливість зіграти на випередження та закласти під час проектування і побудови такої системи елементи, що дозволять покращити показники роботи системи. Стосовно цих підходів також надана інформація про умови, в яких вони можуть бути застосовані.

Ключові слова: мережесистеми управління, моделювання, NCS, контроль, стабільність роботи

ВСТУП

Розвиток людства останні роки активно супроводжувався технічним прогресом, в результаті якого з'явилися нові можливості для покращення нашого життя. Таким чином, все активніше почали створюватися цілі системи, які відповідають за керування, автоматизацію і контроль різних процесів. Для повноцінного задоволення потреб, ці системи весь час збільшуються, а Інтернет стає невід'ємною їх частиною, ці системи можуть поєднуватися і взаємодіяти між собою.

Тривалий час при використанні комунікаційних мереж, які були орієнтовані на управління, в якості першого вибору виступали Control Area Network і DeviceNet в перших прообразах мережевих систем управління. Розвиток мережевих технологій, зростання потреб у великомасштабних системах зробили Інтернет дуже вигідним альтернативним рішенням. Використання Інтернет надає можливість створення великої системи управління зі зменшенням витрат та більш простим обслуговуванням, а також надає гнучкі можливості для внесення змін до конфігурації самої системи.

Враховуючи всі ці особливості, почали привертати увагу на використання мережевих технологій при побудові таких систем. З іншого боку, вдосконалення самих мережевих технологій, збільшення пропускну здатності, та удешевлення, зробили їх ще привабливішими для використання. Завдяки таким вагомим здобуткам в області мережевих систем управління, з'являється додаткова можливість для появи і розвитку таких інновацій, як промисловий контроль, дослідження небезпечної середовища, рятувальні і пошукові операції, аерокосмічні системи, мікромережі та телеуправління, а також використання в управлінні БПА та багато інших. І таким чином, починаючи з 80-х років минулого сторіччя, з'являються перші згадування про мережеві системи управління. Таким чином, в процесі розвитку даної технології, її більш глибокого дослідження, почали з'являтися нові види і класи мережевих систем контролю, сфери застосування і призначення яких, а також їх характеристики, мали значні відмінності.

АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ ТА ПОСТАНОВКА ПРОБЛЕМИ

Мережева система управління — це система, у якій традиційні контури управління замкнуті через мережу зв'язку, отже сигналами системи можна обмінюватися між компонентами через загальну мережу. Мережеві системи управління зазвичай складаються з великої кількості взаємозалежних пристроїв.

Справедливо буде зауважити, що для дослідження ефективності роботи будь-якої системи управління, в тому числі і мережевих систем управління, необхідно визначити ряд відомих проблем і особливостей технології. Слід зазначити, що визначені недосконалості обов'язково мають бути враховані при подальшому моделюванні NCS, і у випадку з мережевими системами управління, ці проблеми можна виділити і визначити головні та найбільш доцільні для вивчення з метою поліпшення їх негативного впливу.

Включення мереж до динамічних систем створює нові проблеми для всієї системи через появу недосконалостей. До них відносяться помилки квантування, різні затримки, пропуск кадрів і т. д. Ці недосконалості мають істотний вплив на стан мережевої системи управління, можуть погіршити її продуктивність і навіть викликати нестабільність у її роботі. Саме тому є велика необхідність у створенні відповідного динамічного уявлення про NCS і знаходження способу, що дасть можливість досягти стабільності за цих обставин.

Дослідження мережевих систем управління зазвичай поділяються на два напрями:

(1) Управління мережею: дослідження та покращення внутрішніх характеристик мереж, наприклад, пропозиція нових мережевих протоколів зв'язку, алгоритмів мережевого планування тощо.

(2) Контроль над мережею: розгляд існуючих мережевих структур, протоколів в якості встановлених умов, на основі яких можуть бути розроблені розумні структури та алгоритми управління для компенсації або зменшення негативного впливу на систему управління[1]

Управління мережею розглядає проблеми мережі зв'язку, такі як протоколи зв'язку, управління маршрутизацією, управління навантаженням і т.д. В іншому випадку управління через мережу фокусується на проектуванні та управлінні системами, які використовують мережу як середовище передачі для отримання бажаної продуктивності.

Показники мережі, такі як швидкість передачі та частота помилок, залежать від QoS, тоді як QoS стосується стабільності системи у різних умовах.

Формулювання проблеми: в даний час мережеві системи управління все активніше використовуються в різних сферах нашої діяльності. Переваги, що несуть у собі ці системи занадто привабливі, щоб їх ігнорувати, проте не потрібно забувати, що як і кожна технологія, винахід, або готовий пристрій, мережеві системи управління мають певний перелік особливостей, які необхідно враховувати і обмеження, які необхідно покращити задля забезпечення максимальної ефективності від використання.

Мета: визначення основних типів і видів недоліків та обмежень мережевих систем управління, їх походження та причин виникнення, а також показників, на які ці недоліки і обмеження впливають при роботі мережевих систем управління.

Задачі:

1. Визначення основних видів і недоліків мережевих систем управління.
2. Аналіз природи походження недоліків і обмежень NCS.
3. Визначення основних підходів, що використовуються для вирішення проблеми наявності цих недосконалостей NCS.
4. Підготовка матеріалів для представлення підсумкового результату та візуальних матеріалів.

МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ

В даній роботі було використано методи структурного і порівняльного аналізу. Було досліджено наукову і методичну літератури з використанням інформаційного і аналітичного підходу, також використано ряд онлайн-ресурсів для отримання інформації по тематиці дослідження.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для того, щоб було зручно візуально сприймати вплив факторів, що впливають на роботу мережевої системи управління, це можна зобразити у вигляді замкненої системи управління, в якій присутній мінімальний набір необхідних елементів для опису цього процесу – контролер, установка(привід, інший виконавчий орган або елемент), а також безпосередньо сама середа, через яку виконується обмін інформацією між елементами – в даному випадку мережа. Умовно розміщено всі фактори, що мають вплив в контур мережі. Компоненти NCS пов'язані між собою через системи зв'язку, як показано на Рис. 1, і таке з'єднання показує вплив недоліків і обмежень мережевої системи. Одне із завдань контролера в даному випадку полягає у тому, щоб якщо ми маємо у наявності один або навіть декілька негативних факторів із тих, що були представлені на малюнку, компенсувати його так, щоб система не виходила за межі своїх заданих параметрів стабільної роботи.

Далі в даній роботі буде більш детально розглянуто кожний із цих негативних чинників, а також підходи, що застосовуються під час вирішення проблеми впливу них на систему. Недосконалості та обмеження у NCS поділяються на п'ять типів[2]:

- 1) помилки квантування під час передачі сигналів;

- 2) випадання пакетів через ненадійну передачу;
- 3) змінні інтервали вибірки/передачі;
- 4) змінні затримки передачі;
- 5) обмеження зв'язку, оскільки не всі сигнали датчиків та виконавчих пристроїв можуть передаватися одночасно.

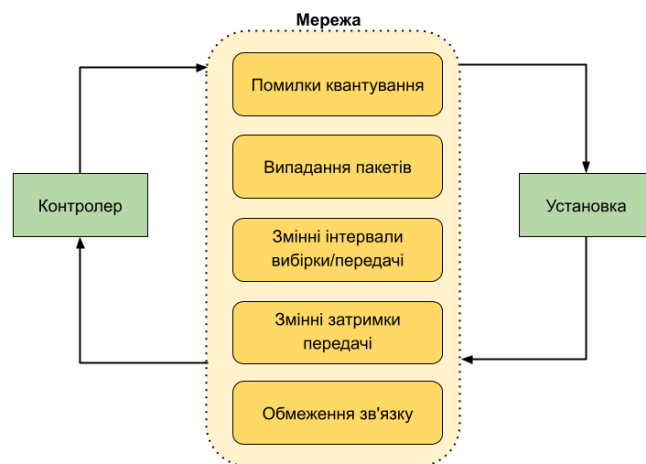


Рис. 1. Типи недоліків і обмежень в NCS

ПОМИЛКИ КВАНТУВАННЯ

Сутність мережі, що використовуються для зв'язку компонентів, має ряд характеристик і особливостей, в число яких входить і обмежена пропускна здатність, і виникає потреба у квантуванні сигналу перед відправкою.

Таким чином, перед відправкою в мережу сигнал керування та вихідний сигнал об'єкта квантуються, як показано на Рис. 2. Квантизатор — це пристрій, який приймає дійсний сигнал і перетворює його на частково-постійний з кінцевим набором значень.

В науковій літературі прийнято розділяти квантування на два типи: логарифмічне квантування і рівномірне квантування[3].

- Логарифмічне квантування розглядається як статичне квантування, і характеризується кращою продуктивністю поруч з початком координат порівняно з рівномірним квантуванням. Також характеризується або безкінечним рівнем квантування, або кінцевим рівнем. не використовуватиметься в налаштуваннях DNS, тож вона потрібна лише для зручності адміністрування.
- Однорідний рівномірний квантизатор характеризується більш простим керуванням, і надає можливості застосування з кількісною

областю довільної форми. Логарифмічні квантизатори переважно використовуються з лінійними системами з безкінечними рівнями квантування. При цьому використання

масштабування є корисним елементом управління, коли застосовується рівномірне квантування, і воно складається з двох етапів: «збільшення» та «зменшення».

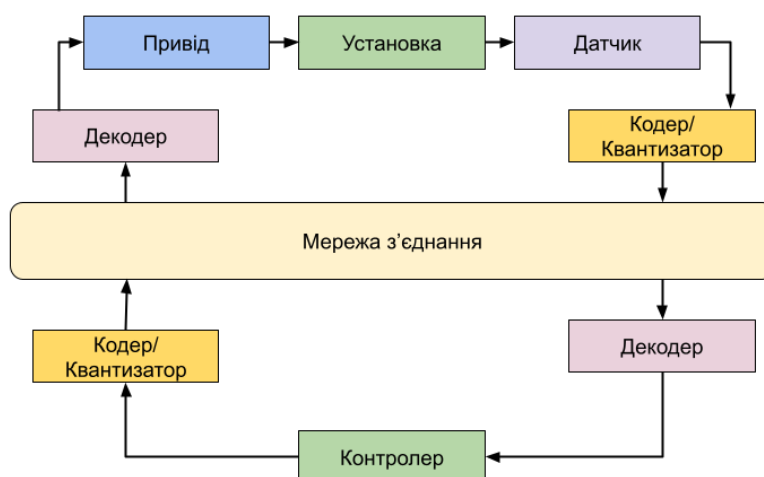


Рис.2. Конфігурація NCS з квантуванням

Реалізація збільшення та зменшення масштабу дає можливість отримати достатню умову асимптотичної стійкості лінійних та нелінійних систем. Внаслідок квантування в системі будуть внесені втрати інформації. Тому створена модель такої системи має це враховувати. Помилка квантування обернено пропорційна кількості бітів, які використовуються для квантування, тобто мала кількість бітів призводить до вищої помилки квантування.

Квантування і стохастичні випадання пакетів[4] можуть розглядатися щодо квадратичної стійкості NCS, а кінцеве квантування використовуватись при реалізації контролера. В такому випадку стабільність вхідного стану буде отримана шляхом застосування стратегії безперервного перемикавання контролера між масштабуванням «зменшення» та «збільшення». Схожим шляхом можна отримати параметризованні вхідні дані для стабільності стану NCS, який має схильність до випадання пакетів і невідомих збурень, але має випадкову довжину областей квантування на основі процесу випадання пакетів.

Квантування викликає два явища:

1) Насичення, яке виникає, коли сигнал перевищує діапазон квантування, що призводить до більш високої помилки квантування, і таким чином викликає нестабільність у системі із замкнутим контуром.

2) Погіршення характеристик навколо вихідної точки, що відбувається поруч із початком координат, коли сигнал не квантується точно через обмеження точності квантизаторів, і це перешкоджатиме досягненню асимптотичної стійкості замкнutoї системи.

ВТРПАТА ПАКЕТИВ

По причині використання мережі для зв'язку, сигнали системи перед передачею необхідно групувати, кожна група таких сигналів називається пакетом і її розмір залежить від мережі. Передача пакетів може бути як одиночною, так і багаторазовою. При передачі одного пакета усі дані від датчиків або контролера групуються та передаються разом. В іншому випадку, при багатопакетній передачі дані передаються кількома пакетами, що призводить до неодноразового надходження даних до контролера або виконавчого пристрою. Обмежений розмір самої мережі — причина використання паралельної передачі, а розподіл датчиків і виконавчих механізмів на великій території додатково ускладнює об'єднання даних в один мережевий пакет, що призводить до використання багаторазової передачі.

Виникнення збоїв або колізій повідомлень на вузлах призводить до випадання пакетів. Для уникнення такої проблеми, більшість протоколів використовують механізми повторної передачі. Однак, якщо повторна

передача завершується невдало протягом обмеженого часу, пакети відкидаються. Оскільки мережа зв'язку є джерелом втрат, цей тип називається випаданням пакетів, викликаним мережею. Більше того, якщо новий пакет, відправлений раніше, стане доступним на вузлі пізніше, то доречно буде його і використовувати останній, і цей тип відсіву пакетів називається відсів активного пакета. Для вирішення цієї проблеми правильним

вважається використання механізму логічного утримання нульового порядку (ZOH)[5] та відхилення повідомлень.

Один метод боротьби з випаданням пакетів полягає в тому, щоб спроектувати контролер так, щоб він витримував верхню межу випадання пакетів у системі. Інший відомий підхід – використати принцип відключення в системі з перемикачем (Рис. 3).

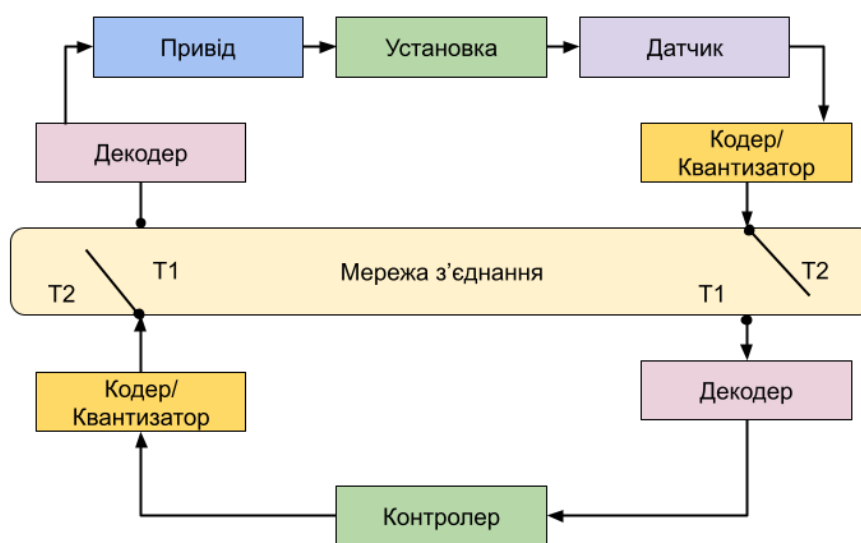


Рис.3. Конфігурація NCS з квантуванням і відкиданням пакетів даних з перемикачем

ЗМІННІ ІНТЕРВАЛИ ВИБІРКИ/ПЕРЕДАЧІ

Сигнали у мережевих системах управління необхідно дискретизувати перед їх передачею. Періоди вибірки зазвичай фіксовані у звичайних системах через її простоту в розробці та аналізі і мають назву «вибірка із синхронізацією за часом», «періодична вибірка» або «рівномірна вибірка». Проте, в сучасних структурах NCS вони відрізняються, оскільки вони знаходяться у черзі перед процесом передачі, що залежать від доступності мережі та протоколу, що використовується. Нещодавно було доведено, що вибірка у часі може мати кращу продуктивність, ніж вибірка через фіксовані інтервали[6].

Інший метод – це вибірка, яка ініційована подіями. Її також називають: "вибірка Лебега", або "вибірка на перетині рівнів", "вибірка за величиною" і т. д. У цьому випадку вибірка та передача відбуваються на основі ініціювання події, такого як зміна одного з вихідних сигналів на певне значення.

Використовується кілька підходів до моделювання інтервалів вибірки/передачі, але найбільш відомим з них

є підхід із затримкою на вході з використанням лінійної матричної нерівності. Застосовуючи цей підхід, легко визначити максимальну верхню межу двох послідовних вибірок та спроектувати відповідний контролер для NCS.

ЗМІННІ ЗАТРИМКИ ПЕРЕДАЧІ

Як показано на Рис. 4, NCS має два основних типи затримок:

- 1) затримка між датчиком і контролером, яка є часом між вибіркою сигналу від датчиків і часом його отриманням контролером;
- 2) Затримка між контролером і приводом, яка є часом між генерацією керуючого сигналу і часом його отриманням приводом.

Деякими із джерел цих затримок є обмежена смуга пропускання даних, мережевий трафік і протоколи, що використовуються в мережі[7].

Випадкові затримки між датчиком і контролером, і між контролером та виконавчим механізмом можуть розглядатися та моделюватися як ланцюг Маркова.

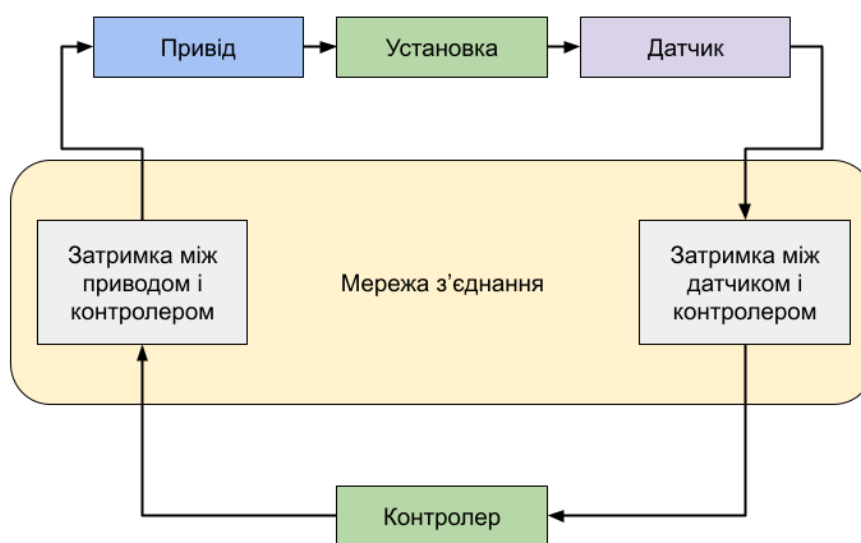


Рис. 4. Конфігурація NCS із затримкою, що спричинена мережею

Існує чотири основні моделі випадкових затримок у мережевих системах контролю:

1) Модель із постійною затримкою

NCS у цій моделі розглядається як детермінована система з постійною затримкою, і зазвичай дорівнює максимальній затримці в системі. Її використовують, коли складно охарактеризувати випадкову затримку у системі. У вузлі контролера (або виконавчого пристрою) створюється буфер приймача, і його розмір дорівнює максимальній затримці. Таким чином, мережеву систему управління можна розглядати як детерміновану систему, після чого об'єктивним буде застосування методів детермінованого управління для досягнення стійкості системи.

2) Взаємозалежна стохастична модель затримки

У тому випадку, якщо ймовірнісна залежність невідома, тоді модель із постійною затримкою та методом детермінованого управління не зможуть досягти необхідної продуктивності системи. Причина полягає у наявності в мережах багатьох стохастичних факторів, таких як: навантаження в мережі, конкуренція між вузлами та перевантаження мережі, і ці фактори створюють затримку стохастичної мережі. Затримку можна змодельовати або як взаємно незалежну, або як імовірно залежну.

3) Модель ланцюга Маркова

Цей тип враховує особливе відношення залежності між затримками, які представляють собою ланцюг Маркова. Ця модель має два типи:

а) Один ланцюг, що включає суму затримок у NCS, тобто «датчик-контролер» та «контролер-датчик».

б) Два ланцюги для моделювання затримки між датчиком і контролером та затримки між контролером та виконавчим механізмом.

4) Прихована марківська модель

У цій моделі всі стохастичні фактори (навантаження в мережі, конкуренція вузлів і навантаження мережі) утворюють приховану змінну і визначаються як стан мережі, і цей стан відповідає за розподіл затримок. Стан мережі не можна спостерігати безпосередньо, його можна оцінити шляхом спостереження за затримками в мережі, тому для опису зв'язку між станом мережі та затримкою в мережі застосовується прихована марківська модель.

ОБМЕЖЕННЯ У КОМУНІКАЦІЇ

У NCS мережа зв'язку зазвичай використовується разом із датчиками і виконавчими органами кількох вузлів, і через обмеження передачі, лише один чи кілька цих вузлів активні одночасно і мають доступом до мережі. Цю причину обмеженого зв'язку можна назвати «обмеженням доступу до середовища». Через це виникає потреба у протоколі для розподілу доступу до мережі кожного вузла. Цей протокол може бути детермінованим або випадковим. Таким чином, модель цих обмежень у NCS може бути як детермінованою, і стохастичною[8].

1) Детермінована модель комунікаційних обмежень

Основною задачею є вибір періодичної послідовності зв'язку і після цього визначення параметрів відповідного контролеру, але цей метод являє собою NP-складне завдання, де задача полягає в тому, щоб спочатку спроектувати контролер, а потім знайти відповідну послідовність зв'язку або в автономному режимі, або в режимі онлайн.

2) Стохастична модель комунікаційних обмежень

У цій моделі використовується протокол управління довільним доступом до середовища передачі (MAC). Вузол перед передачею своїх даних перевіряє відсутність іншого трафіку.

ПІДХОДИ ДО СТАБІЛІЗАЦІЇ NCS

Враховуючи перелічені недосконалості та обмеження, що можуть викликати негативний вплив на мережеві системи управління, а також суттєво погіршити продуктивність її роботи, логічно припустити, що ми маємо перелік певних підходів та рекомендацій, які дозволяють визначити напрямок та метод, за яким можна виконати роботу по покращенню ефективності роботи мережевих систем контролю, в залежності від впливу, що діє на систему. Більш детально вивчення даного питання дає можливість навести кілька методів стабілізації NCS з урахуванням одного чи кількох недоліків. Ці методи, перелічені на Рис. 5, включають: підхід системи вхідної затримки, підхід системи з перемиканням, підхід марківської системи, підхід імпульсної системи, підхід стохастичної системи і підхід прогнозуючого управління.

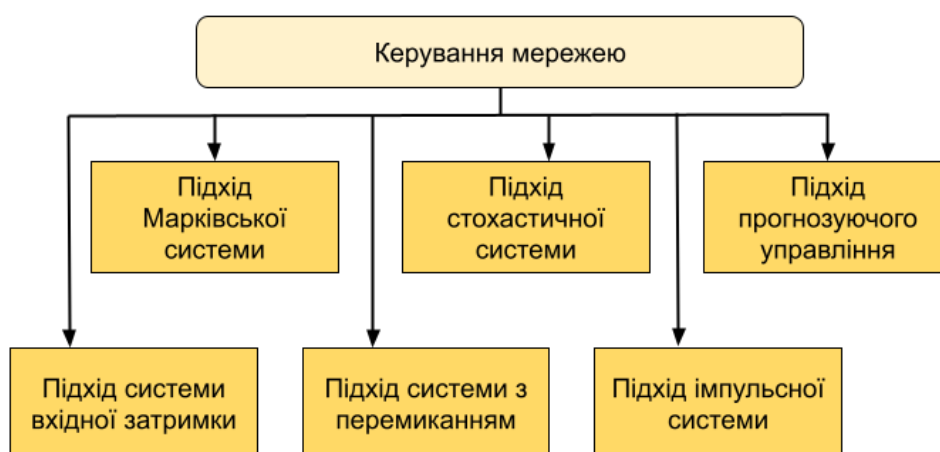


Рис.5. Підходи до стабілізації NCS

Підхід до системи вхідної затримки. При даному підході моделювання мережевої системи управління відбувається у вигляді системи система із змінною у часі затримкою, яка в свою чергу включає: затримку від датчика до контролера, затримку від контролера до виконавчого механізму та подання відключення як затримки. Також за потреби можна врахувати затримку системи під час виконання обчислень.

Підхід до системи вхідної затримки може застосовуватися з іншими протоколами, такими як протокол циклічного перебору і квадратичний протокол, і результати обох з них може бути розширеним та застосованим до дискретного NCS з обмеженнями приводу для систем з більш ніж двома вузлами.

Підхід марківської системи. У цьому підході марківська модель застосовується для представлення NCS замкнутого циклу[9]. Підхід марківської системи може бути застосований до завдання керування транспортним засобом з метою вивчення впливу втрати

пакетів на систему, що використовує бездротову локальну мережу.

Особливий клас гібридних та стохастичних систем називається марківською системою стрибка. Ця система застосовна у багатьох реальних системах, таких як виробничі системи, енергетичні, хімічні, економічні, системи зв'язку та управління.

Підхід до комутованої системи. Даний підхід являє собою систему комутування з дискретним часом і кінцевим числом підсистем. Для мережевої системи управління формується система комутації з дискретним часом і довільною комутацією, при цьому враховувалися ефекти обмеженої невизначеної затримки доступу і втрат пакетів, і в такому випадку, об'єкт являє собою безперервну лінійну стаціонарну систему.

NCS з безперервним часом моделюється як заснована на подіях модель дискретного часу, що допускає нерівномірну вибірку і затримку, що змінюється, перевищує період вибірки, тоді стабільність досягається

шляхом вирішення завдання управління для перемикання політопної системи з додаванням обмеженої невизначеності. Лінійна невизначена система з дискретним перемиканням у часі використовувалася для моделювання NCS, що включає інтервали передачі, що змінюються в часі, де змінюються в часі затримки передачі та обмеження зв'язку. При кожній окремій передачі лише одному відомому вузлу дозволено взаємодіяти з мережею передачі даних. Потім метод опуклого перенаближення у вигляді політопної системи з обмеженою нормою адитивної невизначеності використовувався для визначення критеріїв стійкості цієї NCS. Асимптотична стійкість може бути гарантована на основі квантизатора кінцевим рівнем квантування, а параметри квантизатора будуть відповідним чином налаштовані.

Підхід до комутованої системи можливо реалізувати для NCS із затримками, що викликані мережею, шляхом визначення функції перемикання. NCS із зворотним зв'язком представлена системою перемикання із затримкою та двома режимами перемикання, і кожен режим має контролер з різним коефіцієнтом посилення. Аналіз стійкості проводиться як на основі моделі системи з перемиканням часу, так і на основі методу середнього

часу перебування. Аналогічним чином досягається експоненційна стійкість.

Стохастичний системний підхід. Підхід стохастичної системи застосовується, коли затримки, спричинені мережею, та/або випадання пакетів є випадковими[10].

Імпульсивний системний підхід. У цьому підході NCS представлена гібридною дискретно-безперервною моделлю або, іншими словами, «імпульсною системою».

Підхід до прогнозуючого управління. В даному підході мережний прогнозуючий контролер (NPC) призначений для компенсації впливу тимчасових затримок та втрати пакетів у мережі. Схема системи управління, як показано на Рис. 6, містить у собі генератор управління з прогнозуванням та компенсатор. Набір майбутніх прогнозів керування генерується, інкапсулюється та передається на бік установки генератором прогнозів керування на основі сигналів, що отримані від датчика. Використовуючи останнє керуюче значення з останньої послідовності прогнозування управління, компенсатор призначений для компенсації затримок і провалів, що виникають при передачі від датчика до контролера або від контролера до каналів виконавчого механізму для досягнення бажаної продуктивності

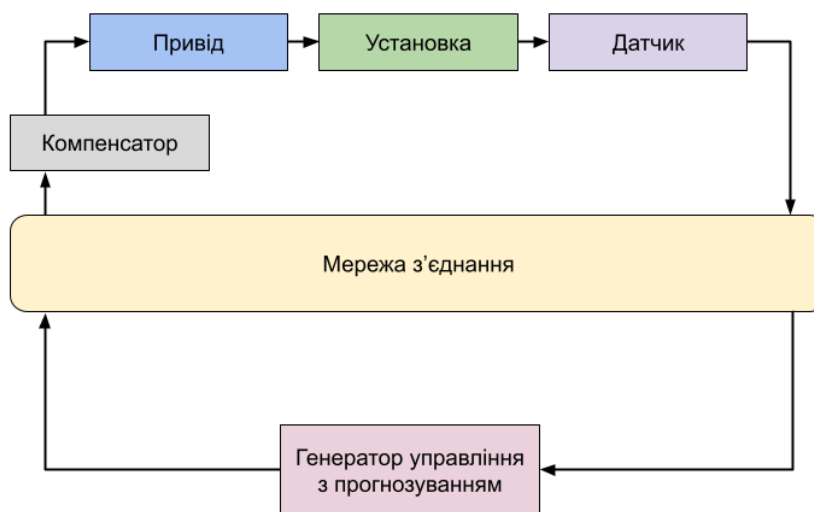


Рис. 6. Конфігурація NCS із використанням прогнозуючого контролера

ВИСНОВКИ

В результаті проведеного аналізу визначено основні поняття та відомості про мережеві системи контролю, їх характеристики і перспективи розвитку і використання даних систем у різних умовах. Можливості мережевих систем контролю дозволяють не лише застосовувати їх при побудові систем різних напрямів - військова сфера, сфера досліджень складно доступної місцевості, космічні

дослідження та багато іншого, більш того переваги NCS дозволяють легко модернізувати, масштабувати і змінювати систему.

Визначено ряд основних недоліків і обмежень мережевих систем контролю. Описано параметри, на які вони найбільше впливають. Приведено схематичне зображення структури NCS з варіантами наявності цих недоліків. Враховуючи це, приведено також і основні

підходи, які використовуються при дослідженні цих аспектів.

Систематизація і класифікація визначених недоліків, а також і підходів для їх усунення дозволяє полегшити процес вибору стратегії забезпечення стабільної роботи мережевої системи управління, розширити перелік параметрів, варіація яких, дозволить збільшити ефективність роботи системи, а також це є вагомим фундаментом для підготовки процесу моделювання подібних систем, оскільки дозволяє спростити процес створення самої моделі системи, контролерів та їх параметрів і більш детально вивчити їх вплив на роботу систему шляхом моделювання, що і планується зробити в подальших роботах для демонстрації наглядних результатів цього.

ЛІТЕРАТУРА

- [1] R. A. Gupta and M. Chow, "Networked control system: overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.
- [2] W. P. M. H. Heemels, D. Nešić, A. R. Teel, and N. van de Wouw, "Net-worked and quantized control systems with communication delays," in *Proc. 48th IEEE Conf. Decision and Control (CDC) held jointly with 28th Chinese Control Conf.*, Shanghai, China, 2009, pp. 7929-7935.
- [3] Y. Q. Xia, Y. L. Gao, L. P. Yan, and M. Y. Fu, "Recent progress in networked control systems - a survey," *Int. J. Autom. Comput.*, vol. 12, no. 4, pp. 343-367, Aug. 201. doi: 10.1007/s11633-015-0894-x
- [4] M. Y. Fu and L. H. Xie, "The sector bound approach to quantized feedback control," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1698-1711, Nov. 2005. DOI: 10.1109/TAC.2005.858689. [Онлайн] URL: <http://surl.li/vmxkbi/>. Дата звернення: 08.04.2024.
- [5] J. L. Xiong and J. Lam, "Stabilization of networked control systems with a logic ZOH," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 358-363, Feb. 2009. DOI: 10.1109/TAC.2008.2008319 [Онлайн] URL: <http://surl.li/fkovsc>. Дата звернення: 08.04.2024.
- [6] D. Zhang, P. Shi, Q. G. Wang, and L. Yu, "Analysis and synthesis of networked control systems: a survey of recent advances and challenges," *ISA Trans.*, vol. 66, pp. 376-392, Jan. 2017. [Онлайн] URL: <http://surl.li/nlxqif>. Дата звернення: 08.04.2024.
- [7] Y. L. Wang and Q. L. Han, "Modelling and controller design for discrete-time networked control systems with limited channels and data drift," *Inf. Sci.*, vol. 269, pp. 332-348, Jun. 2014. [Онлайн] URL: <http://surl.li/kbbgmc>. Дата звернення: 08.04.2024.
- [8] A. L. Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*. New York: McGraw-Hill, 2001.

- [9] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems," in *Proc. 2001 American Control Conf.*, Arlington, VA, USA, pp. 1491-1496.
- [10] D. Yue, E. G. Tian, Z. D. Wang, and J. Lam, "Stabilization of systems with probabilistic interval input delays and its applications to networked control systems," *IEEE Trans. Syst. Man Cybern. A: Syst. Humans*, vol. 39, no. 4, pp. 939-945, Jul. 2009. [Онлайн] URL: <http://surl.li/vhngge>. Дата звернення: 08.04.2024.

RESEARCH OF THE EFFICIENCY OF NETWORKED CONTROL SYSTEMS

Ivan Chystyk, Viktoriya Voropayeva

The purpose of this work is determined by the need for a detailed study and analysis of possible problems and shortcomings that arise or may arise when creating networked control systems (NCS). Each of the identified negative factors has its own influence and arises in different situations, has its own characteristics and different nature of behavior and origin. Since a detailed study of the operation of networked control systems is usually accompanied by the process of modeling such a system, for the adequacy and accuracy of the modeling process, it is necessary to create the correct apparatus that takes these indicators into account.

The analyzed information from various sources and the reviewed materials made it possible to create an idea of the main types and types of negative factors and shortcomings accompanying the operation of networked control systems. This approach allows for a faster and more focused response to the cause of possible delays in the operation of network control systems, which are caused by certain limitations. Considering that the indicators of speed and accuracy in the operation of control systems are one of the key ones, it would be fair to consider that the intention to improve the situation with the negative influence of such factors is an appropriate and necessary way. Since NCS is a complex system that contains elements of different levels, the very essence of the identified problems was also determined, i.e. not only the reasons for the appearance and occurrence of these factors were clarified, but also what they are, what they influence and to which it can lead to consequences in certain situations and under different conditions.

Taking into account the possible presence of such negative factors in networked control systems, it seems logical to try to determine approaches and methods that can be used to reduce or neutralize this or that limitation in various possible conditions. Understanding the possibility of the presence of negative factors and imperfections in the operation of networked control systems, we have the opportunity to play ahead and lay elements during the design and construction of such a system that will improve the performance of the system.

In relation to these approaches, information is also provided on the conditions in which they can be applied.

Keywords: *networked control systems, modeling, NCS, control, stability of operation.*

REFERENCES

- [1] R. A. Gupta and M. Chow, "Networked control system: overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.
- [2] W. P. M. H. Heemels, D. Nešić, A. R. Teel, and N. van de Wouw, "Net-worked and quantized control systems with communication delays, " in *Proc. 48th IEEE Conf. Decision and Control (CDC) held jointly with 28th Chinese Control Conf. , Shanghai, China, 2009*, pp. 7929-7935.
- [3] Y. Q. Xia, Y. L. Gao, L. P. Yan, and M. Y. Fu, "Recent progress in networked control systems - a survey, " *Int. J. Autom. Comput. ,* vol. 12, no. 4, pp. 343-367, Aug. 201. doi: 10.1007/s11633-015-0894-x
- [4] M. Y. Fu and L. H. Xie, "The sector bound approach to quantized feedback control, " *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1698-1711, Nov. 2005. DOI: 10.1109/TAC.2005.858689. [Online] URL: <http://surl.li/vmxkby/>. Accessed: 08.04.2024.
- [5] J. L. Xiong and J. Lam, "Stabilization of networked control systems with a logic ZOH, " *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 358-363, Feb. 2009. DOI: 10.1109/TAC.2008.2008319 [Online] URL: <http://surl.li/fkovsc>. Accessed: 08.04.2024.
- [6] D. Zhang, P. Shi, Q. G. Wang, and L. Yu, "Analysis and synthesis of networked control systems: a survey of recent advances and challenges, " *ISA Trans. ,* vol. 66, pp. 376-392, Jan. 2017. [Online] URL: <http://surl.li/nlxqif>. Accessed: 08.04.2024.
- [7] Y. L. Wang and Q. L. Han, "Modelling and controller design for discrete-time networked control systems with limited channels and data drift, " *Inf. Sci. ,* vol. 269, pp. 332-348, Jun. 2014. [Online] URL: <http://surl.li/kbbgmc>. Accessed: 08.04.2024.
- [8] A. L. Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*. New York: McGraw-Hill, 2001.
- [9] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems, " in *Proc. 2001 America Control Conf. , Arlington, VA, USA, pp. 1491-1496*.
- [10] D. Yue, E. G. Tian, Z. D. Wang, and J. Lam, "Stabilization of systems with probabilistic interval input delays and its applications to networked control systems, " *IEEE Trans. Syst. Man Cybern. A: Syst. Humans*, vol. 39, no. 4, pp. 939-945, Jul. 2009. [Online] URL: <http://surl.li/vhngge>. Accessed: 08.04.2024.

РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ТА ЇЇ РОЗГОРТАННЯ НА ХМАРНОМУ СЕРЕДОВИЩІ

Я.Л. Байтєльман¹, В.В. Поцєпаєв¹

¹ Department of Automation and Telecommunications, Donetsk National Technical University, Lutsk, Ukraine

E-mail: valerii.potsepaiev@donntu.edu.ua

Отримано 19.06.2024

Прийнято до публікації 01.08.2024

Опубліковано 01.11.2024

АНОТАЦІЯ

Метою даної роботи є створення простої та доступної для адаптації системи управління контентом, яка легко інтегрується у веб або мобільні застосунки, і може бути налаштована відповідно до специфічних потреб конкретного проекту. Однією з ключових переваг є її універсальність і гнучкість, що дозволяє розробникам швидко і з мінімальними зусиллями адаптувати програмне рішення до вимог конкретної галузі або бізнес-процесів.

Було проведено комплексну технічну розробку і створено програмне забезпечення, яке ґрунтується на використанні мови програмування PHP, реляційної бази даних MySQL і середовища для хмарного розгортання Amazon AWS. Така комбінація інструментів забезпечує високу продуктивність і надійність, що є критично важливим для сучасних застосунків. Робота містить детальні пояснення основних методів і функцій, які використовуються для обробки та збереження даних. Весь програмний код доступний у відкритому доступі.

Наукова новизна роботи полягає в тому, що матеріал викладено у доступній формі, що значно спрощує реалізацію завдань з управління контентом навіть для осіб, які не мають глибоких знань у програмуванні. Це відкриває нові можливості для міждисциплінарних досліджень і співпраці між різними галузями, зокрема в комп'ютерних науках та інших прикладних дисциплінах, таких як маркетинг, журналістика або освіта.

З точки зору практичної значимості наведені матеріали дозволяють за кілька годин розгорнути повноцінну систему управління контентом у хмарному середовищі Amazon AWS. Програмний код, із відповідними модифікаціями для реалізації різних бізнес-логік, був успішно впроваджений на декількох вебсайтах, які щоденно відвідують тисячі користувачів, що свідчить про ефективність і стабільність запропонованого рішення.

Матеріали роботи можуть бути корисними для навчання студентів технічних і економічних спеціальностей, а також школярів старших класів.

Ключові слова: управління контентом, вебсайт, PHP, MySQL, AWS

ВСТУП

Попри існування великого розмаїття рішень для управління контентом, включно із проектами відкритого коду (open source) в площині вирішення практичних завдань в різних галузях та індустріях часто виникає необхідність розробки компактного, простого для застосування і легкого в підтримці рішення. З урахуванням конкретних вимог до того або іншого проекту використання готових рішень передбачає значну адаптацію (кастомізацію), що вимагає, по-перше, детальне вивчення обраного рішення, а по-друге, внесення необхідних змін. Типовою є ситуація, коли взяте за основу рішення переробляється настільки суттєво, що фактично нічого не залишається від основи, або коли значний обсяг функціоналу взагалі не використовується, є зайвим, і таким чином суттєво ускладнюється підтримка і обслуговування, витрачаються пам'ять та обчислювальні потужності на виконання коду, який не є необхідним, а також є джерелом потенційних безпекових ризиків. Одним з прикладів таких універсальних систем управління контентом є WordPress, іншим – Joomla. Попри популярність цих систем і числені розроблені для них графічні теми та функціональні плагіни, добре відомими в індустрії є також скарги на роботу з ними. Окрім того, ці системи є “монолітними”, що не дозволяє їх розгортання в безсерверному або мікросервісному середовищі, хоча саме безсерверні та мікросервісні рішення на сьогодні вважаються ефективними щодо вартості та обслуговування.

Автор мав власний досвід адаптації та використання WordPress і Joomla в сферах електронної комерції, бібліотечної справи та новин і в кожному проекті на тому чи іншому етапі стикався із економічно обґрунтованим рішенням відмовитись від цих систем на користь більш ефективної розробки.

АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ ТА ПОСТАНОВКА ПРОБЛЕМИ

Мова програмування PHP [1] була вперше представлена в 1994 році у формі простої бібліотеки CGI, в перекладі українською – загального серверного інтерфейсу для виконання на сервері певного програмного коду, який викликається зверненням з боку клієнту через HTTP протокол [2]. Фактично технологія CGI забезпечила розвиток динамічних вебсайтів, на відміну від статичних документів, які просто завантажуються по запиту від клієнта. PHP була не першим втіленням CGI, проте швидко набуло популярності і не втрачає її завдяки своїй гнучкості і перед усім можливості як створювати

швидкі прості рішення на базі об'єднання в одному файлі коду HTML і JavaScript (які виконуються на клієнтській стороні) з кодом PHP (що виконується на стороні серверу), так і повноцінно підтримувати MVC-парадигму з чітким розподілом на модель, вигляд, контролер [3] і об'єктно-орієнтоване програмування [4]. Перше забезпечує розв'язання відносно простих задач на кшталт створення прототипів, перевірку гіпотез, швидке моделювання. Друге послугує потребам всього широкого спектру розробки програмних рішень різного рівня складності. Сервер бази даних MySQL є складовою частиною так званого набору інструментів LAMP, що складається з операційної системи Linux, веб-серверу Apache, серверу баз даних MySQL, мови програмування PHP, цей набір зберігає актуальність вже протягом багатьох років через низьку вартість і доступність [5]. В рамках даної роботи розглядається типова архітектура серверної сторони на основі LAMP. Покрокові інструкції з налаштування хмарного середовища Amazon AWS доступні на веб сайті з офіційною технічною документацією [6]. Варто зауважити, що аналогічне середовище можна розгорнути на потужностях будь-якого іншого хостінг-провайдера або навіть на власній апаратній базі, і запропоновані PHP-скрипти будуть виконуватися, проте пропонується використання деяких комерційних сервісів Amazon AWS, а саме CloudFront [7] для кешування і підтримки безпечного протоколу передачі гіпертексту, Route 53 [8] для налаштування DNS, S3 [9] для збереження статичних файлів, RDS [10] для розміщення бази даних та її автоматичного резервного копіювання, аналоги до яких можуть бути знайдені у інших провайдерів але їхнє використання вимагає внесення змін в технічний дизайн запропонованого рішення.

Формулювання проблеми: перед фахівцями різних сфер бізнесу, науки, державного чи місцевого врядування часто постає проблема організації управління контентом, при чому необхідно забезпечити віддалений доступ до такої системи через мережі загального користування (Інтернет), надійне зберігання інформації, включно із регулярним резервним копіюванням і гарантованим відновленням, а також забезпечити низку інших вимог з імплементації певної бізнес-логіки, що разом знижує привабливість використання готових рішень і натомість створює потребу в мінімалістичному наборі скриптів, які можна легко адаптувати під зазначені вище вимоги.

Мета: розробка програмного коду і моделей даних інструментами LAMP для розгортання на хмарному середовищі і з використанням спеціалізованих сервісів Amazon AWS.

Задачі:

1. Аналіз специфікацій сервісів Amazon AWS.

2. Розробка технічного дизайну.
3. Розробка і тестування програмного коду і моделей даних.
4. Огляд результатів роботи, підготовка пояснювальних матеріалів.

МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ

В роботі використано методи структурного і порівняльного аналізу, з інформаційним і аналітичним підходом розглянуто наукову та методичну літературу, а також онлайн ресурси, застосовано експериментальні методи для перевірки програмного коду.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Під терміном “контент” мається на увазі будь-які текстові або графічні дані, наприклад, матеріали для вебсайту служби новин, або історія пацієнтів для медичної інформаційної системи, або будь-яка інша бібліотека документів. В даній роботі розглядається система з вільним доступом до даних з боку кінцевих користувачів, що відповідає прикладу сайту новин.

З огляду на типи користувачів системи та їхні функції виокремлюються 3 типи:

1. Адміністратори, які мають найбільші повноваження в системі, можуть створювати та редагувати інших адміністраторів та редакторів контенту, надавати їм доступ до системи або блокувати його, також можуть редагувати контент.
2. Редактори контенту, мають права створювати та змінювати контент.
3. Кінцеві користувачі, мають права перегляду контенту.

Система управління контентом включає наступні компоненти:

1. Адміністративна панель, через яку здійснюється створення і редагування контенту адміністраторами системи; складається з скриптами з кодом HTML і JavaScript для виконання на клієнтській стороні та скриптами з кодом PHP для виконання на серверній стороні.
2. База даних, в якій зберігається текстова складова контенту і метадані (дата створення, інформація про автора, дозвіл на виведення у відповідь на запит з боку клієнтського інтерфейсу, належність до групи або розділу, тощо).
3. Файлове сховище, для зберігання файлів зображень (з можливим розширенням до будь-яких файлів, документів, відео і таке інше). Файлове сховище реалізується окремо від серверу, на якому зберігаються

скрипти серверного виконання з метою, за необхідності, перенести скрипти серверного виконання до безсерверного середовища.

4. Скрипти серверного виконання, які обслуговують запити від клієнтського веб або мобільного застосунку, повертаючи контент для відображення на боці користувача.
5. Система кешування та екранування доступу до файлів та скриптів з метою покращення їхнього захисту.
6. Веб або мобільний застосунок користувача (в даній роботі не наводиться програмний код користувацького застосунку, але визначається формат команд для звернення до відповідних скриптів серверного виконання).

Компонента діаграма наведена на Рис. 1.

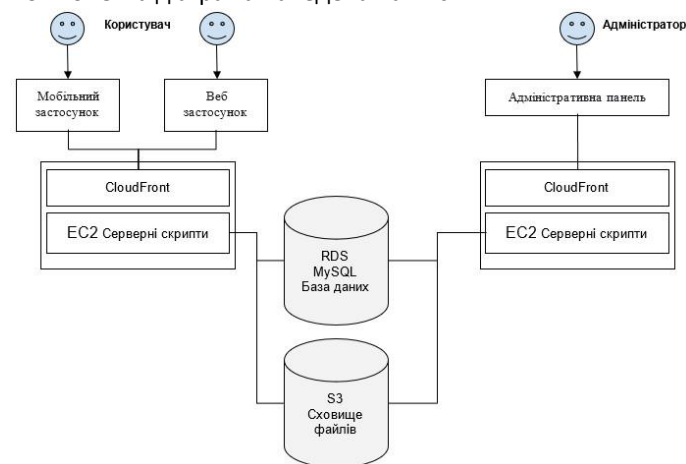


Рис. 1. Компонента діаграма

ОГЛЯД СЕРВІСІВ AWS

На сервісі EC2 (Рис. 2) створюється віртуальний сервер з потрібним обсягом оперативної пам'яті, процесорної потужності, розміром жорсткого диску. Дозволяється вибір операційної системи з списку доступних. Здійснюються налаштування безпеки та моніторингу. З практичного досвіду, для сайту з відвідуванням близько 1000 користувачів на добу, рекомендується обирати тип серверу “t2-micro”. В рамках умов даної роботи обирається операційна система Amazon Linux 2023. Базові конфігурації здійснюються через адміністративну веб-панель AWS EC2. Для встановлення веб серверу Apache та завантаженню файлів доступ до EC2 здійснюється через SSH. Покрокові інструкції наведено в офіційній документації [6]. В адміністративній веб-панелі AWS EC2 рекомендується налаштувати:

- моніторинг навантаження на сервер і відповідно надсилання повідомлення електронною поштою адміністратору в разі перевищення навантаження на процесор більше 50%;
- автоматичне резервне копіювання вмісту жорсткого диску серверу один раз на добу і зберігання копій щонайменше протягом 14 днів, що дасть можливість легкого і швидкого повернення по попередньої версії в разі необхідності;

- відкрити лише порти 443 (HTTPS), 80 (HTTP), 22 (SSH);
- асоціацію даного серверу з публічною IP адресою, яка буде використовуватися для підключення до нього через SSH. Як буде пояснено нижче, ця адреса не використовуватиметься в налаштуваннях DNS, тож вона потрібна лише для зручності адміністрування.

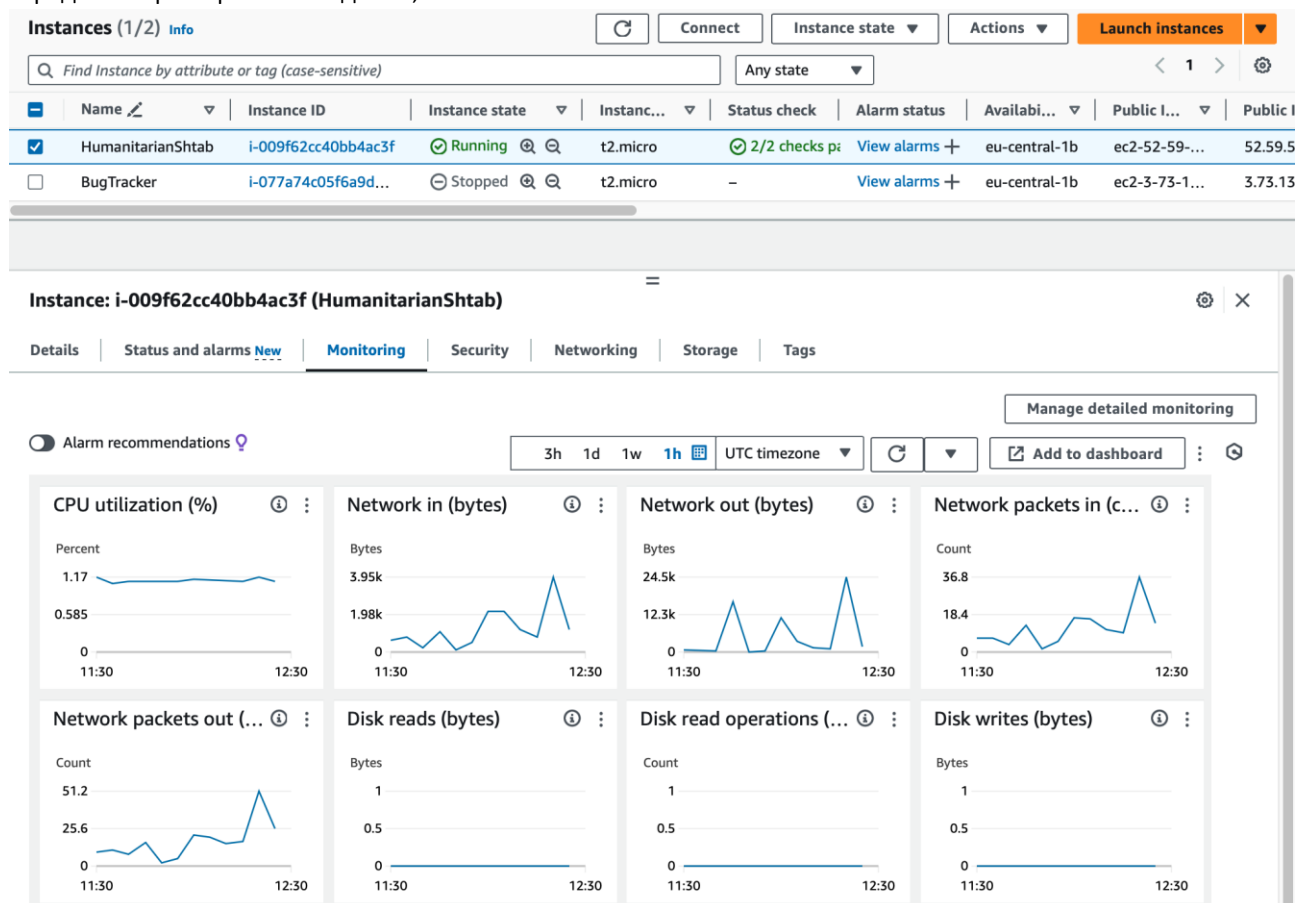


Рис.2. Налаштування EC2

За бажанням замість EC2 можна розміщувати скрипти серверного виконання на безсерверних сервісах, наприклад Lambda [11]. Альтернативно, для обслуговування великої кількості одночасних запитів, можна встановити режим автоматичного збільшення ресурсів серверу EC2, або розподіл запитів через балансир навантаження на декілька EC2 серверів, хоча це виходить за межі даної роботи, автор вважає за потрібне вказати на наявність таких можливостей для втілення різних підходів до динамічних змін потужності в залежності від навантаження та вимог конкретного проекту. Так само важливо зазначити, що AWS підтримує різні географічні регіони, тобто для обслуговування

клієнтів в Північній Америці варто обирати американські географічні регіони, для європейських клієнтів – відповідно європейські, для Азії – свої регіони. Існує можливість синхронізації серверів, розташованих в різних зонах.

CloudFront – це сервіс AWS, що виконує декілька функцій, важливих для захисту і оптимізації доступу до даних. По-перше, він приховує публічну IP адресу EC2 серверу, по-друге здійснює кешування даних, що дозволяє зменшити навантаження на EC2, наприклад, відповідь на запит клієнта на каталог розділів веб сайту можна кешувати, так як така інформація не змінюється часто, тож тривалість життя кешу можна встановити на

рівні кількох годин. Крім того, через CloudFront безкоштовно замовляється SSL сертифікат для безпечного HTTPS з'єднання з сервером. Таким чином всі ресурси в запропонованій системі будуть доступні тільки через

HTTPS, доступ через HTTP буде заблокований, а запити HTTP будуть автоматично переадресовані на HTTPS (Рис. 3). Деталі налаштування CloudFront наведено в

The screenshot displays the AWS CloudFront console for distribution E17UFYFE9RV2NV. It is divided into two main sections: 'Details' and 'Settings'. The 'Details' section includes the distribution domain name (d3s8cq5yvwq4yp.cloudfront.net), the ARN (arn:aws:cloudfront::963776958110:distribution/E17UFYFE9RV2NV), and the last modified date (April 2, 2022 at 5:56:50 PM UTC). The 'Settings' section includes the description (humanitaryishtab.org), price class (Use all edge locations (best performance)), supported HTTP versions (HTTP/2, HTTP/1.1, HTTP/1.0), alternate domain names (www.humanitaryishtab.org, *.humanitaryishtab.org), custom SSL certificate (humanitaryishtab.org), security policy (TLSv1.2_2021), and logging settings (Standard logging: Off, Cookie logging: Off, Default root object: -).

Рис.3. Налаштування CloudFront

документації до цього сервісу[7]. Важливо зауважити, що замовлення SSL потребує налаштування адресації в сервісі домених імен Route 53 (наведено нижче). Крім того, використання CloudFront певною мірою відіграє роль захисту від можливих DDoS атак.

Сервіс RDS дозволяє створювати бази даних без необхідності витрат зусиль на підтримку і обслуговування серверу баз даних (Рис. 4), при чому в залежності від необхідності існує можливість автоматичного збільшення чи зменшення потужності обраного RDS середовища в залежності від навантаження. Для даної роботи обрано найменше доступне середовище RDS з підтримкою MySQL, а саме db.4tg.micro, із загальним обсягом 50GB і автоматичним розширенням до 100GB за умови виникненні необхідності, налаштовано автоматичне щоденне резервне копіювання із зберіганням останніх 7 копій. Докладно про налаштування RDS можна прочитати в технічній документації сервісу [10].

Важливо пам'ятати, що винесення бази даних в окремий сервіс на RDS зменшує навантаження на сервер EC2, проте користування кожним з сервісів AWS не є безкоштовним. Утримання найменшого серверу RDS

коштує приблизно \$19 щомісяця, а найменшого EC2 – приблизно \$16 за місяць, в залежності від трафіку ці цифри можуть зростати. Amazon пропонує програму підтримки стартапів і кожному новому клієнтові надає певний безкоштовний ліміт ресурсів на перші 12 місяців, що є зручним для новачків і дає можливість практично ознайомитися з різними сервісами AWS, але з таким підходом новачки воліють до нехтування вартістю сервісів і через рік стикаються з неприємним сюрпризом, коли отримують перший рахунок. З іншого боку, винесення бази даних в RDS зменшує ризики злому та витоку даних, так як на сервері EC2 не виконується програмне забезпечення серверу баз даних і відповідно його потенційні помилки і безпекові недоліки є повністю виключеними. Те саме вірно і для веб серверу та операційної системи EC2: якщо скрипти серверного виконання розмістити в безсерверному середовищі, наприклад Lambda, тоді буде повністю відсутнім програмне забезпечення Apache і Linux разом з їхніми можливими недоліками і ризиками, проте вартість Lambda може перевищувати вартість серверу EC2. Загальне правило таке: чим більше навантаження, чим більше

користувачів, тим вигіднішими стають безсерверні рішення.

Сервіс зберігання статичних файлів S3 є безсерверним середовищем, із ключовим елементом “bucket”

(українською – “відро”). Його слід розуміти як безрозмірну директорію, де можна зберігати необмежену кількість файлів будь-якого типу і розміру, в ній також можна створювати необмежену кількість

Instance		
Configuration	Instance class	Storage
DB instance ID database-1	Instance class db.t4g.micro	Encryption Enabled
Engine version 8.0.33	vCPU 2	AWS KMS key aws/rds
DB name -	RAM 1 GB	Storage type General Purpose SSD (gp2)
License model General Public License	Availability	Storage 50 GiB
Option groups default:mysql-8-0 In sync	Master username root	Provisioned IOPS -
Amazon Resource Name (ARN) arn:aws:rds:eu-west-1:963776958110:db:database-1	Master password *****	Storage throughput -
Resource ID	IAM DB authentication Not enabled	Storage autoscaling Enabled

Рис. 4. Налаштування RDS

файлів будь-якого типу і розміру, в ній також можна створювати необмежену кількість вкладених директорій, так як це можливо у файлової системі на комп'ютері. Імена файлів в директорії мають бути унікальними. Доступ до файлів і директорій можна налаштувати з будь-яким ступенем відкритості, від повної доступності для будь-кого, навіть через веб, і таким чином bucket можна перетворити на статичний веб сайт, до приватності кожного окремого файлу чи директорії, так щоб доступ до них був лише у власника, і відповідно, bucket може бути приватним архівом. Вартість збереження файлів на S3 є найменшою серед усіх сервісів AWS, меншою за збереження на жорстких дисках EC2 і значно меншою за збереження в базі даних RDS. Саме тому є сенс виносити зберігання статичних файлів в S3. Крім того S3 також має механізми резервного копіювання, в тому числі через збереження файлів в “дзеркальний” bucket. Файли і директорії в S3 зуться об'єктами (Рис. 5). Як і для інших

сервісів Amazon пропонує повну технічну документацію для S3 [9].

Важливо зауважити наступне:

- файли на S3 не можуть виконуватись, на відміну від файлів на жорсткому диску комп'ютера чи сервера, розташований на S3 скрипт з програмним кодом або скомпільований бінарний файл не може бути викликаним для виконання, так як в середовищі S3 немає оперативної пам'яті і немає процесору;
- існують бібліотеки (SDK) для різних мов, в тому числі для PHP, і в цій роботі така бібліотека використовується, для програмного завантаження файлів на S3, їхнього видалення, копіювання, тощо. За принципом дії це схоже до функцій роботи з файлами, тільки у випадку з S3 файли знаходяться на віддаленому сховищі.

Сервіс конфігурації доменних імен Route 53 використовується для налаштувань адресації доступу до фронт-енд скриптів клієнтського застосунку та бек-енду,

причому у відповідності до Рис. 1 і Табл. 1, маємо підключити:

- адресу доменного імені – до дистрибутиву CloudFront, що вказує на розташування HTML, CSS та JavaScript

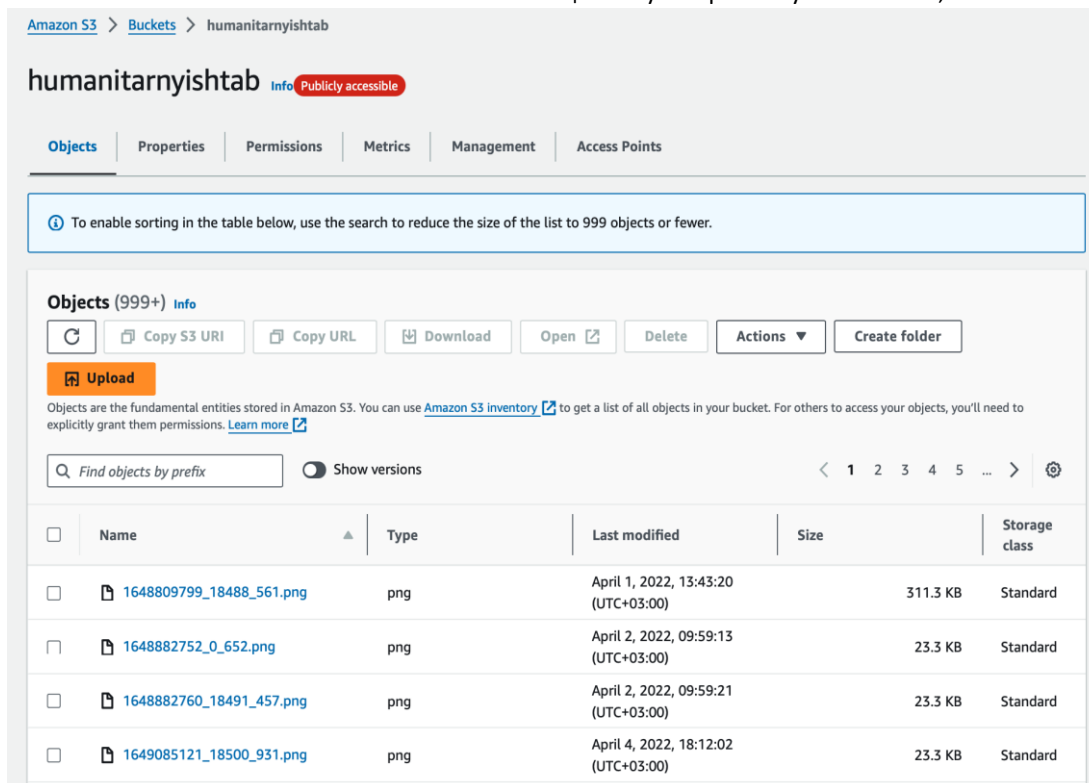


Рис.5. Файли в S3

файлів фронт-енд веб застосунку на S3;

- адресу піддомену скриптів API для обслуговування запитів від фронт-енду – до дистрибутиву CloudFront, що вказує на розташування цих скриптів на EC2;
- адресу піддомену адміністративної панелі – до дистрибутиву CloudFront, що вказує на розташування файлів PHP на EC2;

- адресу піддомену статичного контенту – до дистрибутиву CloudFront, що вказує на розташування статичних файлів, таких як графічні зображення, які завантажуються через адміністративну панель систему управління контентом, на S3.

Табл. 1. Відповідність адрес, дистрибутивів та ресурсів

Адреса	Дистрибутив CloudFront	Сервіс AWS	Призначення
myproj.com.ua	XXX1.cloudfront.net *	S3	Файли HTML, JS, CSS та інші клієнтського веб застосунку
services.myproj.com.ua	XXX2.cloudfront.net *	EC2	Файли бек-енд скриптів для клієнтського застосунку
admin.myproj.com.ua	XXX3.cloudfront.net *	EC2	Файли PHP адміністративної панелі
resources.myproj.com.ua	XXX4.cloudfront.net *	S3	Статичний контент (зображення, тощо)

* XXX1, XXX2 і так далі представляють собою значення, згенеровані сервісом CloudFront.

Таким чином приховуються реальні IP-адреси сервісів (утіліта `ping` буде повертати IP-адреси серверів CloudFront, а не ресурсів, на яких реально розташовані ті чи інші файли), також доступ до кожного сервісу

опиняється під SSL. Route 53 можна використовувати для налаштувань адресації поштових сервісів, проте це виходить в межі даної роботи (Рис. 6).

The screenshot shows the AWS Route 53 console interface for the hosted zone 'humanitaryishtab.org'. At the top, there are navigation links for 'Route 53', 'Hosted zones', and the zone name. Below this, there are buttons for 'Delete zone', 'Test record', and 'Configure query logging'. A section titled 'Hosted zone details' includes an 'Edit hosted zone' button. Below that, there are tabs for 'Records (10)', 'DNSSEC signing', and 'Hosted zone tags (0)'. The 'Records (10)' tab is active, showing a search bar and buttons for 'Delete record', 'Import zone file', and 'Create record'. A table lists the records:

Record name	Type	Routin...	Differe...	Alias	Value/Route traffic to
humanitaryishtab.org	A	Simple	-	Yes	d3s8cq5yvq4yp.cloudfront.net.
humanitaryishtab.org	MX	Simple	-	No	10 inbound-smtp.eu-west-1.amazonaws.com
humanitaryishtab.org	NS	Simple	-	No	ns-813.awsdns-37.net. ns-1776.awsdns-30.co.uk. ns-159.awsdns-19.com. ns-1359.awsdns-41.org.

Рис. 6. Записи в Route 53

ДІАГРАМИ ПОСЛІДОВНОСТІ ВИКОНАННЯ ЗВЕРНЕНЬ

Діаграма послідовності виконання звернень при відображенні контенту за запитом від клієнтського веб застосунку наведена на Рис. 7. Після завантаження коду HTML, JavaScript, CSS відбувається його інтерпретація та виконання в браузері і відповідно до його функцій може виконуватися завантаження додаткових статичних елементів, наприклад, зображень, а також здійснюватися запити на виконання серверних скриптів, наприклад, коду PHP, що знаходиться на сервері EC2 і виконується на ньому, звертається до бази даних для отримання або збереження інформації. Запити до бази даних здійснюються мовою MySQL і виконуються на сервері RDS.

Діаграма послідовності звернень при відображенні контенту за запитом від клієнтського мобільного застосунку наведена на Рис. 8. Мобільний застосунок звертається до API бек-енду та завантажує статичні файли аналогічно до дій веб застосунку. Єдина відмінність полягає в тому, що мобільний застосунок не потребує завантаження власного коду з серверу кожного разу, коли користувач запускає застосунок на виконання.

Діаграма послідовності звернень при збереженні контенту за запитом від адміністративної панелі наведена на Рис. 9.

СТРУКТУРА БАЗИ ДАНИХ

Система управління контентом пропонує організацію даних у формі статей, статті можуть належати до однієї чи більше категорій, до статті можуть бути додані графічні файли. Доступ до редукування здійснюється адміністраторами. Зазначена вище інформація зберігаються в базі даних в наступних таблицях.

Таблиця `admin_users` (Таб. 2) містить дані про користувачів адміністративної панелі. Відповідно до безпекових рекомендацій, паролі зберігаються у хеш-форматі, який не дозволяє зворотної трансформації для отримання значення паролю в його первинній текстовій формі.

Таблиця `categories` (Таб. 3) містить дані про категорії статей. Кожна стаття може належати до однієї категорії. Кожна категорія має асоційовану з нею піктограму з бібліотеки Font Awesome.

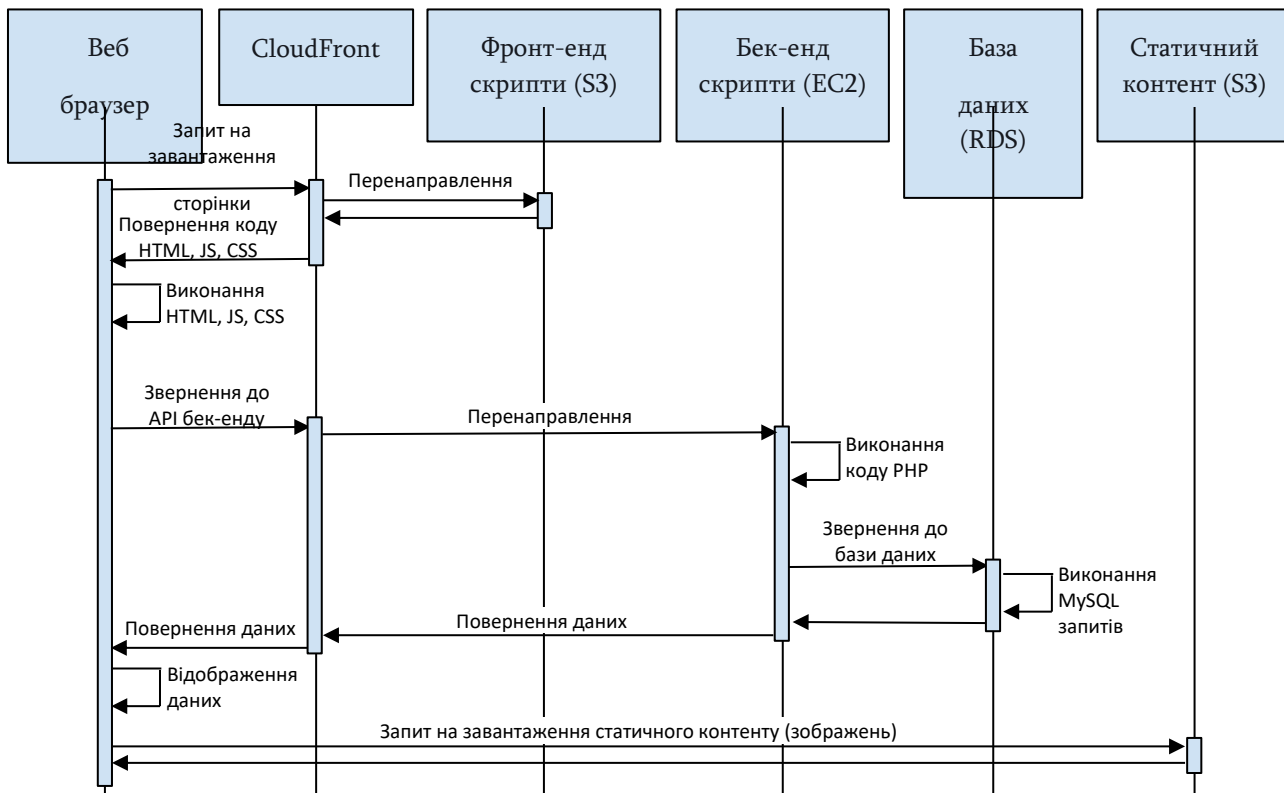


Рис. 7. Запит контенту від вебзастосунку

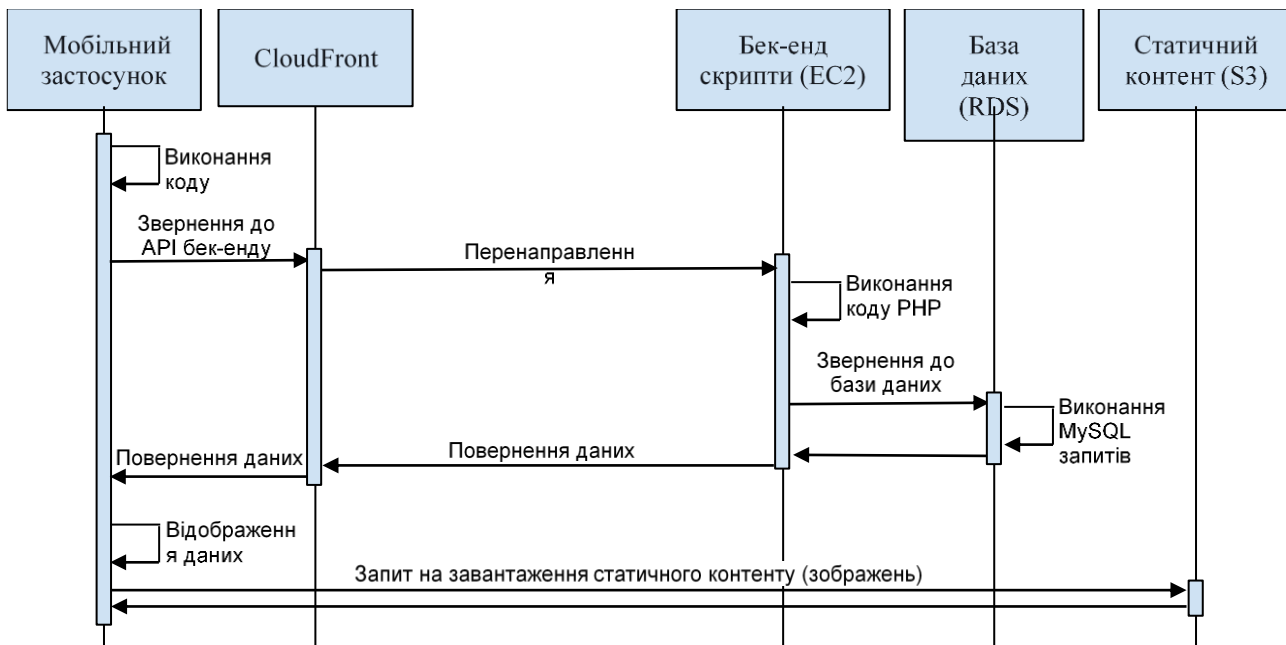


Рис. 8. Запит контенту від мобільного застосунку

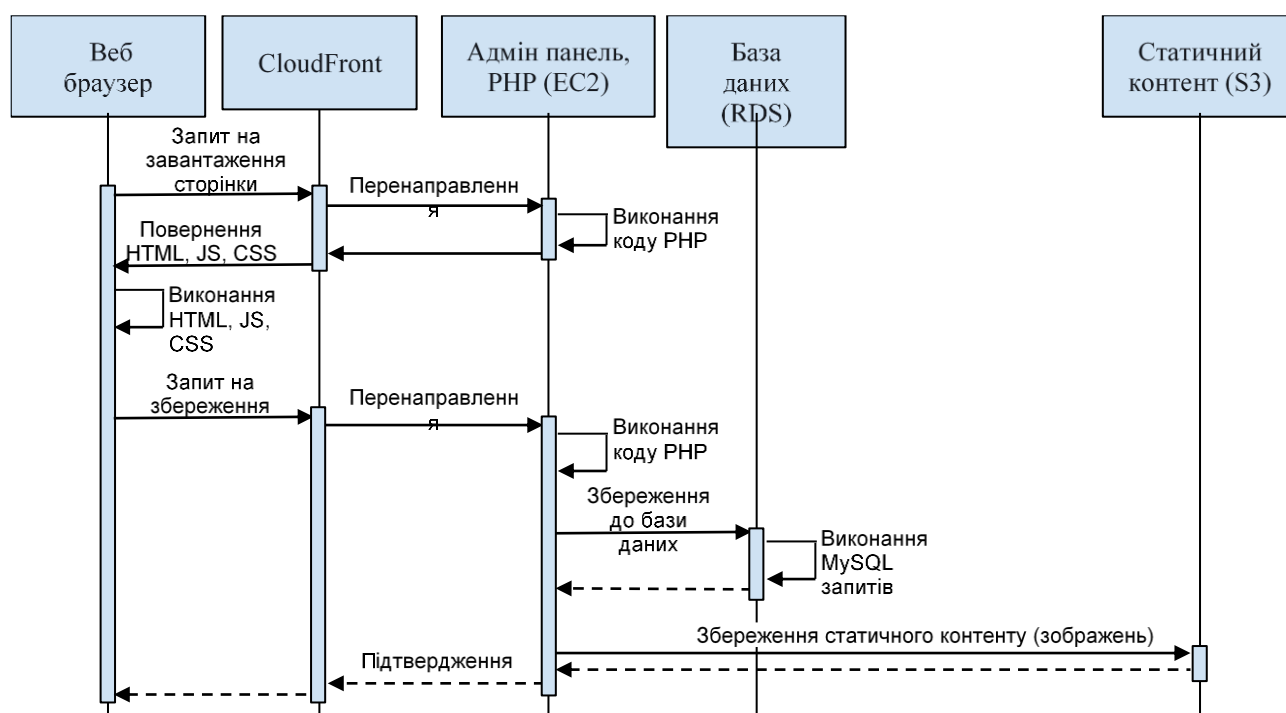


Рис. 9. Збереження контенту

Табл. 2. Структура таблиці admin_users

Поле	Тип	Примітки
id	int (11)	Унікальний ідентифікатор, первинний ключ, автоінкремент
login	varchar (50)	Ім'я користувача
password	varchar (50)	Пароль

Табл. 3. Структура таблиці categories

Поле	Тип	Примітки
id	int (11)	Унікальний ідентифікатор, первинний ключ, автоінкремент.
title	varchar (200)	Назва категорії
icon	varchar (100)	Піктограма
color	varchar (20)	Колір піктограми
bg_color	varchar (20)	Колір фону
t_color	varchar (20)	Колір тексту

відправляє назад лише дані (і далі виконання PHP припиняється, весь HTML код не відправляється знову); JavaScript, отримавши відповідь, виводить результати в певні елементи HTML. Таким чином, все що потрібно, наприклад, для функціоналу збереження та редагування публікації, знаходиться в одному PHP файлі, це є зручним, коли над проектом працює одна особа, проте, при масштабуванні на певному етапі настає потреба забезпечити розподіл обов'язків в команді, і тоді має сенс розділити HTML і JavaScript та PHP частини, перше стає відповідальністю інженера-розробника клієнтської частини, друге – програміста серверної частини. Також, можливо замінити HTML і JavaScript на, наприклад, ReactJS.

inc/db.php містить функції підключення до бази даних, а також ім'я користувача, пароль, адресу серверу бази даних. Крім того, тут містяться деякі спільні функції, які використовуються в інших файлах, наприклад функція створення паролю. Даний файл не викликається безпосередньо, а включається в інші файли.

index.php – скрипт, з якого починається взаємодія з адміністративною панеллю, тут відбувається авторизація

користувачів.

Панель Керування

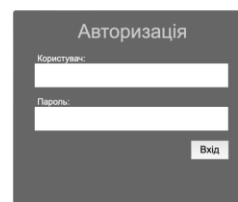


Рис. 11. Index.php

inc/session.php – скрипт для перевірки, чи є користувач авторизованим. Включається до скриптів всіх “внутрішніх” сторінок, доступних лише для авторизованих користувачів (Рис. 11).

inc/menu.php – скрипт з путнками меню, вони однакові для “внутрішніх” сторінок.

articles.php виводить список всього контенту (публікацій), з можливістю посторінкового перегляду та пошуку (Рис. 12).

Гуманітарний штаб: Публікації Користувачі Категорії Різні Вихід									
Публікації									
Пошук <input type="text"/> Категорія <input type="text"/> Область <input type="text"/> <input type="button" value="Шукати"/> <input type="button" value="Нова публікація"/>									
ID	Категорія	Область	Місто	Назва	Кількість	Автор	Дата	Actions	
19901				Звіт допомоги за тиждень	0	Anastasia	26-02-2024	🔗	
19900		Вінницька	Літин	#ДітиМаютьПравоНаДитинство	0	Anastasia	22-02-2024	🔗	
19899	Військова амуніція			18 тепловізорів, 12 прожекторів, 34 зарядні станції Bandera Po...	0	Anastasia	22-02-2024	🔗	
19898	Військова амуніція			П'ять безпілотників Mavic 3T для бійців бригади "Червона ...	0	Anastasia	19-02-2024	🔗	
19897	Продукти харчування			Продукти харчування, вода, медикаменти та окопні свічк...	0	Anastasia	19-02-2024	🔗	
19896	Військова			Mavic 3T для бійців 14-ї штурмової бригади	0	Anastasia	16-02-2024	🔗	

Рис. 12. articles.php

article_edit.php – створення та редагування статті (Рис. 13, 14).

categories.php – перегляд категорій публікацій (Рис. 15).

Публікація

Категорія: Якщо категорію не обрати, це буде просто новина (в стрічці новин) і не буде враховуватися в статистиці.

Додаткові категорії: Якщо відправлення містить предмети із різних категорій.

Область:

Місто:

Дата:

Назва:

Кількість:

Одиниці виміру:

Вступ:

Текст:

B I U Normal

Потреба у "залізних пташках" на полі бою залишається стабільно високою. Бо кожен безпілотною — це і збережені життя наших захисників, і втрати ворога у живій силі та техніці.
Поки наші хлопці роблять там усе можливе задля Перемоги, ми маємо робити те саме тут. Тому об'єднуємо зусилля та працюємо згуртовано далі!

Рис. 13. article_edit.php (редагування тексту)

Потреба у "залізних пташках" на полі бою залишається стабільно високою. Бо кожен безпілотною — це і збережені життя наших захисників, і втрати ворога у живій силі та техніці.
Поки наші хлопці роблять там усе можливе задля Перемоги, ми маємо робити те саме тут. Тому об'єднуємо зусилля та працюємо згуртовано далі!

Розташування зображень:

Зображення:

no file selected




Рис. 14. article_edit.php (редагування зображення)

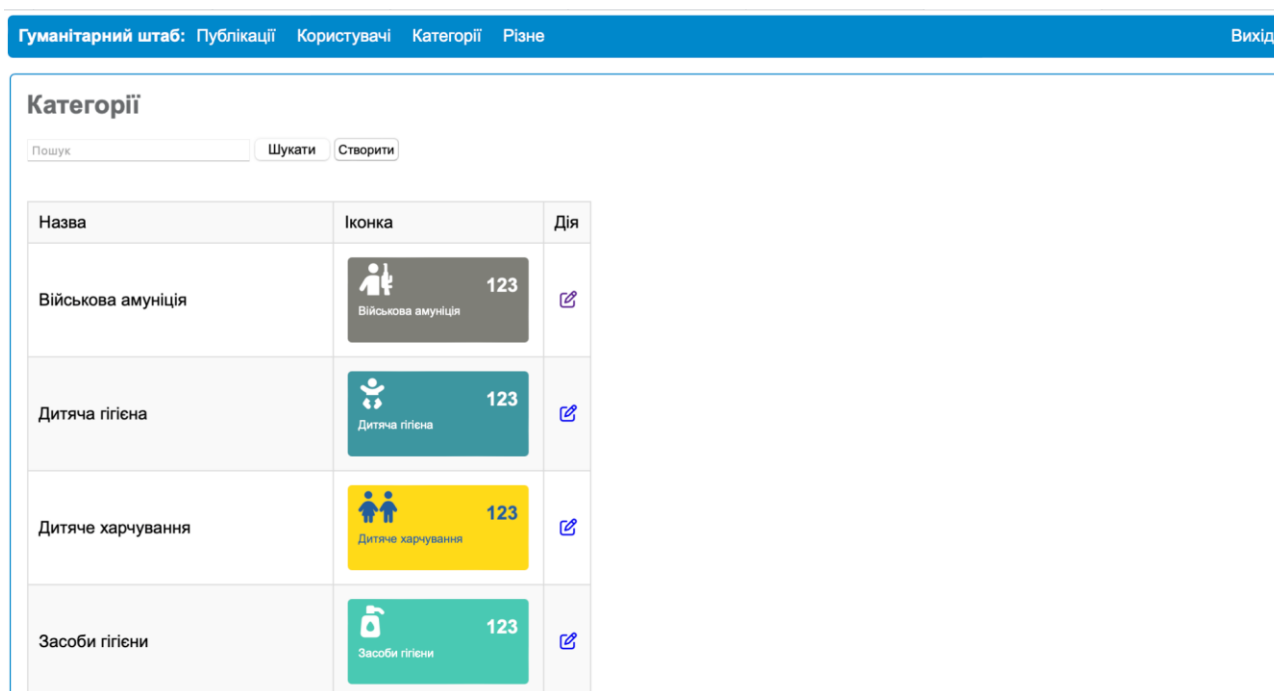


Рис. 15. categories.php

category_edit.php – створення та редагування категорії (Рис. 16).

users.php – перегляд списку користувачів (адміністраторів, Рис. 17). В системі є 2 рівня користувачів – адміністратори з повним доступом та користувачі, які мають доступ лише до редагування контенту.

Назва:

Іконка:

Колір:

Колір фону:

Колір лічильника:

Рис. 16. category_edit.php

ID	Логін	Ел. пошта	Тип	Статус	Дія
15	Anastasia	dzi...@gmail.com	Користувач	Активний	
14	yelena	yel...@mail.com	Адміністратор	Активний	

Рис. 17. users.php

user_edit.php – створення та редагування користувачів. Важливо зауважити, паролі не зберігаються у відкритому вигляді, а шифруються незворотним чином [14] (Рис. 18). Якщо користувач втрачає пароль, відновити його не є можливим, тому буде створений новий пароль. Також слід додати, що авторизація користувачів за одним фактором (паролем) не вважається надійною, тому варто

додати другий фактор, наприклад, одноразовий пароль, що надсилатиметься користувачу по електронній пошті або SMS.

logout.php – очищає дані в сесії та виконує перенаправлення на index.php.

Рис. 18. user_edit.php

ПРИКЛАДИ ВЕБ-САЙТІВ

В Табл. 5 перелічені проекти, де було застосовано дану систему управління контентом.

Табл. 5. Приклади веб сайтів

URL	Розширення базового функціоналу
humanitarnyishtab.org	Додаткові поля в публікаціях, належність публікації до кількох категорій.
vmas.info	2-факторна авторизація, підтримка 3 мов
catholic.co.il	2-факторна авторизація, підтримка 5 мов, розширений каталог категорій
daily-gospel.net	2-факторна авторизація, змінена структура контенту

ВИСНОВКИ

В ході проведення дослідження отримані наступні результати:

1. Проведено аналіз специфікацій сервісів Amazon AWS та PHP.
2. Розроблено архітектуру системи управління контентом.
3. Розроблено і протестовано структуру бази даних та програмний код.
4. Наведено пояснення щодо реалізації програмного коду у вигляді коментарів в кодї та пояснювальних матеріалів в тексті даної роботи.

ДОСТУПНІСТЬ ДАНИХ

Програмний код викладено для вільного доступу за посиланням: <https://btlr.com/code/cms/>.

ЛІТЕРАТУРА

- [1] PHP documentation. [Онлайн]. URL: <https://www.php.net>. Дата звернення: 17.10.2024.
- [2] Understanding HTTP using Browsers. [Онлайн]. URL: <http://surl.li/vwkvym>. Дата звернення: 17.10.2024.
- [3] MVC. [Онлайн]. URL: <http://surl.li/ziawqy>. Дата звернення: 17.10.2024.
- [4] Introduction of Object Oriented Programming. [Онлайн]. URL: <http://surl.li/slyzju>. Дата звернення: 17.10.2024.
- [5] What is a LAMP stack? [Онлайн]. URL: <http://surl.li/gmesgm>. Дата звернення: 17.10.2024.
- [6] Tutorial: Install a LAMP server on AL2023. [Онлайн]. URL: <http://surl.li/pkutti>. Дата звернення: 17.10.2024.
- [7] Amazon CloudFront. [Онлайн]. URL: <http://surl.li/oqwnkq>. Дата звернення: 17.10.2024.
- [8] Amazon Route 53. [Онлайн]. URL: <http://surl.li/tusvxf>. Дата звернення: 17.10.2024.
- [9] Amazon S3. [Онлайн]. URL: <http://surl.li/qbziyg>. Дата звернення: 17.10.2024.
- [10] Amazon Relational Database Service. [Онлайн]. URL: <http://surl.li/awliinq>. Дата звернення: 17.10.2024.
- [11] Amazon Lambda. [Онлайн]. URL: <http://surl.li/brdjuw>. Дата звернення: 17.10.2024.
- [12] PHP documentation json_encode. [Онлайн]. URL: <http://surl.li/ejpfyb>. Дата звернення: 17.10.2024.
- [13] JSON Decode Online. [Онлайн]. URL: <http://surl.li/dsbdou>. Дата звернення: 17.10.2024.
- [14] PHP documentation password_hash. [Онлайн]. URL: <http://surl.li/vjrdlf>. Дата звернення: 17.10.2024.

DEVELOPMENT OF A CONTENT MANAGEMENT SYSTEM AND ITS CLOUD DEPLOYMENT

Yakiv (Jacob) Baytelman, Valerii Potsepaiev

The aim of this work is to create a simple and easily adaptable content management system that can be seamlessly integrated into web or mobile applications and customized according to the specific needs of a particular project. One of its key advantages is its versatility and flexibility, allowing developers to quickly and effortlessly tailor the software solution to the requirements of specific industries or business processes.

A comprehensive technical development process was carried out, resulting in the creation of software based on the use of the PHP programming language, the MySQL relational database, and the Amazon AWS cloud deployment environment. This combination of tools ensures high performance and reliability, which are critically important for modern applications. The work provides detailed explanations of the core methods and functions used for data processing and storage. All of the software code is available as open source.

The scientific novelty of this work lies in the presentation of the material in an accessible format, significantly simplifying the implementation of content management tasks even for individuals without deep programming knowledge. This opens up new possibilities for interdisciplinary research and collaboration across various fields, particularly in computer science and other applied disciplines such as marketing, journalism, or education.

From a practical standpoint, the presented materials allow for the deployment of a full-fledged content management system in the Amazon AWS cloud environment within a few hours. The software code, with appropriate modifications to implement different business logic, has been successfully integrated into several websites that are visited daily by thousands of users, demonstrating the effectiveness and stability of the proposed solution.

The materials of this work can be useful for educating students of technical and economic specialties, as well as senior high school students.

Keywords: content management, website, PHP, MySQL, AWS.

REFERENCES

- [1] PHP documentation. [Online]. URL: <https://www.php.net>. Accessed: 17.10.2024.
- [2] Understanding HTTP using Browsers. [Online]. URL: <http://surl.li/vwkvym>. Accessed: 17.10.2024.
- [3] MVC. [Online]. URL: <http://surl.li/ziawqy>. Accessed: 17.10.2024.
- [4] Introduction of Object Oriented Programming. [Online]. URL: <http://surl.li/slyzju>. Accessed: 17.10.2024.

- [5] What is a LAMP stack? [Online]. URL: <http://surl.li/gmesgm>. Accessed: 17.10.2024.
- [6] Tutorial: Install a LAMP server on AL2023. [Online]. URL: <http://surl.li/pkutti>. Accessed: 17.10.2024.
- [7] Amazon CloudFront. [Online]. URL: <http://surl.li/oqwnkq>. Accessed: 17.10.2024.
- [8] Amazon Route 53. [Online]. URL: <http://surl.li/tusvxf>. Accessed: 17.10.2024.
- [9] Amazon S3. [Online]. URL: <http://surl.li/qbziyg>. Accessed: 17.10.2024.
- [10] Amazon Relational Database Service. [Online]. URL: <http://surl.li/awling>. Accessed: 17.10.2024.
- [11] Amazon Lambda. [Online]. URL: <http://surl.li/brdjuw>. Accessed: 17.10.2024.
- [12] PHP documentation json_encode. [Online]. URL: <http://surl.li/ejpfyb>. Accessed: 17.10.2024.
- [13] JSON Decode Online. [Online]. URL: <http://surl.li/dsbdou>. Accessed: 17.10.2024.
- [14] PHP documentation password_hash. [Online]. URL: <http://surl.li/vjrdlf>. Accessed: 17.10.2024.

ІНФОРМАЦІЙНА СИСТЕМА ОБЛІКУ ВІДВІДУВАНOSTІ СТУДЕНТІВ

А.М. Рожко¹, Я.Ю. Дорогий¹, Г.О. Теличко²

¹ Department of Applied Mathematics and Informatics, Donetsk National Technical University, Luts'k, Ukraine

² Department of Automation and Telecommunications, Donetsk National Technical University, Luts'k, Ukraine

E-mail: alina.rozhko.kita@donntu.edu.ua

Отримано 17.10.2024

Прийнято до публікації 25.10.2024

Опубліковано 01.11.2024

АНОТАЦІЯ

У статті розглядається створення інформаційної системи для обліку відвідуваності студентів, що дозволяє оптимізувати процес фіксації присутності на заняттях. В умовах сучасної освітньої системи, де все більше навчальних закладів переходять на дистанційне навчання через безпекові виклики та необхідність забезпечення гнучкості навчального процесу, традиційні методи обліку, такі як паперові журнали, стають неефективними. Зберігати паперові журнали та передавати інформацію у фізичному вигляді складно і незручно, особливо в умовах у будь-якого форматі навчання. Вони також потребують значних витрат часу, підвищують ризик помилок та не забезпечують швидкого доступу до інформації або можливості оперативно генерувати звіти.

Таким чином, актуальність розробки інформаційної системи для обліку відвідуваності студентів полягає у необхідності впровадження сучасних цифрових рішень, що дозволяють легко та зручно вести облік незалежно від формату навчання. Це рішення допомагає полегшити адміністративні процеси, підвищити їх точність і забезпечити викладачам і адміністрації швидкий доступ до даних.

Для реалізації системи було обрано інтегроване середовище розробки IntelliJ IDEA і мову програмування Java, що забезпечує кросплатформенність і легку інтеграцію з різними операційними системами. У статті також описано архітектуру системи, що охоплює основні компоненти, включаючи графічний інтерфейс користувача та базу даних.

Практична значимість роботи полягає у створенні ефективного та зручного інструменту, який дозволяє значно полегшити процес обліку відвідуваності, підвищити точність даних і зменшити час на обробку інформації. Система підтримує функцію автоматичного генерування звітів, що дозволяє швидко отримувати інформацію про відвідуваність студентів за будь-який період, спрощуючи роботу викладачів та адміністрації. Крім того, система легко інтегрується з іншими освітніми платформами, що робить її гнучким та універсальним рішенням для різних навчальних закладів.

Ключові слова: інформаційна система, облік відвідуваності студентів, автоматизація, IntelliJ IDEA, ООП, Java.

ВСТУП

У сучасній системі освіти значну роль відіграє управління академічним процесом, зокрема контроль відвідуваності студентів. Традиційні методи обліку, такі як паперові журнали, стають дедалі менш ефективними в умовах зростаючого обсягу інформації та збільшення кількості студентів. Викладачі стикаються з проблемами витрат часу, людськими помилками та обмеженими можливостями паперових систем. Цифрові технології та інформаційні системи відкривають нові можливості для автоматизації обліку відвідуваності [1], забезпечуючи значні переваги в управлінні навчальним процесом. Використання інформаційної системи обліку дозволяє не лише оптимізувати адміністративні процеси [2], а й підвищити загальну якість навчання, забезпечивши прозорий контроль присутності студентів.

АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ ТА ПОСТАНОВКА ПРОБЛЕМИ

У сучасній освітній практиці все більша увага приділяється автоматизації обліку відвідуваності та моніторингу успішності студентів. Ряд дослідників вивчали ефективність і можливості таких систем. Зокрема, Васильєв В. у своїй роботі [3] розглядає систему обліку відвідуваності як ефективний інструмент моніторингу рівня знань студентів. Автор підкреслює, що автоматизація процесу обліку дозволяє значно підвищити якість контролю за відвідуваністю і точність оцінювання академічної успішності.

Топольський А. І. та Паламарчук Є. А. [4] провели аналіз практичних реалізацій автоматизованих систем ідентифікації студентів у електронних навчальних системах. Їх дослідження вказує на необхідність інтеграції обліку відвідуваності з існуючими системами управління навчанням (LMS), що забезпечує не тільки контроль присутності, але й можливість автоматичного оновлення даних про успішність студентів. Вони також відзначають важливість використання біометричних технологій та RFID, які значно підвищують точність і надійність даних.

Значну увагу питанню впливу відвідуваності на академічну успішність студентів приділили Credé, Roch і Kieszczyńska [5], які в своєму метааналізі дослідили, як відвідуваність занять у коледжах впливає на рівень академічної успішності. Вони зазначають, що студенти, які регулярно відвідують заняття, демонструють вищі академічні досягнення. Це підтверджує важливість

впровадження ефективних систем обліку відвідуваності для підвищення успішності студентів.

Робота Alghamdi [6] фокусується на використанні "розумних" систем для моніторингу відвідуваності студентів у Taif University. Автор наголошує на перевагах автоматизованих систем, які використовують сучасні технології, такі як розпізнавання обличчя та мобільні додатки для фіксації присутності. Це забезпечує зручність як для викладачів, так і для студентів, зменшуючи ризик помилок та скорочуючи час, необхідний для ведення обліку.

З огляду на наведені дослідження, можна стверджувати, що автоматизовані системи обліку відвідуваності є важливим інструментом для підвищення якості освітнього процесу. Вони не лише спрощують адміністрування, але й створюють умови для більш точного контролю успішності студентів.

Метою даного дослідження є розробка та впровадження інформаційної системи обліку відвідуваності студентів, яка забезпечить автоматизацію процесів збирання та обробки даних про присутність студентів на заняттях. Система повинна мати простий та інтуїтивно зрозумілий інтерфейс, бути доступною для використання на різних платформах та підтримувати зберігання даних у базі MySQL. Додатково досліджується її ефективність з точки зору економії часу та ресурсів викладачів.

Завдання дослідження:

1. Провести аналіз існуючих підходів та рішень у сфері обліку відвідуваності студентів для визначення найбільш ефективних методів реалізації.
2. Вибрати середовище розробки та мову програмування, що найкраще підходять для створення інформаційної системи обліку відвідуваності студентів.
3. Проектувати архітектуру інформаційної системи з використанням UML-діаграм для моделювання взаємодії між основними компонентами та процесами системи.
4. Розробити базу даних для зберігання даних про студентів, групи та відвідуваність, визначити ключові індекси для оптимізації запитів.
5. Створити графічний інтерфейс користувача для спрощення роботи з даними про відвідуваність.
6. Оцінити ефективність інформаційної системи шляхом порівняння її з традиційними методами обліку відвідуваності на основі швидкості введення та точності даних.

МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ

Для розробки інформаційної системи обліку відвідуваності студентів використовувалися IntelliJ IDEA як середовище розробки для мови Java та MySQL для зберігання даних. Основні методи включали аналіз існуючих рішень, проектування архітектури системи з використанням UML-діаграм, розробку оптимізованої бази даних, створення графічного інтерфейсу для зручного управління даними та оцінку ефективності системи шляхом порівняння швидкості введення і точності даних з традиційними методами обліку.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для створення інформаційної системи обліку відвідуваності студентів було розроблено кілька UML-діаграм для планування архітектури системи та візуалізації її компонентів. Це дозволило чітко визначити взаємодію між класами, процеси в системі та її компоненти.

Діаграма прецедентів (use case diagram) відображає основні сценарії взаємодії користувача із системою [9]. Користувач має можливість додавати, редагувати або видаляти інформацію про студентів, а також завантажувати та зберігати дані (Рис. 1).

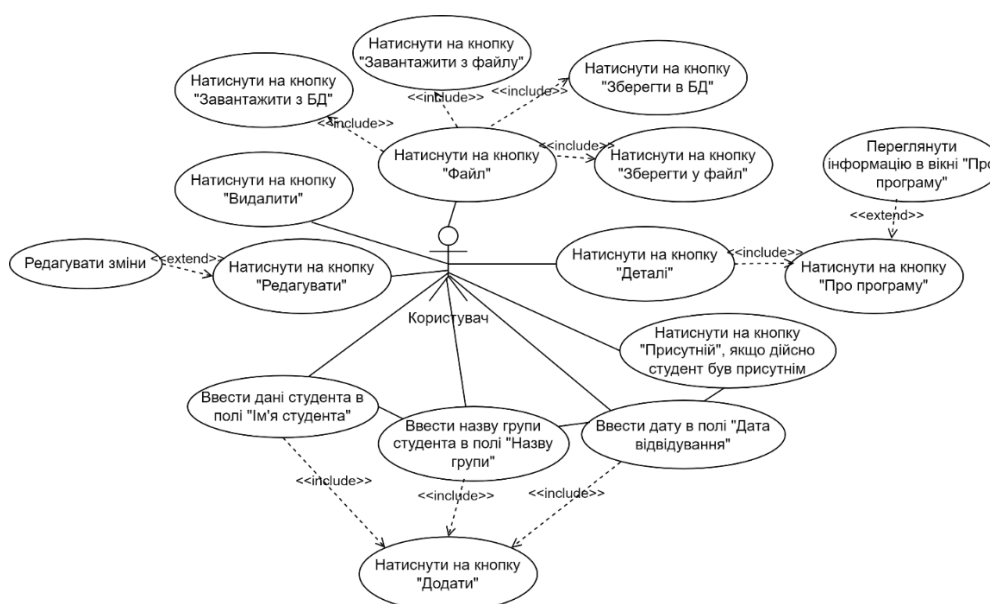


Рис. 1. Діаграма прецедентів системи обліку відвідуваності студентів

На діаграмі прецедентів представлені всі операції, які може виконувати користувач системи: введення даних про студентів, редагування, видалення, завантаження та збереження інформації у базі даних чи в файл у будь-якому пристрою.

Діаграма розгортання (deployment diagram) відображає, як компоненти системи розгорнуті на апаратних ресурсах. Вона показує, як система працює на кінцевих пристроях і взаємодіє з сервером бази даних.

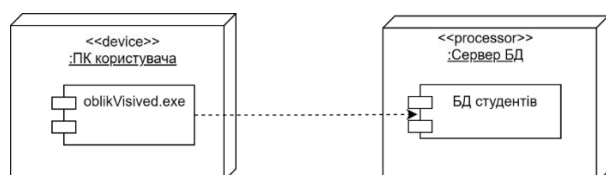


Рис. 2. Діаграма розгортання системи обліку відвідуваності студентів

Система обліку відвідуваності розгорнута на персональному комп'ютері користувача з підключенням до сервера, на якому знаходиться база даних MySQL. Це дозволяє користувачам працювати із системою через графічний інтерфейс, при цьому дані зберігаються централізовано на сервері.

Діаграма класів (class diagram) демонструє ключові класи системи та їх взаємозв'язки. Основні класи включають сутності, пов'язані з обліком відвідуваності, такими як Attendance (присутність студента) і DatabaseConnection (з'єднання з базою даних).

Діаграма класів показує структуру системи, її основні класи, а також їх атрибути і методи. Клас Attendance зберігає інформацію про кожного студента, а DatabaseConnection забезпечує підключення до MySQL бази даних.

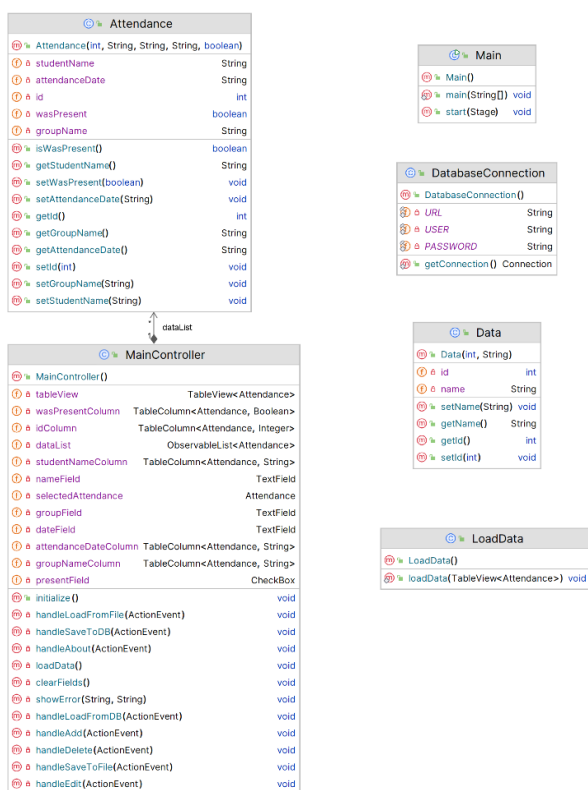


Рис. 3. Діаграма класів для обліку відвідуваності студентів

Діаграма модулів (component diagram) відображає взаємодію між основними компонентами системи. Це дозволяє побачити, як різні частини системи пов'язані між собою і як вони забезпечують роботу всієї системи (Рис. 4).

Основні компоненти включають графічний інтерфейс користувача (GUI), модуль взаємодії з базою даних і контролер. Контролер координує роботу між інтерфейсом і базою даних, забезпечуючи плавну роботу системи.

Для розробки інформаційної системи обліку відвідуваності студентів використовувалося інтегроване середовище розробки IntelliJ IDEA. Це потужний інструмент, що надає зручні можливості для розробки на мовах Java, Python, JavaScript, PHP тощо [8]. IntelliJ IDEA має вбудовані засоби для рефакторингу коду, автодоповнення та багатофункціональний інтерфейс, що робить її ідеальним середовищем для розробників на Java.

Основною мовою програмування було обрано Java, завдяки її платформи-незалежності та кросплатформенності. Java дозволяє розробляти програми, що працюють на різних операційних системах без необхідності адаптації коду під конкретні платформи.

Це забезпечує високу гнучкість та можливість використання програми на будь-яких пристроях.

Для зберігання даних використано реляційну базу даних MySQL. MySQL — це одна з найпопулярніших систем керування базами даних, що забезпечує високу продуктивність та надійність при роботі з великими обсягами інформації [10].

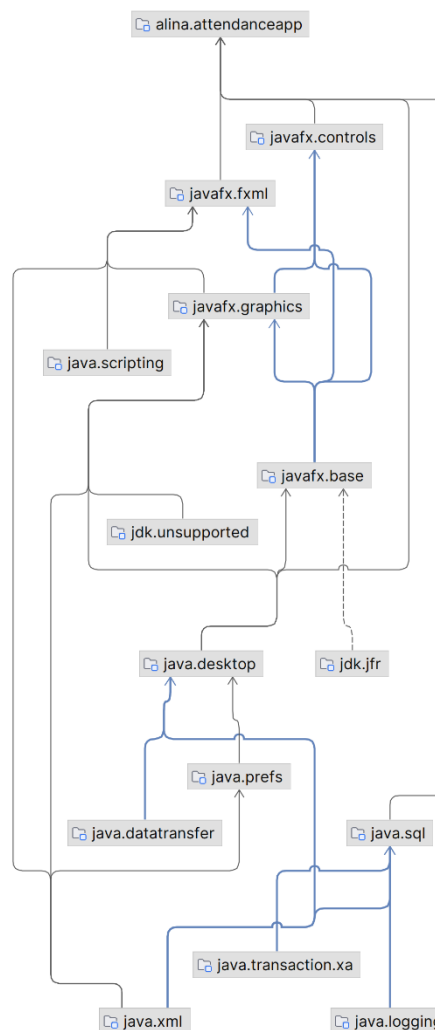


Рис. 4. Діаграма модулів системи обліку відвідуваності студентів

Таблиця attendance, створена для обліку відвідуваності, містить такі основні поля:

- id (int) – ідентифікатор запису (первинний ключ, автоінкремент);
- student_name (varchar(100)) – ім'я студента;
- group_name (varchar(50)) – назва групи;
- attendance_date (date) – дата відвідування;
- was_present (tinyint(1)) – показник присутності (1 – присутній, 0 – відсутній).

Для забезпечення швидкого доступу до даних у таблицях було створено індекси, що підвищують швидкість виконання запитів. Індекси допомагають оптимізувати процес пошуку інформації у великих масивах даних, зменшуючи час, необхідний для виконання запиту до бази.

```

19 -- Table structure for table `attendance`
20 --
21
22 DROP TABLE IF EXISTS `attendance`;
23 /*!40101 SET @saved_cs_client = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `attendance` (
26   `id` int NOT NULL AUTO_INCREMENT,
27   `student_name` varchar(100) NOT NULL,
28   `group_name` varchar(50) NOT NULL,
29   `attendance_date` date NOT NULL,
30   `was_present` tinyint(1) NOT NULL,
31   PRIMARY KEY (`id`)
32 ) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
33 /*!40101 SET character_set_client = @saved_cs_client */;
34
35 --
36 -- Dumping data for table `attendance`
37 --
38

```

Рис. 5. Структура бази даних MySQL для обліку відвідуваності студентів

Visible	Key	Type	Uni...	Columns
<input checked="" type="checkbox"/>	PRIMARY	BTREE	YES	id

Column	Type	Nullable	Indexes
id	int	NO	PRIMARY
student_name	varchar(100)	NO	
group_name	varchar(50)	NO	
attendance_date	date	NO	
was_present	tinyint(1)	NO	

Рис. 6. Таблиця індексів в MySQL

Для підключення до бази даних було створено клас DatabaseConnection.java, який відповідає за з'єднання системи з базою через URL, ім'я користувача та пароль.

```

1 package alina.attendanceapp;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class DatabaseConnection {
8     private static final String URL = "jdbc:mysql://localhost:3306/bd88"; 1 usage
9     private static final String USER = "root"; 1 usage
10    private static final String PASSWORD = "Alina25072002"; 1 usage
11
12    public static Connection getConnection() throws SQLException { 6 usages
13        return DriverManager.getConnection(URL, USER, PASSWORD);
14    }
15 }

```

Рис. 7. Клас DatabaseConnection.java для підключення бази даних MySQL в середовищі IntelliJ IDEA

ОБГОВОРЕННЯ РЕЗУЛЬТАТІВ

В результаті реалізації інформаційної системи було створено повнофункціональну базу даних для обліку відвідуваності студентів. Таблиця attendance дозволяє зберігати інформацію про кожного студента, групу та дати відвідуваності, а індекси, створені в MySQL, забезпечують швидкий доступ до інформації навіть при великому обсязі даних. Це суттєво полегшує роботу викладачів, дозволяючи їм ефективно контролювати присутність студентів на заняттях та швидко генерувати звіти.

На Рисунок 8 показано інтерфейс системи, який надає можливість викладачам зручно вносити інформацію про відвідуваність студентів. Інтерфейс розроблено таким чином, щоб він був інтуїтивно зрозумілим і дозволяв швидке внесення даних про присутність або відсутність студентів. Крім того, система дозволяє експортувати дані в формат CSV, що забезпечує гнучкість у використанні.

Тестування системи показало, що використання автоматизованого обліку значно скорочує час, необхідний для ведення звітності, порівняно з традиційними паперовими методами.

Ефективність системи проявляється також у зниженні кількості помилок, які виникають при ручному введенні даних, та у можливості швидкого доступу до історичних даних про відвідуваність. Крім того, система легко масштабована, що дозволяє її використовувати в навчальних закладах будь-якого рівня, незалежно від кількості студентів.

ID	Ім'я студента	Група	Дата відвідування	Присутній
1	Дробчак Віталій	КН-20	2024-06-01	true
2	Лубовець Дмитро	КН-20	2024-06-01	false
3	Мельник Данило	КН-20	2024-06-01	true
4	Стоянович Любомир	КН-20	2024-06-01	true
6	Троцький Максим	КН-20	2024-06-01	false
7	Тютюнник Микола	КН-20	2024-06-01	true
8	Бондарчук Софія	ІПЗ-20	2024-06-01	true
9	Добродомов Дмитро	ІПЗ-20	2024-06-01	true
10	Донець Дмитро	ІПЗ-20	2024-06-01	false
11	Мартинцов Олександр	ІПЗ-20	2024-06-01	true
12	Сергієнко Олександр	ІПЗ-20	2024-06-01	true
13	Мороз Євген	КІБ-20	2024-06-01	false
14	Паращевін Марк	КІБ-20	2024-06-01	false

Рис. 8. Інтерфейс інформаційної системи обліку відвідуваності студентів

Подальші кроки в розвитку системи включають інтеграцію з мобільними додатками для швидкого внесення даних зі смартфонів та впровадження додаткових модулів для аналізу відвідуваності, що

дозволить покращити якість контролю навчального процесу.

ВИСНОВКИ

Розроблена інформаційна система обліку відвідуваності студентів надає низку переваг, зокрема економію часу та ресурсів для викладачів, а також підвищення точності та зручності зберігання інформації. Вибір Java та MySQL як технологій для реалізації забезпечує масштабованість та стабільність системи. Подальші дослідження можуть бути спрямовані на розширення функціоналу, зокрема додавання можливостей інтеграції з іншими системами управління навчанням.

ЛІТЕРАТУРА

- [1] О.В. Грицунов, “Інформаційні системи та технології: навч. посіб. для студентів за напрямом підготовки “Транспортні технології” / О.В. Грицунов; Харк. нац. акад. міськ. госп-ва. – Х.: ХНАМГ, 2010. – 222 с. [Онлайн]. URL: <http://surl.li/hrnzan>. Дата звернення: 14.10.2024.
- [2] І.А. Жуков, І.А. Клименко, І.М. Кравець, “Адміністративно-навчальна інформаційна система інституту комп’ютерних технологій”. Проблеми інформатизації та управління: зб. наук. пр., vol. 1, no. 19, pp. 56–67, 2007. doi: 10.18372/2073-4751.1.9059
- [3] В. М. Васильєв, “Система обліку відвідуваності й успішності студентів – як засіб моніторингу рівня знань у студентів”. Ukrainian Journal of Educational Studies and Information Technology, vol. 3, no. 1, pp. 12–16, 2016. doi: 10.32919/uesit.2016.01.12-16
- [4] А. І. Топольський, С. А. Паламарчук, “Аналіз практичних реалізацій автоматизованих систем ідентифікації студентів в електронних навчальних системах”. Вісник Вінницького політехнічного інституту, vol. 2, pp. 61–70, 2024. doi: 10.31649/1997-9266-2024-173-2-61-70
- [5] M. Credé, S. G. Roch, U. M. Kieszczyńska, “Class Attendance in College,” Review of Educational Research, vol. 80, no. 2, pp. 272–295, 2010. doi: 10.3102/0034654310362998
- [6] S. Alghamdi, “Monitoring student attendance using a smart system at Taif University”. International Journal of Computer Science & Information Technology (IJCSIT), vol. 11, no. 1, pp. 107–115, 2019. doi: 10.5281/zenodo.26024204.
- [7] О. М. Васильєв, “Програмування мовою Java”. Тернопіль: Навчальна книга – Богдан, 2020. — 696 с. [Онлайн]. URL: <http://surl.li/szxcjo>. Дата звернення: 14.10.2024.
- [8] IntelliJ IDEA. [Онлайн]. URL: <http://surl.li/gnhcux>. Дата звернення: 14.10.2024.
- [9] Use case diagram. [Онлайн]. URL: <http://surl.li/zxgkiz>. Дата звернення: 14.10.2024.

- [10] MySQL Explained: Your Guide to Mastering This Powerful Database. [Онлайн]. URL: <http://surl.li/ilwwza>. Дата звернення: 14.10.2024.

INFORMATION SYSTEM FOR RECORDING STUDENT ATTENDANCE

Alina Rozhko, Iaroslav Dorohyi, Hanna Telychko

The article discusses the creation of an information system for recording student attendance, which allows optimising the process of recording attendance at classes. In today's educational system, where more and more educational institutions are switching to distance learning due to security challenges and the need to ensure flexibility in the learning process, traditional accounting methods such as paper logs are becoming inefficient. Storing paper logs and transferring information in physical form is difficult and inconvenient, especially in any learning format. They are also time-consuming, increase the risk of errors, and do not provide quick access to information or the ability to generate reports quickly.

Thus, the relevance of developing an information system for recording student attendance lies in the need to implement modern digital solutions that allow for easy and convenient record-keeping regardless of the learning format. This solution helps to facilitate administrative processes, improve their accuracy and provide teachers and administration with quick access to data.

To implement the system, we chose the IntelliJ IDEA integrated development environment and the Java programming language, which provides cross-platform compatibility and easy integration with various operating systems. The article also describes the system architecture, covering the main components, including the graphical user interface and the database.

The practical significance of the work is to create an effective and convenient tool that can significantly facilitate the process of attendance accounting, improve data accuracy and reduce the time for information processing. The system supports the function of automatic generation of reports, which allows you to quickly receive information on student attendance for any period, simplifying the work of teachers and administration. In addition, the system can be easily integrated with other educational platforms, making it a flexible and versatile solution for various educational institutions.

Keywords: information system, student attendance, automation, IntelliJ IDEA, OOP, Java.

REFERENCES

- [1] Hrytsunov, O.V., “Information Systems and Technologies: a textbook for students in the field of study ‘Transport Technologies’” / O.V. Hrytsunov; Kharkiv National Academy of Municipal Economy – Kharkiv: KhNAMG, 2010. 222 p. [Online]. URL: <http://surl.li/hrnzan>. Accessed: 14.10.2024.
- [2] I.A. Zhukov, I.A. Klimenko, I.M. Kravets, “Administrative and educational information system of the Institute of Computer Technologies”. Problems of informatisation and management: a collection of scientific papers, vol. 1, no. 19, pp. 56–67, 2007. doi: 10.18372/2073-4751.1.9059
- [3] V. M. Vasiliev, “The system of attendance and academic performance of students – as a means of monitoring the level of knowledge of students”. Ukrainian Journal of Educational Studies and Information Technology, vol. 3, no. 1, pp. 12–16, 2016. doi: 10.32919/uesit.2016.01.12-16
- [4] A. I. Topolsky, E. A. Palamarchuk, “Analysis of practical implementations of automated student identification systems in electronic learning systems”. Bulletin of Vinnytsia Polytechnic Institute, vol. 2, pp. 61–70, 2024. doi: 10.31649/1997-9266-2024-173-2-61-70
- [5] M. Credé, S. G. Roch, U. M. Kieszczyńska, “Class Attendance in College,” Review of Educational Research, vol. 80, no. 2, pp. 272–295, 2010. doi: 10.3102/0034654310362998
- [6] S. Alghamdi, “Monitoring student attendance using a smart system at Taif University”. International Journal of Computer Science & Information Technology (IJCSIT), vol. 11, no. 1, pp. 107–115, 2019. doi:10.5281/zenodo.26024204.
- [7] O. M. Vasyliiev, “Java Programming”. Ternopil: Training book – Bogdan, 2020. 696 p. [Online]. URL: <http://surl.li/szxjco>. Accessed: 14.10.2024.
- [8] IntelliJ IDEA. [Online]. URL: <http://surl.li/gnhcux>. Accessed: 14.10.2024.
- [9] Use case diagram. [Online]. URL: <http://surl.li/zxgkiz>. Accessed: 14.10.2024.
- [10] MySQL Explained: Your Guide to Mastering This Powerful Database. [Online]. URL: <http://surl.li/ilwwza>. Accessed: 14.10.2024.

КОМПАРАТИВНИЙ АНАЛІЗ МОДЕЛЕЙ ДЛЯ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ФЕЙКОВИХ НОВИН З ВИКОРИСТАННЯМ GRU

В.М. Коваленко¹, Я.Ю. Дорогий¹, К.С. Дорошенко²

¹ Department of Automation and Telecommunications, Donetsk National Technical University, Luts'k, Ukraine

² Department of Information Systems and Technologies, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

E-mail: vitalii.kovalenko.kita@donntu.edu.ua

Отримано 15.04.2024

Прийнято до публікації 01.08.2024

Опубліковано 01.11.2024

АНОТАЦІЯ

У статті представлено компаративний аналіз моделей для виявлення та класифікації фейкових новин із використанням GRU (gated recurrent unit) – сучасної архітектури нейронних мереж, що є альтернативою LSTM. Метою дослідження є оцінка ефективності моделі GRU порівняно з іншими поширеними моделями обробки природної мови (NLP), такими як BERT, RoBERTa, LSTM, у контексті задачі ідентифікації фейкових новин. Актуальність тематики зумовлена необхідністю точного і своєчасного виявлення дезінформації в сучасному інформаційному просторі, що впливає на суспільні процеси та прийняття рішень.

Методологія дослідження базується на компаративному аналізі із використанням певних критеріїв. GRU, як рекурентна нейронна мережа, має спрощену архітектуру порівняно з LSTM, що робить її менш ресурсоємною, зберігаючи при цьому здатність обробляти довгі послідовності тексту. Основний акцент робиться на порівнянні продуктивності GRU з іншими моделями у завданнях виявлення та класифікації фейкових новин, з урахуванням особливостей обробки контексту.

Результати компаративного аналізу демонструють, що GRU забезпечує конкурентні результати з точки зору точності і швидкості навчання, порівняно з LSTM та трансформерними моделями (BERT, RoBERTa), особливо в умовах обмежених обчислювальних ресурсів. Модель GRU показує ефективність при роботі з великими обсягами тексту та при аналізі складних контекстуальних зв'язків. Завдяки меншій складності архітектури, GRU є перспективною моделлю для впровадження в системи моніторингу та виявлення фейкових новин у реальному часі.

Наукова новизна статті полягає в дослідженні ефективності GRU у порівнянні з іншими NLP-моделями для задач класифікації тексту, що дозволяє покращити процеси ідентифікації дезінформації. Практична значимість роботи полягає в тому, що результати можуть бути як рекомендації щодо вибору певного класу моделей для розв'язання різних видів завдань при

розробці систем боротьби з фейковими новинами у різних сферах, включно зі ЗМІ, соціальними мережами та аналітичними центрами.

Ключові слова: виявлення фейкових новин, класифікація тексту, гейтована рекурентна одиниця (GRU), глибоке навчання, обробка природної мови (NLP), точність алгоритму, оптимізація обчислень, система класифікації новин, машинне навчання, аналіз текстових даних

ВСТУП

Нейронні мережі відіграють ключову роль у сучасних технологіях, зокрема в аналізі та виявленні фейкових новин. Ці складні алгоритми, натхненні структурою людського мозку, здатні вивчати, аналізувати та прогнозувати дані, що робить їх надзвичайно потужними інструментами для обробки інформації. Використання нейронних мереж відкриває нові можливості у таких областях, як обробка природної мови, де вони можуть ідентифікувати та класифікувати новини на основі їхньої достовірності.

Завдяки здатності навчатися на великих обсягах даних, нейронні мережі забезпечують високий рівень точності у вирішенні складних завдань, зокрема у виявленні фейкових новин. Наприклад, моделі BERT, RoBERTa та XLNet використовуються для аналізу контексту та семантики тексту, що дозволяє ефективно визначати патерни, які можуть свідчити про дезінформацію. LSTM (Long Short-Term Memory) може аналізувати послідовності слів та виявляти емоційні відтінки, що допомагає виявити маніпулятивні прийоми у текстах.

Інтернет-ресурси, такі як Google Colab, Kaggle та інші платформи, надають доступ до потужних обчислювальних ресурсів, необхідних для ефективного навчання моделей для виявлення фейкових новин. Це знижує бар'єри для входу у цю галузь, дозволяючи дослідникам та розробникам з усього світу експериментувати з новими алгоритмами та ідеями.

Крім того, нейронні мережі, такі як HAN (Hierarchical Attention Networks) і ESIM (Enhanced Sequential Inference Model), використовують механізми уваги для аналізу важливих частин тексту. Це дозволяє їм зосередитися на елементах, які мають значення для ідентифікації фейкових новин, ігноруючи непотрібну інформацію. Таким чином, моделі можуть виявляти маніпуляції, перевіряти факти та класифікувати новини на справжні чи фейкові.

Інтернет-ресурси забезпечують доступ до широкого спектра навчальних матеріалів, включаючи курси, документацію та тьюторіали, що дозволяє новачкам

освоювати основи аналізу та виявлення фейкових новин. Вебінари та онлайн-спільноти сприяють обміну знаннями та досвідом, що важливо для швидкого розвитку технологій у цій динамічній сфері.

ПОСТАНОВКА ПРОБЛЕМИ

У умовах стрімкого розвитку інформаційних технологій та зростання обсягу інформації, що поширюється через соціальні мережі, все більш актуальною стає проблема фейкових новин. Фейкові новини можуть мати серйозні негативні наслідки, включаючи дезінформацію суспільства, вплив на політичні процеси, формування помилкових думок та загальну недовіру до медіа. З огляду на це, виникає необхідність у розробці автоматизованих систем, здатних ефективно виявляти та класифікувати фейкові новини.

Традиційні методи для виявлення фейкових новин базуються на ручній перевірці фактів або простих лінгвістичних аналізах, що є трудомістким і повільним процесом. У зв'язку з цим, використання сучасних методів глибокого навчання, таких як Gated Recurrent Unit (GRU), набуває все більшої популярності. GRU дозволяє будувати моделі, які можуть ефективно обробляти великі обсяги текстових даних і враховувати контекст новин для точнішого визначення їхньої достовірності.

Проте, незважаючи на ефективність цих методів, залишаються відкритими питання щодо підвищення точності класифікації, адаптації моделей до нових джерел даних та зменшення кількості помилкових спрацьовувань. Таким чином, розробка системи для виявлення та класифікації фейкових новин із використанням GRU є важливою науковою та практичною **задачею**, яка спрямована на зниження впливу дезінформації у сучасному інформаційному середовищі.

Метою даної роботи є аналіз ролі нейронних мереж у виявленні фейкових новин, а також вивчення впливу інтернет-ресурсів на процес навчання цих моделей. Завдяки інтеграції різних платформ з бібліотеками, такими як TensorFlow і PyTorch, користувачі можуть легко реалізувати свої проекти, що стимулює інновації і дозволяє швидше переходити від теорії до практики в

аналізі текстів новин. Таким чином, значення нейронних мереж у сучасних технологіях не лише в їхній здатності вирішувати практичні задачі, але й у спрощенні доступу до потужних інструментів для навчання та розвитку у цій галузі.

Завдання дослідження:

1. Провести детальний огляд існуючих методів та моделей глибокого навчання, що використовуються для класифікації фейкових новин.

2. Описати архітектуру та функціональні можливості нейронних мереж BERT, RoBERTa, XLNet, LSTM, HAN та ESIM.

3. Порівняти ефективність цих моделей на основі результатів, отриманих в різних наукових дослідженнях та експериментах.

4. Виявити переваги та недоліки кожної моделі в контексті задачі виявлення фейкових новин.

5. Розробити рекомендації щодо вибору нейронних мереж для подальших досліджень і практичного використання у виявленні дезінформації.

МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ

В роботі використано методи структурного і порівняльного аналізу, з інформаційним і аналітичним підходом розглянуто наукову та методичну літературу, а також онлайн ресурси для виділення та категоризації проблем тематики дослідження, а також підходів, що використовуються для проведення необхідного аналізу моделей, які використовуються для виявлення фейкових новин.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Проведемо детальний аналіз основних існуючих методів та моделей глибокого навчання для пошуку фейкових новин.

BERT

BERT (Bidirectional Encoder Representations from Transformers) — це модель глибокого навчання на основі трансформерів, розроблена компанією Google у 2018 році. Вона є однією з найпотужніших моделей для обробки природної мови (NLP), яка виявилася надзвичайно ефективною для різних задач, таких як класифікація тексту, заповнення пропусків у текстах, переклад та виявлення фейкових новин [1].

BERT відрізняється від інших моделей, таких як GPT, своєю двонаправленою природою. Це означає, що при аналізі тексту BERT враховує контекст з обох боків слова — як попередній, так і наступний. Інші моделі зазвичай працюють односпрямовано, аналізуючи текст лише зліва

направо або справа наліво. Двонаправленість дає можливість BERT краще розуміти значення слів у контексті всього речення [1].

BERT попередньо навчається на великому обсязі текстових даних (наприклад, Вікіпедія та інші тексти), виконуючи два основні завдання:

Масковане моделювання мови (Masked Language Model, MLM). Частина слів у тексті "маскують" (закривають), і модель повинна передбачити ці слова на основі контексту навколо них.

Прогнозування наступного речення (Next Sentence Prediction, NSP). Модель навчається передбачати, чи є одне речення логічним продовженням іншого. Це корисно для задач, пов'язаних із текстовими послідовностями, такими як виявлення фейкових новин, де важливо розуміти зміст і логічний зв'язок між пропозиціями.

BERT побудована на базі трансформера — механізму, що використовує самоувагу (self-attention), дозволяючи моделі ефективно обробляти довгі послідовності тексту. Самоувага допомагає кожному слову враховувати інші слова у реченні, що забезпечує глибший рівень розуміння контексту (Рис. 1).

Перевагами BERT є:

1) *Універсальність*. BERT є універсальною моделлю для широкого спектра задач NLP, включаючи класифікацію текстів, розпізнавання іменованих сутностей, пошук інформації, аналіз настроїв та питання-відповіді. Це означає, що після попереднього навчання її можна "доучити" для конкретних завдань з мінімальними витратами часу та ресурсів [1].

2) *Попереднє навчання на великих наборах даних*. Модель BERT попередньо навчена на величезних обсягах текстових даних, таких як Вікіпедія та інші масиви даних. Це дає їй глибоке розуміння структури мови, завдяки чому вона добре працює навіть у випадках, коли надано мало навчальних прикладів для конкретної задачі.

3) *Висока точність у багатьох NLP-завданнях*. BERT демонструє високі результати у різних задачах обробки текстів, що підтверджується досягненням рекордних показників на багатьох стандартних тестах (GLUE, SQuAD тощо). Це робить модель одним із найкращих виборів для вирішення складних задач, таких як виявлення фейкових новин, автоматичне перекладення та розуміння складних текстів.

4) *Гнучкість у адаптації*. Завдяки використанню трансформерів, BERT може ефективно працювати з довгими текстами, враховуючи складні залежності між словами. Модель легко адаптується для нових задач через процес доучування (fine-tuning), що дозволяє

швидко застосовувати її у різних галузях — від пошукових систем до аналізу соціальних медіа.

5) *Підтримка багатомовності*. BERT підтримує різні мови, включаючи українську, російську та інші. Це

важливо для глобальних задач NLP, де потрібна робота з текстами на багатьох мовах. Мультинаціональна версія BERT (mBERT) успішно працює з багатьма мовами одночасно.

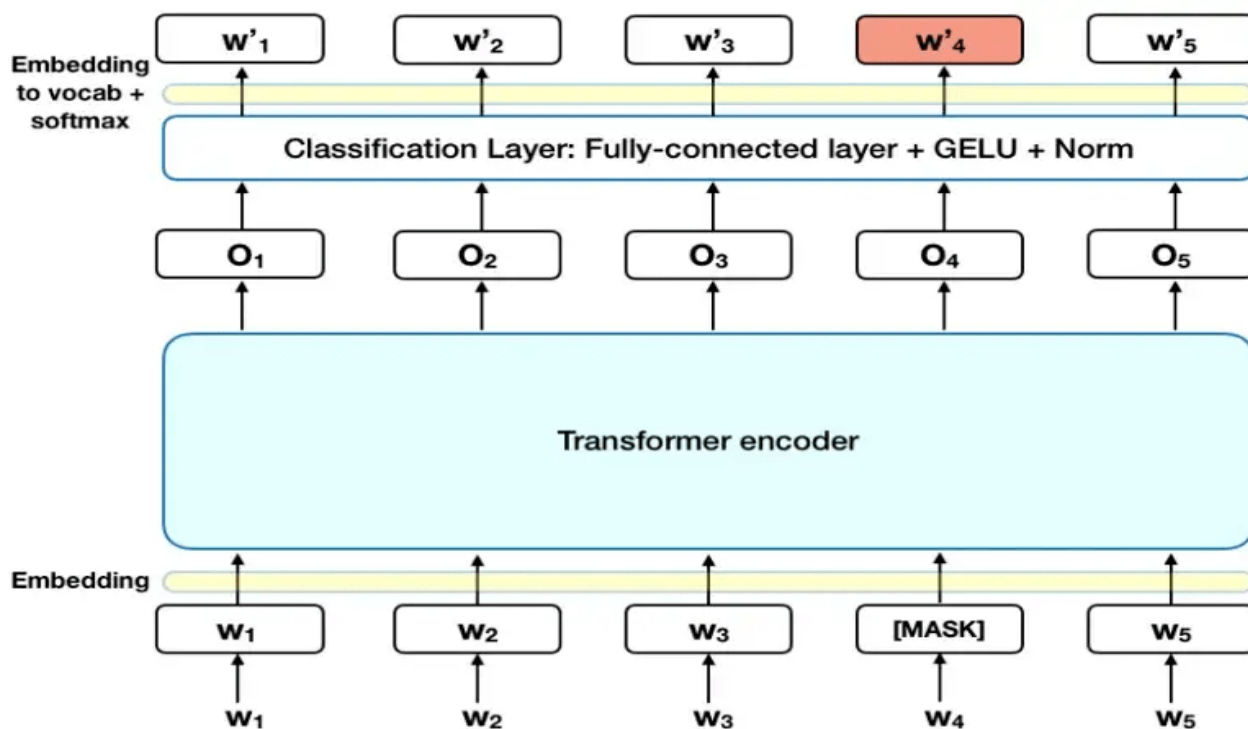


Рис. 1. Структура BERT

Серед недоліків можна назвати наступні:

1) *Високі обчислювальні вимоги*. BERT — це модель з великою кількістю параметрів (зокрема, базова версія BERT має 110 мільйонів параметрів, а BERT Large — 340 мільйонів). Це означає, що для її тренування та інференсу потрібні значні обчислювальні ресурси, включаючи потужні графічні процесори (GPU) або тензорні процесори (TPU). Для деяких компаній або дослідників це може стати перешкодою через високу вартість обчислювальної інфраструктури.

2) *Великий розмір моделі*. Модель BERT, особливо її розширені версії (BERT Large), має значний обсяг пам'яті, що ускладнює її використання на пристроях із обмеженими ресурсами, таких як смартфони або мікроконтролери. Оптимізація та стиснення моделі (через техніки, такі як знекорчування або квантизація) часто необхідні для її ефективного використання в реальному часі.

3) *Проблеми зі швидкістю*. Хоча BERT дає чудові результати, її двонаправлений підхід вимагає більше обчислень, що знижує швидкість обробки тексту

порівняно з односпрямованими моделями, такими як GPT. Це особливо важливо в додатках, де швидкість є критичною (наприклад, у реальному часі).

4) *Нечутливість до доменних знань*. Незважаючи на те, що BERT навчається на великому наборі загальних текстів, він може показувати гірші результати для специфічних доменів (наприклад, у медичних або юридичних текстах). Для таких задач необхідне додаткове навчання на специфічних наборах даних, що може вимагати додаткових ресурсів і часу.

5) *Відсутність інтерпретованості*. Як і більшість сучасних глибоких нейронних мереж, BERT є "чорною скринькою". Це означає, що важко зрозуміти, як і чому модель приймає певні рішення. У деяких випадках, таких як виявлення фейкових новин або судові процеси, це може стати проблемою, оскільки важливо мати можливість пояснити результати моделі.

6) *Чутливість до неправдивої інформації*. BERT може бути вразливою до добре сформульованих неправдивих текстів або маніпулятивної інформації. Наприклад, фейкові новини, які навмисно створені з

використанням реальних фактів або маніпуляцій, можуть залишитися непоміченими або неправильно класифікованими, оскільки модель базується на лінгвістичному аналізі, а не на фактичній перевірці даних.

7) *Відсутність підтримки роботи у реальному часі.* Через великі вимоги до ресурсів та високе навантаження, BERT важко використовувати для обробки великих потоків даних у реальному часі без спеціальних оптимізацій. Для задач, де потрібна миттєва реакція, можуть бути кращими інші, менш ресурсомісткі моделі [1].

BERT використовується для наступних задач:

1) *Виявлення фейкових новин.* BERT широко використовується для виявлення дезінформації та фейкових новин. Модель навчається на великих наборах новинних даних для ідентифікації фейків на основі контексту та семантичного аналізу тексту. BERT аналізує структуру тексту, ключові слова та фрази, щоб виявити неправдиву інформацію або маніпулятивні заголовки, що дозволяє автоматизувати процес перевірки новин [2].

2) *Пошукові системи (Google Search).* Google інтегрував BERT у свою пошукову систему для покращення розуміння запитів користувачів. Модель допомагає краще інтерпретувати довгі та складні пошукові запити, враховуючи контекст кожного слова, що покращує точність результатів пошуку. Наприклад, BERT дозволяє Google розуміти відтінки значень запитів та забезпечувати більш релевантні відповіді [3].

3) *Автоматизовані системи запитань та відповідей (Q&A Systems).* BERT використовується в системах запитання-відповідь для надання точних відповідей на запити користувачів. Це корисно для автоматизованих служб підтримки клієнтів та чат-ботів. Наприклад, якщо користувач запитує інформацію про товар або послугу, BERT може швидко знайти та надати релевантну відповідь із бази знань або документації [2].

4) *Аналіз настроїв у соціальних медіа.* Модель BERT може бути використана для аналізу настроїв у текстах соціальних мереж, блогах та відгуках. Завдяки своїй здатності розуміти контекст, вона ефективно класифікує тексти на позитивні, негативні або нейтральні. Це корисно для брендів та компаній, які аналізують відгуки користувачів про продукцію або послуги [2].

5) *Автоматичний переклад текстів.* BERT допомагає покращувати автоматичний переклад текстів між різними мовами, враховуючи складний контекст фраз і речень. Це дозволяє досягати більш точного перекладу з меншою кількістю помилок, особливо у випадках, коли необхідно розуміти значення фраз у ширшому контексті [16].

6) *Виявлення токсичності в коментарях та контенті.* BERT використовується для автоматичного фільтрування токсичних коментарів та образливого контенту в соціальних мережах та форумах. Модель аналізує коментарі та визначає рівень токсичності, що дозволяє видаляти або маркувати образливий контент [2].

7) *Перевірка правопису та граматики (Grammarly, Microsoft Word).* BERT може використовуватися в таких інструментах, як Grammarly, для покращення перевірки граматики та стилістики тексту. Модель допомагає знаходити помилки у тексті, виправляти структуру речень і пропонувати покращення з огляду на контекст [10].

8) *Класифікація електронної пошти (спам/не спам).* BERT допомагає класифікувати електронні листи як спам або справжні повідомлення, ґрунтуючись на семантиці тексту, заголовках та вмісті листів. Це покращує системи фільтрації спаму, забезпечуючи захист користувачів від небажаних та шкідливих повідомлень. [6].

9) *Пошук подібних документів.* BERT використовується для пошуку документів або статей, які схожі за змістом на заданий текст. Це може бути корисним у наукових дослідженнях, коли потрібно знайти матеріали, пов'язані з конкретною темою, або для автоматизації юридичного пошуку [7].

10) *Виявлення дублікатів контенту.* У випадках, коли потрібно виявити дублікат контенту (наприклад, у новинах або соціальних мережах), BERT допомагає знайти ідентичні або схожі за змістом тексти, навіть якщо вони перефразовані. Це корисно для боротьби з плагіатом або перевірки унікальності інформації [8].

GRU

GRU (Gated Recurrent Unit) — це один з типів рекуррентних нейронних мереж (RNN), розроблений для роботи з послідовними даними. Він був запропонований у 2014 році і є спрощеною версією LSTM (Long Short-Term Memory). Основною метою GRU є поліпшення навчання та обробки інформації в послідовностях, таких як текст, аудіо та часорядні дані.

Основні особливості GRU:

1) *Структура.* GRU має два основних гейта:

Оновлення (Update Gate): контролює, яку інформацію з попереднього стану потрібно зберегти та яку нову інформацію додати.

Скидання (Reset Gate): визначає, яку частину попередньої інформації можна забути.

2) *Відсутність комірки пам'яті.* На відміну від LSTM, GRU не має окремої комірки пам'яті, що спрощує структуру моделі.

3) *Швидкість навчання.* Завдяки спрощеній архітектурі GRU, він зазвичай навчається швидше, ніж LSTM, при порівнянних умовах.

4) *Ефективність.* GRU показує конкурентоспроможну продуктивність в задачах, пов'язаних з обробкою тексту, і часто використовується для задач, пов'язаних з аналізом природної мови, таких як класифікація тексту та аналіз настроїв.

GRU застосовується в основному для розв'язання наступних завдань:

1) *Аналіз фейкових новин.* GRU може бути використано для виявлення фейкових новин, аналізуючи текст новин на основі їх контексту та стилістичних особливостей. Модель може виявляти аномалії в структурі речень або вживанні емоційних слів [17].

2) *Класифікація тексту.* GRU може автоматично класифікувати документи або новини на категорії "справжні" та "фейкові", використовуючи попередньо навчений модельний підхід на мітках даних [18].

3) *Аналіз настроїв.* GRU можна використовувати для аналізу настроїв у соціальних мережах або коментарях, де модель оцінює емоції, виражені в текстах, і виявляє, чи є вони позитивними, негативними чи нейтральними.

4) *Системи запитань і відповідей.* GRU може бути застосовано в системах Q&A, де модель аналізує запитання і генерує відповіді на основі контексту, що дозволяє швидко знаходити релевантну інформацію.

Основними переваги GRU є:

1) *Спрощена архітектура.* Менше параметрів порівняно з LSTM, що може призвести до швидшого навчання.

2) *Ефективність у обробці послідовностей.* GRU добре справляється з тривалими послідовностями, завдяки своїм гейтам.

3) *Гнучкість.* GRU може бути адаптовано до різних типів задач в обробці природної мови.

Недоліки GRU є:

1) *Обмеження в контекстуальній пам'яті.* Хоча GRU ефективно справляється з більшістю задач, у деяких випадках LSTM може продемонструвати кращі результати в обробці дуже довгих послідовностей, де критично важливо зберігати пам'ять про старі входи.

2) Не завжди оптимальне рішення.

3) *Обробка контексту.* Трансформери використовують механізм уваги, який дозволяє моделі фокусуватися на різних частинах вхідних даних

одночасно. Це робить їх особливо ефективними для задач, де важливі віддалені залежності в даних.

4) *Паралелізація.* Трансформери можуть обробляти всю послідовність даних паралельно, що значно прискорює навчання на великих наборах даних. У GRU та LSTM, де дані обробляються послідовно, це може стати вузьким місцем.

5) *Гнучкість.* Трансформери можуть легко масштабуватися до великих моделей, що дає їм можливість досягати високої продуктивності на різних задачах, включаючи обробку природної мови та генерацію тексту.

RoBERTa

RoBERTa (A Robustly Optimized BERT Pretraining Approach) — це вдосконала версія моделі BERT, створена дослідниками компанії Facebook AI у 2019 році. RoBERTa вирішує деякі обмеження оригінальної BERT і пропонує ряд оптимізацій, які роблять модель потужнішою і ефективнішою для багатьох завдань обробки природної мови (NLP).

Основні особливості RoBERTa наступні:

1) *Оптимізація процесу попереднього навчання.* Основна відмінність RoBERTa від BERT полягає в тому, що у RoBERTa покращено процес попереднього навчання. Дослідники виявили, що оригінальна BERT не використовувала весь потенціал своєї архітектури, і це можна було покращити за допомогою тривалішого навчання на більшому обсязі даних.

2) *Відмова від завдання прогнозування наступного речення (Next Sentence Prediction, NSP).* BERT використовував завдання NSP для поліпшення розуміння зв'язку між реченнями. Однак у RoBERTa це завдання було видалено, оскільки дослідники виявили, що воно не суттєво впливає на продуктивність моделі у реальних NLP-задачах. Замість цього всі ресурси було спрямовано на масковане моделювання мови (Masked Language Modeling, MLM), що покращило загальну точність.

3) *Збільшення обсягів навчання та використання більших наборів даних.* RoBERTa було навчено на набагато більшому обсязі текстових даних порівняно з BERT. Якщо BERT навчався на наборі даних обсягом близько 16 ГБ (Вікіпедія + BookCorpus), то RoBERTa використовував до 160 ГБ даних із таких джерел, як Common Crawl, новинні статті, книги та інші відкриті джерела. Це дало можливість моделі краще "зрозуміти" природну мову.

4) *Динамічне маскування.* У RoBERTa застосовується динамічне маскування токенів на кожній ітерації навчання, що дозволяє моделі ефективніше навчатися на більш різноманітних текстових даних. У BERT масковані

слова були вибрані лише один раз для всього навчання, що могло обмежувати варіативність прикладів.

5) *Збільшення розміру міні-пакетів (mini-batches) і числа ітерацій.* Ще одним удосконаленням було збільшення розміру міні-пакетів (batch size) та кількості ітерацій у процесі тренування. Це дозволило RoBERTa краще узагальнювати знання та збільшити точність при вирішенні багатьох задач, таких як класифікація тексту, питання-відповіді та розпізнавання іменованих сутностей.

6) *Універсальність архітектури.* Подібно до BERT, RoBERTa також є трансформерною моделлю, що дозволяє їй працювати з довгими текстами, ефективно використовувати механізм самоуваги (self-attention) і враховувати складні залежності між словами. Проте завдяки покращенням у навчанні RoBERTa показує ще кращі результати на багатьох NLP-задачах [2].

Структура моделі представлена на Рис. 2.

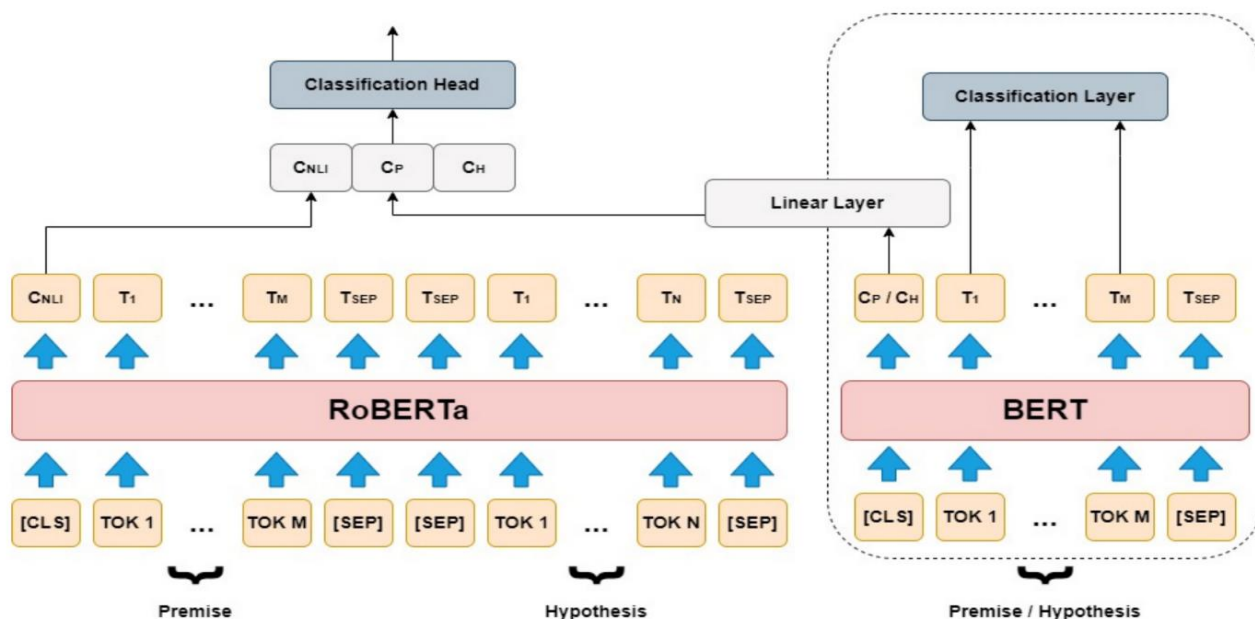


Рис. 2. Структура RoBERTa

До основних переваг RoBERTa можна віднести наступні:

1) *Підвищена продуктивність.* Завдяки вдосконаленому процесу навчання RoBERTa перевершує оригінальну BERT на багатьох популярних тестових наборах даних, таких як GLUE, RACE та SQuAD. Це робить RoBERTa кращим вибором для завдань, де необхідна точність і глибокий аналіз тексту.

2) *Відсутність завдання NSP.* Виключення завдання прогнозування наступного речення (NSP) спростило модель і дозволило повністю зосередитися на покращенні якості розуміння тексту через завдання MLM. Це знизило потребу в ресурсах та зробило модель більш ефективною.

3) *Масштабованість.* Модель RoBERTa демонструє відмінні результати при масштабуванні на більші набори даних і триваліші процеси навчання. Вона показала, що більша кількість даних і обчислювальних потужностей може значно покращити якість моделей глибокого навчання в NLP.

4) *Гнучкість і адаптивність.* Як і BERT, RoBERTa легко адаптується до різних задач через fine-tuning (доучування) на спеціалізованих наборах даних. Це дозволяє застосовувати модель у різних доменах, включаючи виявлення фейкових новин, аналіз настроїв, розпізнавання іменованих сутностей та інші задачі [2].

Основними недоліками є:

1) *Високі обчислювальні ресурси.* Як і в випадку з BERT, RoBERTa потребує значних обчислювальних потужностей для тренування і використання. Через збільшення обсягів даних і розмірів міні-пакетів, RoBERTa вимагає ще більше ресурсів, ніж оригінальна модель.

2) *Великий розмір моделі.* Модель RoBERTa має великий розмір (через кількість параметрів і обсяги даних для навчання), що ускладнює її використання на пристроях із обмеженими ресурсами або для задач, де необхідна низька затримка. Для реальних додатків може бути необхідно оптимізувати модель або застосовувати спеціальні техніки для її стиснення.

3) *Відсутність інтерпретованості*. Як і BERT, RoBERTa є "чорною скринькою", що робить її важкою для інтерпретації. Це означає, що навіть при високій точності результатів важко зрозуміти, які саме фактори вплинули на прийняте рішення. Це може бути проблемою для застосувань, де важлива прозорість результатів, наприклад, в юридичній чи медичній сфері.

4) *Велика вартість навчання*. Враховуючи необхідність у великих обсягах даних і обчислювальних ресурсах, тренування RoBERTa є дорогим процесом. Це робить модель менш доступною для невеликих компаній або окремих дослідників без доступу до потужних серверів або хмарних обчислень [2].

RoBERTa використовується ефективно для розв'язання наступних завдань:

1) *Аналіз фейкових новин*. RoBERTa використовується для аналізу текстів новин, щоб виявляти дезінформацію та фейкові новини. Завдяки кращій здатності аналізувати контекст і використовувати великі набори даних, вона ефективніше виявляє маніпуляції у новинних матеріалах [9].

2) *Класифікація тексту*. RoBERTa можна застосовувати для класифікації тексту в різних доменах, включаючи медицину, право, фінанси. Вона може бути використана для автоматизації аналізу документів та покращення пошукових систем [9].

3) *Аналіз настроїв і емоцій*. RoBERTa використовується для аналізу настроїв у текстах, наприклад, у відгуках клієнтів, коментарях у соціальних мережах чи новинних статтях. Це допомагає компаніям та організаціям краще розуміти настрої аудиторії [11].

4) *Системи запитання-відповіді (Q&A)*. RoBERTa є однією з найкращих моделей для систем автоматизованих запитань-відповідей. Завдяки покращеній обробці тексту, модель може ефективно знаходити відповідь на конкретне запитання в рамках великого тексту або документа [9].

XLNET

XLNet — це трансформерна модель, яка поєднує в собі особливості ауторегресійних і аутокодерних підходів. Вона використовує механізм пермутації, що дозволяє моделі враховувати різні порядки слів під час навчання.

XLNet має два основні елементи:

1) *Ауторегресійний підхід* — модель передбачає ймовірність наступного слова в реченні, базуючись на всіх попередніх словах, без маскування.

2) *Пермутаційний механізм* - у процесі навчання XLNet розглядає всі можливі порядки слів у вхідному тексті, що допомагає моделі запам'ятовувати контекстуальні залежності [3].

Структура моделі XLNet представлена на Рис. 3.

Основними перевагами моделі є наступні:

1) *Вища точність*. XLNet демонструє покращені результати на багатьох завданнях обробки природної мови, таких як питання-відповіді, аналіз тексту та інші, досягаючи високих балів на наборах даних SQuAD і GLUE.

2) *Краще розуміння контексту*. Завдяки механізму пермутації, XLNet має можливість враховувати глобальні залежності між словами, що покращує розуміння контексту в тексті.

3) *Гнучкість*. XLNet може бути використана для різних завдань без необхідності значного перенавчання, що робить її універсальним інструментом для досліджень у сфері обробки природної мови.

4) *Відсутність маскування*. У порівнянні з BERT, XLNet не маскує слова, що зменшує ризик втрати інформації [3].

Основними недоліками моделі є:

1) *Складність навчання*. XLNet може бути складнішою в навчанні порівняно з іншими моделями, такими як BERT, через використання механізму пермутації та ауторегресійної природи.

2) *Вимоги до ресурсів*. Модель вимагає більше обчислювальних ресурсів і часу для навчання через свою складність, що може бути недоступним для деяких користувачів.

Складність налаштування. Залучення до специфічних завдань може вимагати більш глибокого розуміння механізму моделі та її параметрів [3].

XLNet є потужним інструментом для обробки природної мови, який пропонує ряд переваг у порівнянні з попередніми моделями. Хоча вона має деякі недоліки, пов'язані з навчанням і ресурсами, її точність і гнучкість роблять її цінним активом у багатьох сферах, пов'язаних із штучним інтелектом та обробкою текстів [3].

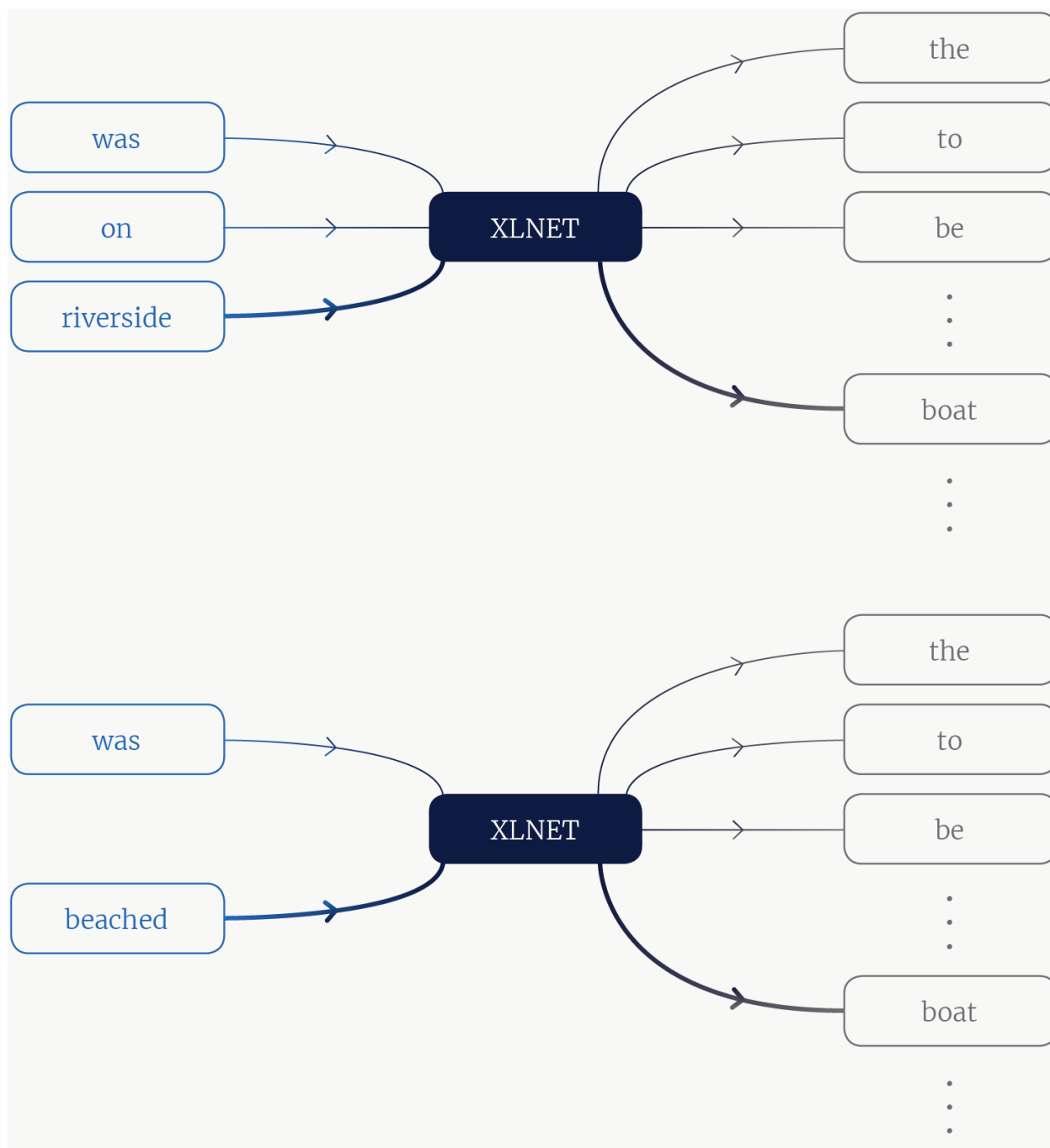


Рис.3. Структура моделі XLNet

LSTM

LSTM (Long Short-Term Memory) — це тип рекуррентної нейронної мережі (RNN), розроблений для подолання проблеми забуття інформації в традиційних RNN, зокрема для збереження та передачі інформації протягом тривалих послідовностей. Ця архітектура особливо корисна в задачах, пов'язаних із аналізом тексту, включаючи виявлення фейкових новин.

LSTM має спеціальну архітектуру, що включає три основні компоненти (входи, виходи та стан).

Стан комірки (cell state) — основний елемент, що зберігає інформацію протягом тривалого часу. Цей стан може бути оновлений або збережений, дозволяючи моделі запам'ятовувати важливу інформацію.

Шлюзи (gates) — LSTM використовує три типи забиральників для контролю інформаційного потоку:

Вхідний шлюз (input gate) — визначає, які нові дані будуть збережені у стані комірки.

Шлюз забування (forget gate) – вирішує, які дані будуть викинуті з комірки, що дозволяє моделі позбутися непотрібної інформації.

Вихідний шлюз (output gate) – контролює, які дані з комірки будуть виведені на вихід.

Активація – зазвичай LSTM використовує функцію активації, таку як \tanh або сигмоїда, для обробки даних. [4].

Структура моделі представлена на Рис. 4.

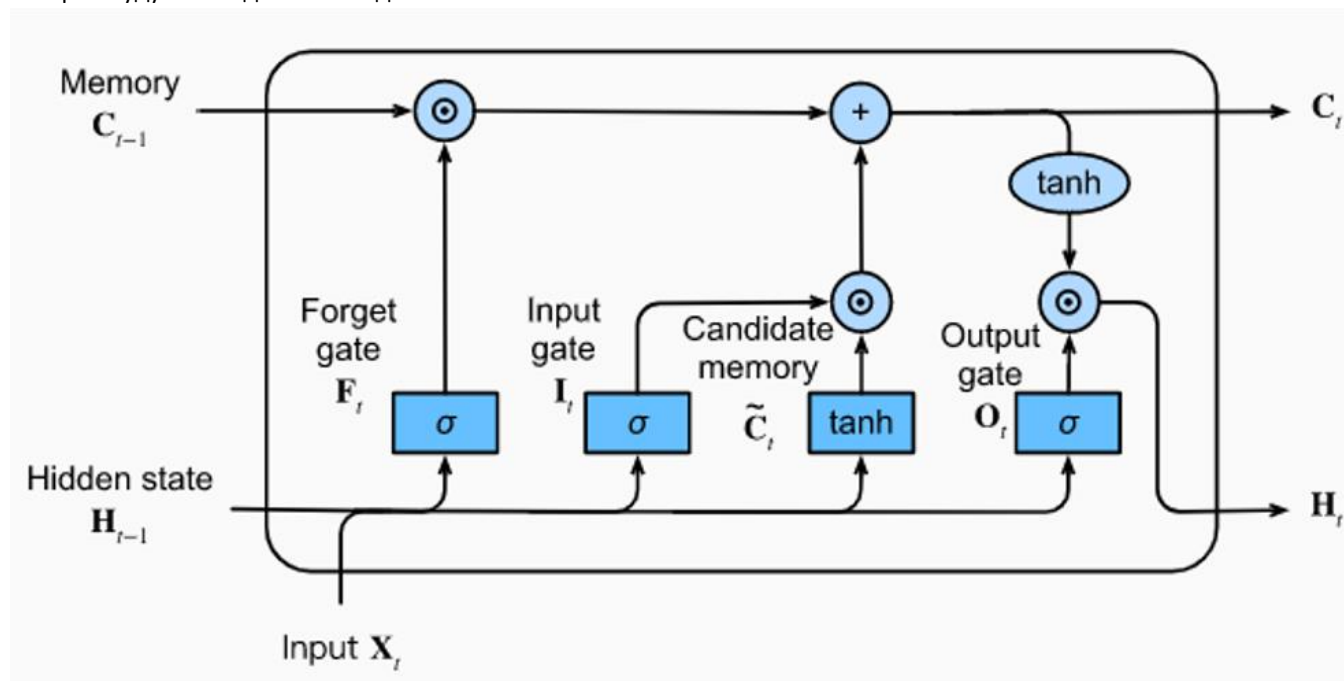


Рис. 4. Структура моделі LSTM

LSTM (Long Short-Term Memory) мережі стали однією з найефективніших архітектур у галузі обробки послідовних даних завдяки своїм унікальним властивостям. Ось деякі з основних переваг, які роблять LSTM надзвичайно потужними для різноманітних завдань:

1) *Пам'ять на тривалій дистанції.* Однією з найбільш важливих характеристик LSTM є їх здатність зберігати інформацію на тривалі періоди часу. Це дозволяє моделі враховувати контекст, який може бути віддаленим від ключових слів у тексті або елементах послідовності. Завдяки такій пам'яті, LSTM можуть ефективно аналізувати й генерувати текст, де важливість окремих слів або фраз може залежати від загального змісту, який був згаданий раніше.

2) *Управління забуттям.* LSTM використовують механізм шлюзів, які дозволяють моделі вибірково зберігати або відкидати інформацію. Це значно підвищує їхню продуктивність, особливо на складних послідовностях, де не вся інформація є релевантною для подальшого аналізу. Завдяки цій гнучкості, моделі можуть адаптуватися до динаміки даних і вчитися на основі нових інформаційних патернів.

3) *Гнучкість.* LSTM володіють чудовою адаптивністю до різних типів даних і завдань. Вони

можуть бути успішно застосовані в багатьох сферах, включаючи обробку тексту, зображень і аудіо. Завдяки своїй архітектурі, LSTM можуть легко налаштуватися під конкретні вимоги завдань, що робить їх універсальним інструментом для машинного навчання.

4) *Стійкість до зникаючого градієнта.* Одна з основних проблем традиційних рекурентних нейронних мереж (RNN) полягає в зникаючому градієнті, що ускладнює навчання на довгих послідовностях. LSTM значно покращують цю ситуацію, дозволяючи моделі ефективно навчатися на великих обсягах даних. Це забезпечує стабільність і ефективність навчання, навіть коли модель працює з тривалими контекстами.

Недоліками моделі є наступні:

1) *Складність.* Архітектура LSTM є складнішою в порівнянні з традиційними рекурентними нейронними мережами (RNN). Це пов'язано з тим, що LSTM містить кілька компонентів, таких як забиральники, стан комірки та механізми активації. Ця складна структура ускладнює розуміння та налаштування моделі, особливо для тих, хто не має глибокого досвіду в машинному навчанні. Програмістам потрібно добре знати, як ці компоненти взаємодіють один з одним, щоб ефективно налаштувати модель для досягнення найкращих результатів. Ця

складність може призвести до необхідності більше часу на налаштування гіперпараметрів та вибір архітектури.

2) *Час навчання.* Через свою складну архітектуру та великий обсяг параметрів, LSTM зазвичай потребують більше часу для навчання в порівнянні з простішими моделями, такими як звичайні RNN або навіть деякі моделі на основі механізмів уваги. Процес навчання включає в себе обчислення градієнтів для великої кількості параметрів, що може бути дуже ресурсоємним. Це особливо актуально, коли мова йде про великі набори даних, оскільки навчання може зайняти години або навіть дні, вимагаючи значних обчислювальних потужностей, таких як GPU або TPU. Крім того, затримка в навчанні може вплинути на швидкість розгортання моделі в продуктивних середовищах, де швидкість реагування є критично важливою.

3) *Вразливість до перенавчання.* Як і багато інших складних моделей, LSTM можуть бути схильні до перенавчання, особливо коли навчаються на малих обсягах даних. Перенавчання відбувається, коли модель вивчає не тільки основні патерни, а й шум та випадкові флуктуації в навчальних даних. Це може призвести до поганої генералізації, коли модель демонструє відмінні результати на навчальних даних, але погано справляється з новими, невідомими даними. Щоб зменшити ризик перенавчання, можуть знадобитися додаткові техніки, такі як регуляризація, дроп-аут або збільшення обсягу навчальних даних, що в свою чергу може ще більше ускладнити процес навчання і вимагати додаткових ресурсів.

Ці недоліки, хоча й істотні, часто компенсуються перевагами, які LSTM пропонують в задачах, що вимагають збереження та обробки контексту в довгострокових послідовностях [4].

Модель активно використовується для виявлення фейкових новин та для окремих задач, таких як:

1) *Аналіз тексту.* Моделі LSTM здатні здійснювати глибокий аналіз тексту новин, виявляючи складні патерни, стилістичні риси та структури, які можуть свідчити про їхню достовірність. Вони можуть виявляти використання емоційної мови, що є поширеним прийомом у фейкових новинах, оскільки емоційні слова часто спрямовані на маніпуляцію читачем. Наприклад, заголовки, що містять сильні емоції або відчуття терміновості, можуть бути індикаторами ненадійності. LSTM також можуть виявляти певні конструкції речень або повторювані фрази, які характерні для фейкових новин, такі як надмірне використання суперлативів чи узагальнень. Крім того, аналіз може включати в себе оцінку граматичних структур, щоб виявити типові

помилки, які можуть свідчити про низьку якість джерела [12].

2) *Виявлення контексту.* LSTM мають здатність зберігати контекст повідомлень протягом тривалих послідовностей, що дозволяє їм глибше аналізувати текст. Ця здатність до зберігання інформації в контексті важлива для виявлення маніпуляцій або непослідовностей в подачі інформації. Наприклад, якщо новина спочатку стверджує одне, а потім протирічить цій інформації, LSTM можуть вловити ці зміни і вказати на ймовірну ненадійність або маніпуляцію. Таким чином, завдяки своїй архітектурі, LSTM можуть відстежувати, як інформація змінюється в межах тексту, що дозволяє виявляти потенційні фейкові новини або дезінформацію [12].

3) *Порівняння з перевіреними фактами.* Для підвищення ефективності LSTM можна інтегрувати з базами даних перевірених фактів. Це означає, що модель не лише аналізує текст новини, але й порівнює інформацію з уже відомими, перевіреними фактами. Наприклад, якщо новина стосується конкретних подій або заяв, LSTM може перевірити, чи ці факти підтверджені незалежними джерелами. Таке порівняння може включати перевірку даних про місця, дати, цитати та інші елементи, що допомагає зменшити кількість неправдивої інформації, яку поширюють фейкові новини [12].

4) *Класифікація новин.* Моделі LSTM можуть бути використані для автоматичної класифікації новин на категорії "справжні" та "фейкові". Для цього моделі тренуються на великих наборах даних, де новини вже мають мітки, що вказують на їхню правдивість. Під час навчання LSTM вчаться виявляти особливості, що відрізняють фейкові новини від реальних, враховуючи такі аспекти, як тон, структура, використання джерел і наявність фактів. Після навчання модель може аналізувати нові статті і автоматично надавати їм оцінку достовірності, що значно спрощує процес виявлення фейкових новин і зменшує навантаження на журналістів та аналітиків [12].

Підсумовуючи вищенаведене, LSTM є потужним інструментом для аналізу та обробки послідовних даних, таких як текст. Його здатність зберігати та обробляти тривалі контексти робить його особливо корисним для завдань виявлення фейкових новин, де важливо аналізувати не лише окремі слова, а й загальний зміст та структуру повідомлення. Хоча LSTM має свої недоліки, його переваги роблять його важливим інструментом у боротьбі з дезінформацією в медіа.

HAN

HAN (Hierarchical Attention Networks) — це тип нейронної мережі, спеціально розроблений для обробки тексту, який має ієрархічну структуру. HAN був вперше представлений у статті "Hierarchical Attention Networks for Document Classification" авторів Yang et al. у 2016 році. Його основна мета полягає в тому, щоб ефективно аналізувати великі обсяги тексту, зокрема документи, що складаються з абзаців та речень, використовуючи механізми уваги.

HAN має дві основні ієрархічні рівні такі як:

1) *Рівень речень*, де мережа обробляє кожне речення в документі. Вона використовує механізм уваги,

щоб визначити, які слова в реченні є важливими для розуміння його змісту. Результати обробки всіх речень потім агрегуються для подальшої обробки.

2) *Рівень документу*, де після обробки речень, результати з кожного речення передаються на верхній рівень, де мережа знову застосовує механізм уваги. Це дозволяє моделі визначити, які речення в документі є найбільш важливими для загального розуміння тексту.

Ця ієрархічна структура дозволяє моделі краще справлятися з контекстом, оскільки вона враховує зв'язки не тільки між словами в одному реченні, але й між різними реченнями в документі [5].

Структура моделі представлена на Рис. 5.

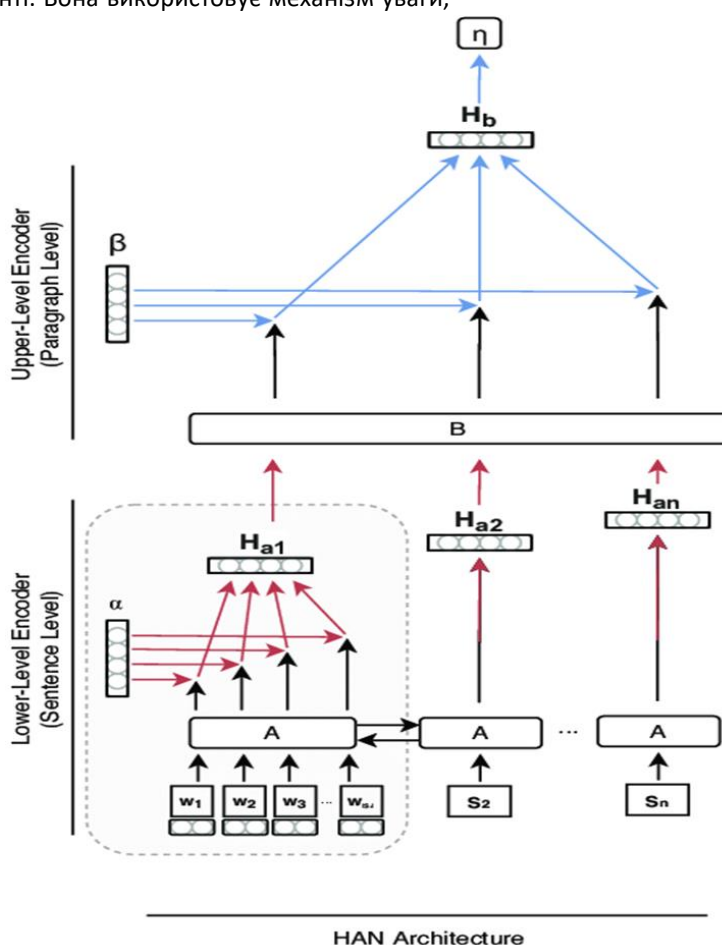


Рис. 5. Структура моделі HAN

Модель має наступні переваги:

1) *Контекстуальна обробка.* HAN може ефективно обробляти текст, беручи до уваги контекст на різних рівнях, зокрема на рівні речення та документа. Це особливо важливо в задачах, де контекст має значення для розуміння змісту, таких як аналіз складних новин або наукових статей. Завдяки цьому підходу модель може краще справлятися з різноманітними нюансами мови, враховуючи семантичні зв'язки між словами. Наприклад,

у великих текстах, де одне речення може спиратися на інформацію, надану в попередніх реченнях, HAN може відстежувати ці зв'язки, забезпечуючи більш точне розуміння.

2) *Механізм уваги.* Використання механізму уваги є одним із найсильніших аспектів HAN. Цей механізм дозволяє моделі фокусуватися на важливих частинах тексту, ігноруючи зайву інформацію, що підвищує точність результатів. Коли HAN обробляє текст, вона може

виділити ключові слова або фрази, які є критично важливими для аналізу, такі як специфічні факти, терміни чи емоційні вирази. Це робить модель більш гнучкою та ефективною в обробці різних типів текстів, оскільки вона може адаптуватися до особливостей даних, з якими працює.

3) *Ієрархічна структура*. Ієрархічний підхід, притаманний HAN, дозволяє ефективно обробляти великі документи, що складаються з багатьох речень. Ця структура забезпечує збереження важливої інформації на всіх рівнях обробки, від окремих слів до цілих абзаців. Таким чином, модель може зрозуміти загальний зміст документа, не втрачаючи деталей, які можуть бути суттєвими для його оцінки. Наприклад, при аналізі новинних статей HAN може враховувати як конкретні факти, так і загальний наратив, що допомагає виявляти маніпуляції або неузгодженості.

4) *Гнучкість*. HAN можна легко налаштувати для різних завдань, що робить його універсальним інструментом в аналізі тексту. Модель може бути використана не лише для класифікації тексту (як справжнього, так і фейкового), а й для аналізу настроїв, виділення ключових фраз, побудови рекомендаційних систем або навіть у чат-ботах для обробки запитів користувачів. Завдяки цій гнучкості, HAN може адаптуватися до різних доменів і типів текстів, що робить його корисним у багатьох сферах, від журналістики до маркетингу та соціальних наук [5].

Серед основних недоліків моделі наступні:

1) *Складність моделі*. Як і багато інших моделей на основі глибокого навчання, HAN має складну архітектуру, що вимагає значних обчислювальних ресурсів та часу для навчання. Це означає, що для успішного використання HAN необхідно мати доступ до потужних комп'ютерних систем або кластерів, що підтримують графічні процесори (GPU) або інші обчислювальні платформи. Ця складність може створити бар'єри для впровадження технології в менш забезпечених середовищах, таких як невеликі компанії або академічні установи, де доступ до ресурсів може бути обмежений.

2) *Час навчання*. Через ієрархічну структуру та механізми уваги навчання HAN може займати більше часу порівняно з простішими моделями. Процес навчання включає обробку багатьох рівнів інформації, що збільшує обчислювальні витрати. Це може стати проблемою, якщо потрібно швидко отримати результати або адаптувати модель до нових даних. Наприклад, у випадках, коли швидкість реакції є критично важливою, таких як виявлення фейкових новин у реальному часі, тривалий час навчання може стати серйозним обмеженням.

3) *Необхідність великих обсягів даних*. Для досягнення хороших результатів HAN потребує великих обсягів навчальних даних, що може бути проблемою, якщо доступні дані обмежені. В умовах, коли дані є рідкісними або їх важко зібрати, модель може не мати достатньої інформації для навчання, що може призвести до зниження її ефективності. У таких випадках результати можуть бути ненадійними, і модель може не виявляти ключові патерни, що є критичними для успішного аналізу тексту.

4) *Вразливість до перенавчання*. Як і інші складні моделі, HAN може бути схильна до перенавчання, особливо при недостатньому обсязі даних. Перенавчання відбувається, коли модель починає запам'ятовувати деталі навчальних даних замість того, щоб виявляти загальні закономірності. Це може призвести до погіршення її продуктивності на нових, невідомих даних. У разі роботи з обмеженими наборами даних або надто складними моделями важливо впроваджувати стратегії, такі як регуляризація або крос-валідація, щоб зменшити ризик перенавчання [5].

HAN активно використовується для розв'язання наступних задач:

1) *Класифікація новин*. HAN може використовуватися для класифікації новин як справжніх чи фейкових. Модель може аналізувати текст новин, щоб визначити, які частини тексту є найбільш важливими для оцінки достовірності, а також порівнювати стилістичні риси з відомими ознаками фейкових новин [13].

2) *Аналіз стилю та змісту*. Завдяки своїй здатності обробляти тексти на різних рівнях, HAN може виявляти специфічні патерни стилю, характерні для фейкових новин, такі як використання емоційної мови, перебільшення або неточності в фактах [13].

3) *Контекстуальне виявлення маніпуляцій*. HAN може зберігати контекст у великих текстах, що дозволяє виявляти маніпуляції або непослідовності в подачі інформації, які можуть свідчити про фейкові новини [13].

4) *Взаємодія з перевіреними фактами*. HAN можна інтегрувати з системами перевірки фактів для аналізу новин і перевірки їх на відповідність відомим даним, що підвищує точність оцінки достовірності новин [14].

ESIM

ESIM (Enhanced Sequential Inference Model) — це модель для обробки природної мови, розроблена для задач, пов'язаних із семантичним текстовим порівнянням, таких як відповідь на питання, класифікація тексту і виявлення фейкових новин. ESIM була представлена в 2018 році в статті "Enhanced Sequential

Inference Model for Natural Language Inference" авторів Chen et al. Модель покращує традиційні підходи до аналізу тексту, інтегруючи механізми уваги та контекстуального розуміння, та використовує наступні елементи:

1) *Векторизація слів*. На початковому етапі ESIM перетворює слова у вектори за допомогою попередньо навчених векторів, таких як GloVe або Word2Vec. Ці вектори дозволяють моделі захоплювати семантичну інформацію про слова, їх значення та зв'язки з іншими словами. Наприклад, вектори можуть вловлювати подібності між словами, такими як "корова" і "молоко", які мають спільні тематичні асоціації. Цей процес дозволяє моделі не лише розуміти індивідуальні слова, але й враховувати їхні контекстуальні значення у реченнях.

2) *Паралельне кодування*. ESIM кодує два вхідні текстові сегменти (наприклад, питання та відповідь або два речення) паралельно, використовуючи однакову архітектуру. Це підхід дозволяє моделі ефективно виявляти зв'язки та відмінності між сегментами, а також зрозуміти, як інформація з одного тексту може впливати на інтерпретацію іншого. Паралельне кодування підвищує ефективність аналізу та дозволяє моделі формувати глибше розуміння взаємодії між текстами.

3) *Механізм уваги*. Механізм уваги є ключовим елементом ESIM, оскільки він дозволяє моделі зосередитися на важливих частинах тексту, які є найбільш релевантними для завдання. Наприклад, у процесі порівняння речень модель може виділити конкретні слова або фрази, які містять важливу інформацію або емоційний підтекст. Це підвищує точність порівняння між вхідними сегментами, оскільки модель може ігнорувати непотрібну або другорядну інформацію, зосереджуючи свою увагу на суті.

4) *Інференційний блок*. Інференційний блок ESIM виконує глибокий аналіз взаємодій між двома текстами, використовуючи обчислені контекстуальні вектори. Це дозволяє моделі формувати вгадування про те, як один текст відноситься до іншого. Наприклад, при аналізі новин модель може оцінювати, чи підтверджує один текст інформацію з іншого, чи є він суперечливим. Цей блок сприяє більш точному виявленню семантичних відносин і патернів, що виникають у текстах.

5) *Фінальна класифікація*. На завершальному етапі ESIM об'єднує всю отриману інформацію та проводить класифікацію, щоб визначити семантичні відносини між текстами. Цей процес включає в себе оцінку всіх аспектів, виявлених на попередніх етапах, і прийняття рішення про те, чи є текст справжнім, фейковим чи суперечливим. Завдяки інтеграції інформації з попередніх етапів, модель

може забезпечити більш точні результати, ніж традиційні методи, які не враховують комплексні взаємодії між текстами [6].

До основних переваг моделі можна віднести:

1) *Контекстуальне розуміння*. ESIM здатна захоплювати контекстуальні залежності між словами та фразами, що робить її ефективною для складних завдань, таких як виявлення нюансів у значеннях.

2) *Механізм уваги*. Завдяки механізму уваги модель може виділяти ключові частини тексту, що дозволяє зосереджуватися на важливих аспектах інформації, покращуючи результати аналізу.

3) *Гнучкість*. ESIM можна налаштувати для різних завдань обробки природної мови, включаючи класифікацію, порівняння тексту та відповідь на питання, що робить її універсальним інструментом.

4) *Висока точність*. Модель показує конкурентоспроможні результати в багатьох задачах, пов'язаних із семантичним текстовим порівнянням, завдяки своїй здатності до контекстуального аналізу та механізму уваги [7].

Основними недоліками моделі є:

1) *Складність моделі*. Як і багато інших моделей глибокого навчання, ESIM має складну архітектуру, яка включає кілька рівнів та компонентів, що взаємодіють один з одним. Ця складність може призвести до високих вимог до обчислювальних ресурсів. Наприклад, для ефективного навчання моделі можуть знадобитися потужні графічні процесори (GPU) або спеціалізовані апаратні рішення, що збільшує загальні витрати на реалізацію. Ця складність також може ускладнити процес налагодження моделі, адже проблеми можуть виникати на різних етапах, і виявлення джерел помилок може вимагати додаткових зусиль.

2) *Час навчання*. Модель ESIM, завдяки своїй архітектурі та кількості параметрів, може потребувати значного часу для навчання. Це особливо важливо в контексті сучасних додатків, де швидкість реакції є критично важливою. Наприклад, у випадках виявлення фейкових новин у реальному часі, затримки в навчанні можуть призвести до втрати актуальності або своєчасності результатів. Коли модель тренується на великих обсягах даних, тривалість навчання може стати суттєвим обмеженням для розробників та користувачів.

3) *Вразливість до перенавчання*. Як і інші складні моделі, ESIM може бути схильна до перенавчання, особливо коли обсяг навчальних даних є недостатнім або неповним. Перенавчання відбувається, коли модель надто сильно адаптується до специфіки навчального набору даних, втрачаючи здатність узагальнювати на нових, невідомих даних. Це може призвести до суттєвого

зниження продуктивності моделі на реальних даних, що не входять до навчального набору. Щоб зменшити ризик перенавчання, необхідно впроваджувати різноманітні стратегії регуляризації та валідації, але це може ускладнити навчальний процес.

4) *Необхідність великих обсягів даних.* Для досягнення високих результатів модель ESIM вимагає великих обсягів навчальних даних. Це може стати серйозною проблемою в тих випадках, коли дані є рідкісними, складними для збору або

неструктурованими. Наприклад, у випадках виявлення фейкових новин, де необхідно мати доступ до великої кількості анотацій та перевірених фактів, недостатня кількість даних може призвести до обмеження ефективності моделі. Крім того, обмежений обсяг даних може ускладнити тренування моделі, адже вона може не отримати достатньої інформації для виявлення важливих патернів і тенденцій.

Структура моделі представлена на Рис. 6.

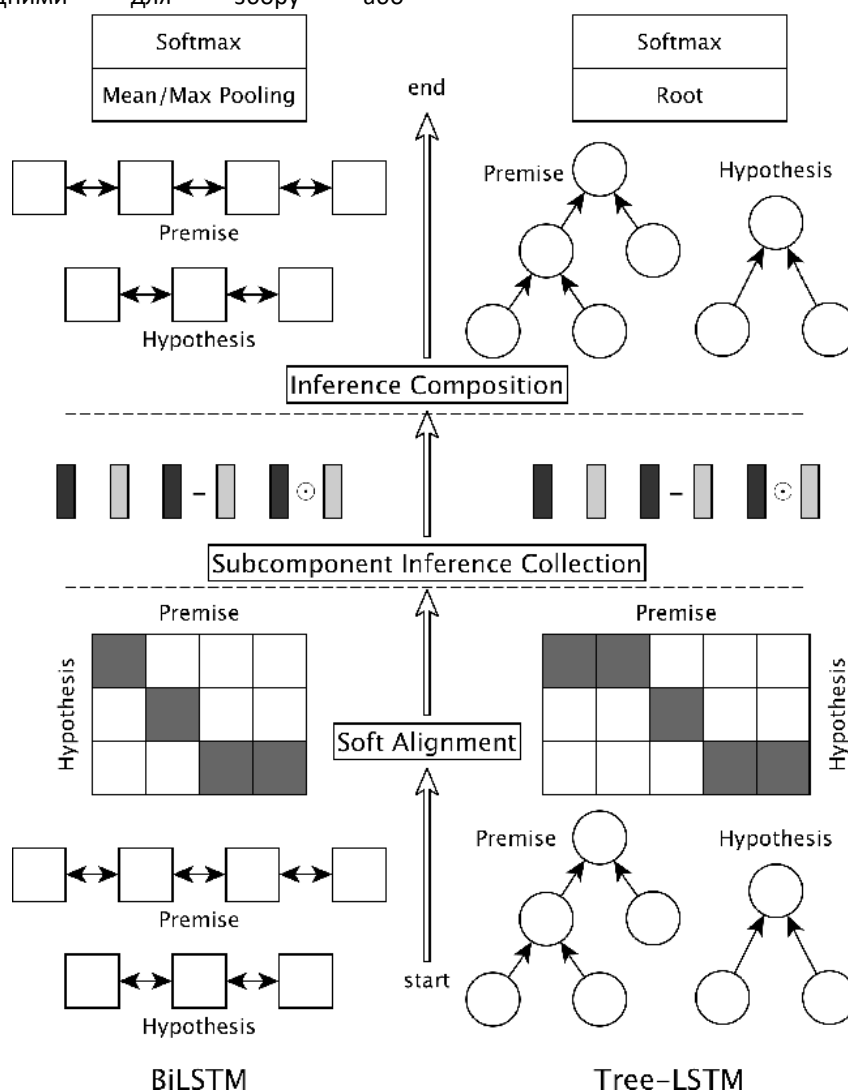


Рис.6. Структура моделі ESIM

Представлена модель використовується для розв'язання наступних завдань:

1) *Класифікація новин.* ESIM може бути ефективно використана для класифікації новин на справжні та фейкові. Завдяки своїй здатності аналізувати зв'язки між текстами новин та вже відомими фактами або авторитетними джерелами інформації, модель може виявляти дезінформацію на основі контексту та змісту. Наприклад, якщо новина містить твердження, що

суперечить науково обґрунтованим даним або заявам від авторитетних джерел, ESIM може ідентифікувати цю новину як потенційно фейкову. Завдяки механізму уваги, модель також може акцентуватися на ключових фразах і термінах, що часто вживаються в дезінформації, підвищуючи точність класифікації [15].

2) *Порівняння фактів.* Одна з ключових можливостей ESIM полягає в її здатності порівнювати тексти новин з перевіреними фактами. Це дозволяє

моделі виявляти дезінформацію або маніпуляції у поданні інформації. Наприклад, модель може аналізувати новини про певні події і звіряти інформацію з базами даних перевірених фактів, що містять об'єктивні дані, статистику або свідчення очевидців. Такий підхід допомагає забезпечити високий рівень точності виявлення неправдивих заяв, особливо в ситуаціях, коли новини можуть бути спотворені або подані в односторонньому контексті [15].

3) *Аналіз контексту.* ESIM також здатна виявляти маніпуляції в подачі інформації шляхом відстеження контекстуальних залежностей у текстах. Завдяки ієрархічній структурі моделі, вона може зберігати інформацію про зміст і контекст на різних рівнях, що дозволяє їй виявляти несуперечливості в подачі інформації. Наприклад, якщо новина містить різні твердження, які не відповідають одна одній або суперечать раніше заявленим фактам, ESIM може ідентифікувати ці розбіжності, підвищуючи вірогідність виявлення фейкових новин [15].

4) *Виявлення стилістичних рис.* Завдяки механізму уваги, ESIM може детально аналізувати стилістичні риси текстів, такі як використання емоційної мови, маніпулятивних прийомів або специфічних термінів, які часто вживаються в фейкових новинах. Наприклад, модель може ідентифікувати тексти, що містять надмірно емоційні чи провокаційні формулювання, які зазвичай використовуються для маніпуляцій з думкою читачів. Виявлення таких стилістичних ознак може суттєво допомогти в автоматизації процесу перевірки новин та підвищенні обізнаності споживачів інформації про потенційні дезінформаційні джерела [15].

УЗАГАЛЬНЕНЕ ПОРІВНЯННЯ МОДЕЛЕЙ

У Табл. 1 наведено узагальнені переваги та недоліки розглянутих моделей. При цьому окремо виділено оцінку бюджетування використання кожної з моделей.

Табл.1. Основні переваги та недоліки моделей

Модель	Переваги	Недоліки	Бюджет
BERT	- Висока точність для завдань NLU -Паралельне навчання	- Високі вимоги до ресурсів - Можливість перенавчання	Середній/Високий
RoBERTa	-Покращена точність у порівнянні з BERT - Гнучка архітектура	- Велика потреба в пам'яті - Час навчання	Високий
XLNet	-Краще розуміння контексту -Відсутність припущення про порядок слів	-Складність моделі - Час навчання	Високий
LSTM	-Сильні можливості для роботи з послідовними даними - Гнучкість	- Довший час навчання - Вразливість до перенавчання	Низький/Середній
HAN	- Ефективна обробка тексту - Виявлення важливих елементів	- Складна архітектура - Потреба в великих обсягах даних	Середній/Високий
ESIM	- Висока точність у порівнянні з простішими моделями - Здатність аналізувати текст	- Складність моделі - Час навчання	Середній/Високий

У відповідності з проведеним аналізом можна зробити певні рекомендації щодо вибору моделей для певного виду завдань.

Для завдань обробки природної мови (NLP), доцільно обрати моделі BERT або RoBERTa. Вони підходять для завдань, що потребують високої точності в розумінні мови, таких як текстова класифікація або відповіді на

запитання. RoBERTa може бути кращим вибором для задач, де необхідно підвищити точність порівняно з BERT.

Для обробки довгих послідовностей рекомендовано застосовувати модель LSTM, яка є ефективною при роботі з даними, що містять довгі послідовності, або для збереження контексту протягом тривалого часу. Це особливо актуально для аналізу тексту, де важлива

послідовність слів, наприклад, у задачах генерації тексту або машинного перекладу.

У випадках, коли завдання полягає у виявленні стилістичних особливостей тексту або важливих частин у великих документах, рекомендується використовувати HAN. Завдяки ієрархічній структурі та механізму уваги, ця модель забезпечує точний аналіз стилістичних рис.

Для порівняння зв'язків між різними текстами, наприклад, у задачах зіставлення новин або запитань-відповідей, доцільно використовувати модель ESIM, яка має інференційні блоки та механізм уваги, що забезпечують високу точність.

Для складних задач, що потребують глибокого розуміння контексту та структурованих відносин між

словами, доцільно застосовувати XLNet. Ця модель є ефективною для завдань, які вимагають аналізу не тільки змісту тексту, але й структури його побудови.

Якщо існують обмеження щодо обчислювальних ресурсів, модель LSTM може бути оптимальним вибором, оскільки її ресурсні вимоги є відносно низькими, хоча час навчання може бути тривалішим.

Для обробки великих обсягів даних моделі BERT, RoBERTa або XLNet забезпечують максимальну точність, проте вимагають значних обчислювальних ресурсів.

Модель HAN, завдяки своїй ієрархічній структурі, є гнучким інструментом, який дозволяє адаптуватися до різних типів текстів та умов завдань.

ВИСНОВКИ

Вибір моделей для обробки природної мови (NLP) є важливим фактором, який визначає ефективність та точність вирішення задач, пов'язаних з аналізом текстової інформації. Моделі, такі як BERT, RoBERTa, XLNet, LSTM, HAN і ESIM, мають різні характеристики, що дозволяє обирати оптимальний інструмент залежно від специфіки завдання. Наприклад, для задач із високими вимогами до точності, таких як аналіз настроїв або класифікація тексту, перевага надається BERT або RoBERTa, тоді як LSTM підходить для обробки послідовних даних.

Окрім точності, важливим аспектом є ресурсозатратність моделей. Складні моделі, такі як XLNet, можуть вимагати значних обчислювальних ресурсів, що варто враховувати при наявності обмеженого бюджету. У таких випадках більш прості моделі, як-от LSTM, можуть забезпечити хороший результат за менших витрат. Застосування моделей із механізмами уваги, таких як HAN і ESIM, показує високу ефективність при виявленні важливих частин тексту, що є особливо корисним для завдань, пов'язаних із виявленням фейкових новин або дезінформації.

Таким чином, правильний вибір моделі, що ґрунтується на аналізі конкретних потреб завдання і наявних ресурсів, дозволяє досягти оптимальних результатів. Зважаючи на швидкий розвиток галузі NLP, важливо постійно стежити за новими моделями та підходами для забезпечення найкращих результатів у практичних проектах. Проведений аналіз та надані в статті рекомендації надають можливість такий вибір зробити.

ЛІТЕРАТУРА

- [1] BERT Explained: State of the art language model for NLP. [Онлайн]. Available: <http://surl.li/xymxsn>. Дата звернення: 10.11.2023.
- [2] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". [Онлайн]. Available: <http://surl.li/vlpsnz>.
- [3] Google AI Blog, "Understanding Searches Better Than Ever Before". [Онлайн]. Available: <http://surl.li/vvhmqi>
- [4] Google Jigsaw, "Perspective API Using BERT". [Онлайн]. Available: <https://urlzs.com/vczAs>
- [5] W3School, "Machine Learning - Linear Regression". [Онлайн]. Available: <http://surl.li/rlbrd>. Дата звернення: 10.11.2023.
- [6] "Spam Classification with BERT," arXiv:2102.07004. [Online]. Available: <https://arxiv.org/abs/2102.07004>. Accessed: 10.11.2023
- [7] "Legal Document Search with BERT," arXiv:2105.09121. [Online]. Available: <https://arxiv.org/abs/2105.09121>. Accessed: 10.11.2023
- [8] ["Plagiarism Detection with BERT," arXiv:1910.00291. [Online]. Available: <https://arxiv.org/abs/1910.00291>. Accessed: 10.11.2023.
- [9] J. Leal, "Using RoBERTa for text classification," 2020. [Online]. Available: <https://urlzs.com/qafnF>. Accessed: 10.11.2023
- [10] Grammarly. "Grammar and AI Writing." [Онлайн]. URL: <https://urlzs.com/w9y66>. Дата звернення: 10.11.2023
- [11] "RoBERTa: A Robustly Optimized BERT Pretraining Approach." [Онлайн]. URL: <http://surl.li/xvxnbw>. Дата звернення: 10.11.2023
- [12] J. Leal. "A Novel Approach to Fake News Classification Using LSTM-Based Deep Learning Models." [Онлайн]. URL: <http://surl.li/quygw1>. Дата звернення: 10.11.2023.
- [13] R. Zhang, M. Zhang, Y. Zhang, "Hierarchical Co-Attention Selection Network for Interpretable Fake News Detection." [Онлайн]. URL: <http://surl.li/pdkggu>. Дата звернення: 10.11.2023
- [14] S. K. Lee, H. Jeong, "A Deep Neural Network for Fake News Detection." [Онлайн]. URL: <https://urlzs.com/BnPri>. Дата звернення: 10.11.2023.
- [15] A. Dasgupta, S. Agerri, "Natural Language Inference over Interaction Space." [Онлайн]. URL: <https://urlzs.com/FYcRg>. Дата звернення: 10.11.2023

- [16] S. Devlin, M. Chang, K. Lee, "BERT for Language Translation." [Онлайн]. URL: <http://surl.li/xthhii>. Дата звернення: 10.11.2023
- [17] P. Wu, Z. Liu, L. Han, "Fake News Detection via GRU-Based Ensemble Model in Social Media," IEEE Access, vol. 8, pp. 47103-47115, 2020. DOI: 10.1109/ACCESS.2020.2973912. (Q1)
- [18] E. E. Lazaridis, A. Drosos, "Fake News Classification Using GRU with Attention Mechanism: A Comparative Study," Information Processing & Management, vol. 58, no. 4, 2021, pp. 102-109. DOI: 10.1016/j.ipm.2021.102594. (Q2)

COMPARATIVE ANALYSIS OF MODELS FOR FAKE NEWS DETECTION AND CLASSIFICATION USING GRU

Vitalii Kovalenko, Iaroslav Dorohyi, Katerina Doroshenko

The article presents a comparative analysis of models for detecting and classifying fake news using GRU (gated recurrent unit), a modern neural network architecture that serves as an alternative to LSTM. The aim of the study is to evaluate the efficiency of the GRU model in comparison with other popular natural language processing (NLP) models, such as BERT, RoBERTa, and LSTM, in the context of identifying fake news. The relevance of the topic is driven by the need for accurate and timely detection of disinformation in today's information space, which significantly impacts societal processes and decision-making.

The research methodology is based on a comparative analysis using specific criteria. GRU, as a recurrent neural network, has a simpler architecture compared to LSTM, making it less resource-intensive while maintaining the ability to process long sequences of text. The main focus is on comparing the performance of GRU with other models in tasks related to fake news detection and classification, taking into account contextual processing capabilities.

The results of the comparative analysis show that GRU delivers competitive performance in terms of accuracy and training speed compared to LSTM and transformer-based models (BERT, RoBERTa), especially in resource-constrained environments. GRU proves effective when handling large volumes of text and analyzing complex contextual relationships. Due to its simpler architecture, GRU is a promising model for implementation in real-time fake news monitoring and detection systems.

The scientific novelty of the article lies in the exploration of GRU's effectiveness compared to other NLP models for text classification tasks, which can improve disinformation identification processes. The practical significance of the study is that the results can serve as recommendations for selecting a specific class of models to solve various tasks when developing systems for

combating fake news in different domains, including media, social networks, and analytical centers.

Keywords: *fake news detection, text classification, gated recurrent unit (GRU), deep learning, natural language processing (NLP), algorithm accuracy, computational optimization, news classification system, machine learning, text data analysis.*

REFERENCES

- [1] BERT Explained: State of the art language model for NLP. [Online]. Available: <http://surl.li/xymxsn>. Accessed: 10.11.2023.
- [2] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". [Online]. Available: <http://surl.li/vlpsnz>.
- [3] Google AI Blog, "Understanding Searches Better Than Ever Before". [Online]. Available: <http://surl.li/wvhmqi>.
- [4] Google Jigsaw, "Perspective API Using BERT". [Online]. Available: <https://urlzs.com/vczAs>.
- [5] W3School, "Machine Learning - Linear Regression". [Online]. Available: <http://surl.li/rlbrd>. Accessed: 10.11.2023.
- [6] "Spam Classification with BERT," arXiv:2102.07004. [Online]. Available: <https://arxiv.org/abs/2102.07004>. Accessed: 10.11.2023.
- [7] "Legal Document Search with BERT," arXiv:2105.09121. [Online]. Available: <https://arxiv.org/abs/2105.09121>. Accessed: 10.11.2023.
- [8] "Plagiarism Detection with BERT," arXiv:1910.00291. [Online]. Available: <https://arxiv.org/abs/1910.00291>. Accessed: 10.11.2023.
- [9] J. Leal, "Using RoBERTa for text classification," 2020. [Online]. Available: <https://urlzs.com/qafnF>. Accessed: 10.11.2023.
- [10] Grammarly. "Grammar and AI Writing." [Online]. Available: <https://urlzs.com/w9y66>. Accessed: 10.11.2023.
- [11] "RoBERTa: A Robustly Optimized BERT Pretraining Approach." [Online]. Available: <http://surl.li/xvxnbw>. Accessed: 10.11.2023.
- [12] J. Leal. "A Novel Approach to Fake News Classification Using LSTM-Based Deep Learning Models." [Online]. Available: <http://surl.li/quygwI>. Accessed: 10.11.2023.
- [13] R. Zhang, M. Zhang, Y. Zhang, "Hierarchical Co-Attention Selection Network for Interpretable Fake News Detection." [Online]. Available: <http://surl.li/pdkggu>. Accessed: 10.11.2023.
- [14] S. K. Lee, H. Jeong, "A Deep Neural Network for Fake News Detection." [Online]. Available: <https://urlzs.com/BnPri>. Accessed: 10.11.2023.
- [15] A. Dasgupta, S. Agerri, "Natural Language Inference over Interaction Space." [Online]. Available: <https://urlzs.com/FYcRg>. Accessed: 10.11.2023.
- [16] S. Devlin, M. Chang, K. Lee, "BERT for Language Translation." [Online]. Available: <http://surl.li/xthhii>. Accessed: 10.11.2023.
- [17] P. Wu, Z. Liu, L. Han, "Fake News Detection via GRU-Based Ensemble Model in Social Media," IEEE Access, vol.

- 8, pp. 47103-47115, 2020. DOI: 10.1109/ACCESS.2020.2973912. (Q1)
- [18] E. E. Lazaridis, A. Drosos, "Fake News Classification Using GRU with Attention Mechanism: A Comparative Study," *Information Processing & Management*, vol. 58, no. 4, 2021, pp. 102-109. DOI: 10.1016/j.ipm.2021.102594. (Q2)

ОЦІНКА АПАРАТНИХ ВИТРАТ ДЛЯ ФІЛЬТРА БЛУМА З ЛІЧИЛЬНИКАМИ В СИСТЕМАХ ЗАХИСТУ ІНФОРМАЦІЇ НА ПЛІС

С.Я. Гільгурт¹

¹ Department of Mathematical and Econometric Modeling, Georgy Pukhov Institute for Modelling in Energy Engineering of the National Academy of Sciences of Ukraine, Kyiv, Ukraine

E-mail: hilgurt@ukr.net

Отримано 21.10.2024

Прийнято до публікації 25.10.2024

Опубліковано 01.11.2024

АНОТАЦІЯ

Мета роботи полягає в дослідженні рішень та шляхів практичної реалізації щодо організації динамічної зміни складу патернів, які розпізнаються фільтром Блума в апаратних (на базі ПЛІС) сигнатурних системах захисту інформації (в тому числі критичної інфраструктури), таких як мережеві системи виявлення вторгнень, антивіруси, спам-фільтри тощо. Під зміною складу мається на увазі як додавання, так і вилучення окремих елементів з переліку патернів, які відшукує сигнатурна система в потоці вхідних даних, зокрема в тілах пакетів мережевого трафіку.

Для досягнення мети в роботі проаналізовані принципи побудови та функціонування фільтра Блума, його переваги та недоліки порівняно з іншими схемами розпізнавання в термінах відомих показників ефективності. Розглянуті модифікації та різновиди, запропоновані розробниками з метою покращення його характеристик протягом всього періоду використання в системах кіберзахисту комп'ютерних мереж. Проаналізовано особливості апаратної реалізації фільтра Блума на ПЛІС.

З двох виявлених підходів до вирішення проблеми динамічного переналаштування в роботі розглянутий той, що забезпечує більшу швидкодію за рахунок зміни апаратної структури пристрою, а саме фільтр Блума з лічильниками. На прикладі однієї з модифікацій апаратної схеми, так званого спрощеного фільтра Блума, розглянуто можливий варіант побудови його цифрової структури. Для підвищення ефективності процесу розробки систем захисту інформації з використанням даної схеми сформована функція оцінки апаратних ресурсів, яка дозволяє знаходити кількісні характеристики витратна синтез цифрових пристроїв у ПЛІС без виконання витратної процедури повної компіляції проєкту. Проведено попереднє порівняння отриманого виразу з функцією оцінки для схеми спрощеного фільтра Блума без лічильників.

Ключові слова: МСВВ, множинне розпізнавання патернів, ПЛІС, фільтр Блума з лічильниками, апаратні витрати.

ВСТУП

У 1970 році Бёртоном Говардам Блумом була запропонована схема обробки даних, заснована на використанні геш-функцій [1]. Сьогодні вона відома як фільтр Блума (ФБ), англійська назва – Bloom Filter (BF). Її метою було забезпечити більш компактне зберігання даних в застосуваннях, коли є прийнятною певна частка хибних спрацювань. Винахід виявилася вдалим і почав широко використовуватися спочатку в сфері традиційного застосування гешування, а згодом і в суміжних технічних галузях.

Оскільки задачі обробки даних, де використовувався ФБ, пов'язані з інтенсивною обробкою інформації, виникла потреба у прискоренні операцій, які він здійснює. Ефективною основою для апаратного прискорення виявилися пристрої на базі програмованих логічних інтегральних схем (ПЛІС) типу Field-programmable gate array (FPGA). Почалося використання програмованої логіки для синтезу ФБ ще в середині 1990-х років. Історично першим застосуванням програмованої логіки для побудови цієї схеми обробки даних з метою швидкого розпізнавання тексту де-факто стала робота [2], хоча власне назва "фільтр Блума" в ній не згадувалася. Згодом подібні розробки почали з'являтися все частіше. Причому більша їх частка була присвячена мережевим застосуванням.

Починаючи з середини 2000-х, фільтр Блума активно використовується в галузі технічного захисту інформації, зокрема в якості основного засобу розпізнавання при побудові мережевих систем виявлення вторгнень (МСВВ). При цьому в переважній більшості застосувань ФБ вирішує ресурсоємну задачу множинного розпізнавання патернів (pattern matching) – ключову обчислювальну задачу сигнатурних засобів захисту інформації [3]. Сутність цієї задачі полягає в одночасному порівнянні фрагменту вхідних даних (наприклад, кількох байтів тіла мережевого пакету) з великою кількістю патернів – фіксованих послідовностей символів, що входять до складу сигнатур – описів відомих атак. Такому застосуванню ФБ свого часу посприяла активна діяльність проф. Джона Вільяма Локвуда з Вашингтонського університету та його численні наукові публікації за даною темою [4]. Але й досі розробники мережевих засобів захисту інформації не втратили інтерес до фільтра Блума, використовуючи його, зокрема, як "перший ешелон" захисту [5]. Найбільш поширеною платформою для

використання ПЛІС на сьогодні є реконфігуровні обчислювачі або прискорювачі, англійська назва – Reconfigurable Accelerator (RA), які крім кристала ПЛІС містять бортовий оперативний запам'ятовуючий пристрій (ОЗП), контролер зв'язку з хостовою комп'ютерною системою, у випадку мережевих застосувань – Ethernet-порти та інші компоненти [6].

Нижче наведено короткий літературний огляд, в якому висвітлені основні властивості фільтра Блума, його переваги та недоліки порівняно з іншими схемами розпізнавання, які використовуються в системах захисту інформації, шляхи усунення недоліків та підвищення ефективності застосування ФБ. Також розглянуто специфіку реалізації цифрової схеми ФБ на ПЛІС, складності та проблеми, що виникають при цьому, та можливі рішення щодо їх подолання. Розглянуто критерії оцінки технічних показників ФБ з точки зору застосування для побудови систем кіберзахисту й метод прискореного обчислення їх кількісних характеристик. Сформульована технічна проблема, пов'язана з неможливістю класичного ФБ змінювати набір даних, що підлягають виявленню.

АНАЛІЗ ЛІТЕРАТУРНИХ ДАНИХ ТА ПОСТАНОВКА ПРОБЛЕМИ

На сьогодні застосування ФБ охоплює велику кількість предметних галузей (Рис. 1) [7].

Результати поглибленого дослідження існуючої літератури з оптимізації ФБ наведено в публікації [8].

Багато прикладів використання ФБ в галузі захисту інформації можна знайти у вичерпному дослідженні [9], а також в більш сучасному огляді [10]. Проводяться дослідження також щодо застосування ФБ для вирішення проблем кіберзахисту об'єктів критичної інфраструктури, зокрема, SCADA-систем [11].

Структура та принцип дії класичного фільтра Блума.

ФБ складається з двох ключових компонентів, як показано на Рис. 2: комплекту з K блоків, що обчислюють геш-функції $h_1(x)$, $h_2(x)$, $h_3(x)$, ..., $h_K(x)$ та масиву з M бітових комірок (компонент R_g на рисунку) [12]. Назвемо цей масив регістром бітів (РБ). В початковому стані він заповнений нулями.

На етапі програмування або навчання (Рис. 2, а) на входи всіх геш-функцій послідовно подається кожен з N елементів словнику патернів (довжиною W бітів). Для кожного елемента обчислюються значення всіх K геш-функцій, отримані значення яких інтерпретуються як

номер (адреса) комірки у РБ. У відповідні комірки заносяться значення "1".

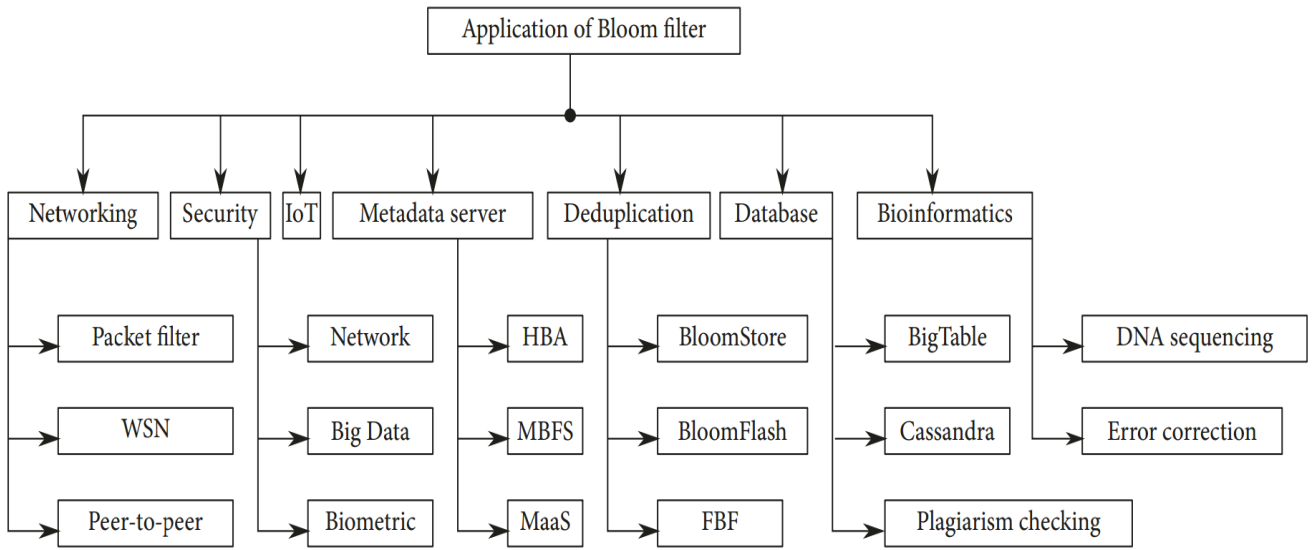


Рис. 1. Галузі застосування фільтра Блума [7]

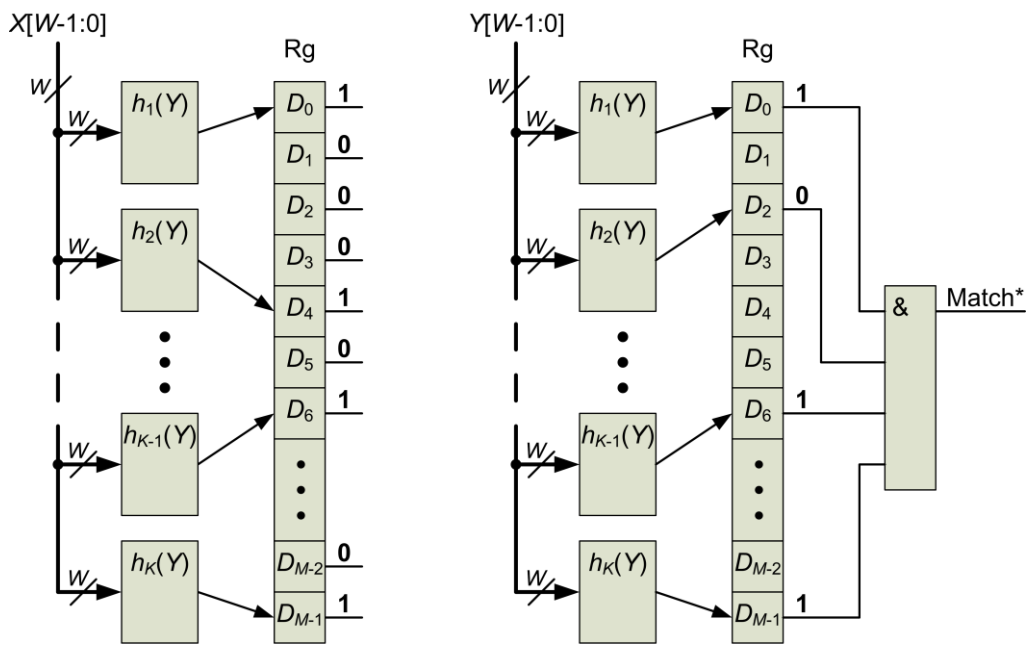


Рис. 2. Функціонування фільтра Блума: а – в режимі програмування; б – в процесі розпізнавання

В процесі функціонування фільтра Блума (Рис. 2, б) на його вхід подається фрагмент вхідної послідовності символів (також довжиною W бітів). Для нього обчислюються значення всіх K геш-функцій. По отриманим адресам здійснюється звернення до комірок РБ. Якщо у всіх позиціях, на які вкажуть геш-функції,

містяться одиниці, вважається, що вхідна комбінація символів з певною вірогідністю співпадає з одним з патернів, що брали участь у програмуванні ФБ. Але якщо хоча б одна геш-функція вкаже на комірку з нульовим значенням, це гарантовано свідчить про відсутність

відповідної послідовності символів серед навчальної множини патернів.

Отже, ФБ функціонує з деякою вірогідністю помилки розпізнавання другого роду (false positive), але без помилок першого роду (false negative). Тобто інколи він може сигналізувати про наявність збігу, якого насправді немає, але ніколи не пропустить справжній збіг. Значення вірогідності помилки другого роду для заданих значень кількості патернів N , розміру M регістра бітів, та кількості геш-функцій K знаходиться згідно виразу [9]:

$$p_{\text{пом.2}} = \left(1 - \left(1 - \frac{1}{M}\right)^{KN}\right)^K \approx \left(1 - e^{-\frac{KN}{M}}\right)^K. \quad (1)$$

При цьому значення M покладається досить великим, що відповідає дійсності.

Як можна бачити, збільшення кількості патернів призводить до збільшення системної помилки ФБ, в той час як збільшення розміру РБ – до зменшення помилки. Кількість геш-функцій K ще більш суттєво впливає на вірогідність помилки розпізнавання другого роду. Доведено [9], що ця вірогідність мінімальна при певному співвідношенні значень N та M :

$$K = \frac{M}{N} \ln 2 \approx 0,6931 \frac{M}{N}. \quad (2)$$

При цьому вірогідність помилки

$$p_{\text{пом.2}} = 2^{-K} \approx 0,6185^{M/N}. \quad (3)$$

Оскільки геш-функції обчислюються цифровими схемами у двійковій системі числення, при забезпеченні даного співвідношення під час вибору параметрів ФБ розмір M регістра бітів обирають, як правило, таким, що дорівнює ступеню двійки.

Аналізуючи наведені вище відомості щодо основ функціонування фільтра Блума можна зробити попередню оцінку його можливостей щодо розпізнавання.

По-перше, ФБ дозволяє дуже економно використовувати ресурси пам'яті. Розмір словнику патернів не впливає на прямо на розмір РБ. Додання нових патернів до вже присутніх у цієї структурі даних призводить лише до підвищення вірогідності помилки розпізнавання, але не до збільшення об'єму запам'ятовуючого пристрою. По-друге, кількість та складність геш-функцій, які потрібно обчислювати в процесі розпізнавання, тобто, апаратна витратність та продуктивність при даному підході теж не залежать від об'єму словника патернів. Нарешті, довжина патернів також не впливає на прямо ні на кількість ресурсів пам'яті, ані на продуктивність. Отже ФБ добре масштабується по двох напрямках: за об'ємом словнику патернів та за довжиною патернів, що відшукуються. Остання властивість має важливе значення: навіть дуже довгі

рядки після перетворення геш-функціями потребують для зберігання ті ж самі K комірок РБ.

Кластеризація. На жаль, фільтру Блума притаманний важливий недолік (як будь-яким рішенням на базі геш-функцій): розмір вхідної послідовності символів, тобто довжина патерну має бути фіксованою для обраного набору геш-функції. Інакше кажучи, один ФБ здатен розпізнавати патерни тільки однакової довжини. Єдиною можливістю подолання даного недоліку є кластеризація, тобто ранжування патернів по довжині та спільне застосування кількох ФБ, що розпізнають патерни різної довжини (див. Рис. 2. у [12]). Це призводить до збільшення витрат ресурсів, що значно погіршує вартівні показники ефективності ФБ. Але це погіршення не стосується швидкісних показників, про що свідчить, зокрема, той факт, що певна кількість успішних практичних розробок МСВВ на базі фільтра Блума фактично є системами запобігання вторгнень, для яких висувуються значно жорсткіші вимоги щодо продуктивності [4, 13, 14].

Уточнення результатів. Як було вказано вище, Фільтру Блума притаманні системні помилки розпізнавання другого роду. Тому результати розпізнавання потрібно уточнювати, тобто переконатися, чи насправді виявлена послідовність символів співпадає з патерном. Операція уточнення може бути виконана програмно на хост-комп'ютері, апаратними засобами із використанням частки ресурсів програмованої логіки [4] або за допомогою вбудованих в ПЛІС процесорних ядер універсальної архітектури [13]. У двох останніх випадках цю операцію зазвичай виконують із застосуванням вторинної геш-таблиці, яка зберігається в бортовому ОЗП реконфігуровного обчислювача, зовнішньому по відношенню до мікросхеми ПЛІС.

Операція уточнення виконується значно повільніше аніж процедура розпізнавання. Це не тільки уповільнює роботу системи, але також підвищує вразливість від атак на МСВВ шляхом насичення мережевого трафіку пакетами, що імітують напад (тобто співпадають з патернами бази даних сигнатур МСВВ). Втім, вірогідність хибного спрацьовування ФБ можна зменшити до потрібного значення шляхом вибору належного співвідношення параметрів K та M .

Розпаралелювання. Підвищити продуктивність фільтра Блума можливо за рахунок паралельного підключення із зсувом кількох однакових блоків, кожен з яких містить потрібний комплект фільтрів Блума. (див. Рис. 3 у [12]). Дана техніка збільшує апаратні витрати рівно в стільки разів, в скільки пришвидшує процес

розпізнавання. Тобто фільтр Блума також має добру масштабованість за швидкодією.

Подальше підвищення швидкодії. Відомо про багато спроб дослідників підвищити швидкісні характеристики підходу до розпізнавання патернів на основі ФБ без розпаралелювання. Але чи не єдиним здобутком виявилася техніка каскадування, яка на практиці призвела до суттєвого зниження енергозбереження, але не до прискорення, на що вказано в роботі [15]. В якості ілюстрації докладених зусиль можна навести це ж саме дослідження, в якому хоч і вдалося підвищити продуктивність системи МСВВ у цілому, але за рахунок маніпулювань із запитами на обробку вхідних даних на рівні мережевих пакетів, не змінюючи власне структуру фільтра Блума.

Особливості реалізації фільтра Блума на ПЛІС. Для створення ефективного ФБ реконфігурованими засобами має важливе значення вибір геш-функцій. Ці функції мають, по-перше, однозначно відображати множину комбінацій вхідних символів на множину вихідних значень, по-друге, цифрова схема їх реалізації мати якомога простішу апаратну структуру, щоб забезпечити високу швидкодію при невеликих ресурсних витратах, по-третє, мають легко реалізовуватися на ПЛІС. Сформульованим вимогам задовольняють так звані геш-функції класу H_3 [16]. Притаманна ним властивість рекурсивності призводить до регулярності цифрової схеми та дозволяє знизити потрібні ресурси.

Іншою особливістю базової схеми фільтра Блума (Рис. 1) є необхідність одночасного доступу до масиву бітів R_g багатьох геш-функцій. У разі його реалізації на єдиному пристрої зберігання даних можуть виникнути колізії. Помірні вимоги ФБ до обсягу пам'яті дозволяють реалізовувати бітовий масив в на базі блоків вбудованої пам'яті BRAM мікросхеми ПЛІС. Такі блоки дозволяють синтезувати двопортові пристрої пам'яті, що забезпечує одночасний доступ до комірок бітового масиву двох геш-функцій. Але реальна кількість геш-функцій в практичних додатках зазвичай в кілька разів більша.

Рішення полягає у розбитті бітового масиву на декілька пристроїв, для чого потрібно, по-перше, розподілити виходи блоків геш-функцій між відповідними ОЗП, по-друге, на функціональність цих блоків накласти додаткові обмеження, щоб їх вихідний діапазон укладався у зменшений об'єм запам'ятовуючого пристрою. Такі обмеження впливають на вірогідність помилок другого роду ФБ, втім не суттєво [12].

Оцінка ефективності ФБ та потреба в динамічному переналаштуванні. В роботі [17] розглянуто критерії

оцінки ефективності та відповідні показники для реконфігурованих сигнатурних систем захисту інформації. Потреба в засобах швидкої оцінки кількісних характеристик таких систем та їх компонентів обґрунтована у [3]. Там же розглянуто метод прискореного розрахунку таких характеристик без потреби виконання витратної за часом повної процедури компіляції проекту щодо синтезу цифрової схеми в ПЛІС.

Одним з важливих не кількісних показників ефективності сигнатурних реконфігурованих систем захисту інформації є здатність змінювати склад *патернів* (додавати та вилучати), що них має відшукувати система у вхідному потоку даних, без припинення її функціонування та без реконфігурації ПЛІС [17, 18]. Назвемо цю функцію здатністю до *динамічного переналаштування*. Класична структура ФБ програмується (конфігурується) перед початком роботи і принципово не здатна в процесі розпізнавання вилучати патерни, які були неявно внесені до регістра бітів на етапі програмування. Однак мінлива та динамічна природа завдань захисту інформації в комп'ютерних мережах в деяких ситуаціях вимагає зміни набору патернів, що розпізнаються.

Метою дослідження є дослідження можливих рішень та шляхів їх практичної реалізації щодо організації динамічної зміни складу патернів, що розпізнаються реконфігурованим фільтром Блума на базі ПЛІС, без припинення його функціонування та без перезавантаження в ПЛІС конфігураційної послідовності. Важливим завданням також є знаходження функції оцінки для знайденого рішення, що дозволить задіяти метод прискореного розрахунку кількісних характеристик при проектуванні цифрової схеми ФБ з новою властивістю під час практичної реалізації даного рішення.

МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ

В роботі використовувалися методи булевої алгебри, цифрової схемотехніки, теорії обчислювальних систем, а також основи теорій обчислень на рядках та реконфігурованих обчислень. Втім, в якості основного інструменту застосовано метод прискореної оцінки кількісних характеристик реконфігурованих сигнатурних систем захисту інформації [3, 18]. Метод базується на використанні так званих функцій оцінки (ФО), які дозволяють знаходити ресурсні та часові параметри модулів розпізнавання таких систем без виконання процедури компіляції проекту створення в ПЛІС обчислювальної структури. Головну увагу в даному дослідженні зосереджено на ресурсній складовій ФО,

тобто на апаратних витратах на створення ФБ з лічильниками.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

МОЖЛИВОСТІ ФІЛЬТРА БЛУМА ЩОДО ДИНАМІЧНОГО ПЕРЕНАЛАШТУВАННЯ

Аналіз базової схеми ФБ свідчить про те, що якщо додавати патерни в процесі його функціонування можливо, то вилучати їх дана схема не дозволяє принципово. Розмаїття відомих модифікацій ФБ надає два варіанти вирішення проблеми його динамічного переналаштування.

Перший варіант пов'язаний з використанням так званого фільтра Блума з лічильниками, Counting Bloom Filter (CBF). Вперше подібна модифікація була запропонована ще в 2000-му році [19]. В схему додається M лічильників (по числу розрядів регістра бітів). Кожний лічильник розташовується між виходом блоку обчислення геш-функцій та відповідним входом регістра бітів. В процесі програмування фільтра (або додання під час роботи) обчислені значення геш-функцій подаються на лічильників як вхідні імпульси в режимі додавання. Якщо в процесі функціонування ФБ виникає потреба вилучити патерн з набору, значення виходів геш-функцій керують лічильниками в режимі віднімання. Перехід числа, накопиченого в лічильнику, зі значення "0" у значення "1" призводить до запису "1" у відповідну комірку РБ фільтра Блума; перехід лічильника зі стану "1" в стан "0" – до запису "0". В іншому принцип функціонування схеми не змінюється. Таким чином, техніка використання лічильників дозволяє динамічно переналаштовувати ФБ в реальному часі без перепрограмування ПЛІС та тимчасового припинення функціонування сигнатурної системи захисту інформації.

Недоліком схеми фільтра Блума з лічильниками є додаткові витрати ресурсів ПЛІС. Причому точне значення розрядності лічильників складно передбачити, тому що кількість "влучань" різних геш-функцій в одну й ту саму комірку РБ є випадковою величиною. Якщо використати лічильники недостатньої розрядності, їх переповнення призводитиме до виникнення помилок першого роду (false negative), тобто, схема буде розпізнавання не всі патерни, що були запрограмовані. У загальному сенсі це гірше за притаманні ФБ помилки другого роду, які технічно можуть бути виправлені додатковими заходами. Але, як свідчать розрахунки, вірогідність таких подій невелика. В роботі [19] наведено вирази для оцінки вірогідності помилок першого роду. Виходячи з їх аналізу

робиться припущення, що 4-розрядних лічильників достатньо для більшості практичних застосувань.

Другий варіант вирішення проблеми передбачає перерахунок значень РБ програмними засобами з послідовним завантаженням до нього вже модифікованого набору бітів. Такий прийом не потребує додаткових апаратних витрат порівняно з базовою схемою. Взагалі, будь-які зміни завантаженої в ПЛІС конфігурації не потрібні. (Відповідно, для функцій оцінки кількісних характеристик ФБ можуть бути використані відомі вирази для класичних схем ФБ). Але, по-перше, в цьому випадку необхідно задіяти спеціальний режим конфігурації, при якому всередині ПЛІС відновлюється тільки зміст внутрішньої, блокової пам'яті BRAM, а інші частини кристалу залишаються незмінними. По-друге, програмний режим оновлення значень РБ суттєво уповільнює процес динамічного переналаштування (у випадку потреби вилучити патерн з набору, що розпізнається, потрібно повністю повторити процедуру програмування). І, якщо першу умову в сучасних реконфігурованих прискорювачах реалізувати відносно нескладно, другий момент може звести нанівець всі переваги такого підходу, якщо на процедуру динамічного переналаштування накладаються суворі часові обмеження.

Тому нижче розглянуто реалізацію на ПЛІС саме схеми CBF. Також розраховано та наведено ресурсну ФО для даної схеми.

СТРУКТУРНА СХЕМА ФІЛЬТРА БЛУМА З ЛІЧИЛЬНИКАМИ

В роботі [20] докладно досліджено різні схеми розпізнавання для реконфігурованих систем захисту інформації, включаючи різні модифікації ФБ.

Схема генераторів геш-функцій (ГГФ) не залежить від модифікації та має B входів і K виходів по G розрядів кожний, де B – кількість розрядів у патерні, яка дорівнює здобутку кількості символів у патерні та кількості бітів у слові (8 – при байтовому кодуванні), K – кількість геш-функцій у ФБ, G – розрядність геш-функцій (двійковий логарифм від довжини M регістра бітів).

Як згадувалося вище під час обговорення особливостей реалізації ФБ на ПЛІС, вбудована блокова пам'ять BRAM зазвичай дозволяє створити, максимум, двопортові пристрої пам'яті, які надають одночасний доступ до комірок бітового масиву лише двох геш-функцій. З іншого боку, кількість K геш-функцій визначає вірогідність помилки розпізнавання другого роду: ця вірогідність обернено пропорційна двійці у ступеню K .

Наприклад, якщо задіяне 5 блоків BRAM, $K = 10$, і ця вірогідність $p_{\text{пом.2}} = 2^{-10} \approx 0,00098$.

При оптимальному співвідношенні основних параметрів ФБ для $K = 10$ і $M = 4096$ (типовий розмір блоку BRAM) схема буде здатна розпізнавати $N \approx 1420$

патернів. У випадку, коли цієї кількості патернів достатньо, схема РБ спрощується до модифікації так званого спрощеного фільтра Блума – Simplified Bloom Filter (SBF). Реалізація схеми регістра бітів для даної модифікації на ПЛІС наведена на Рис. 3.

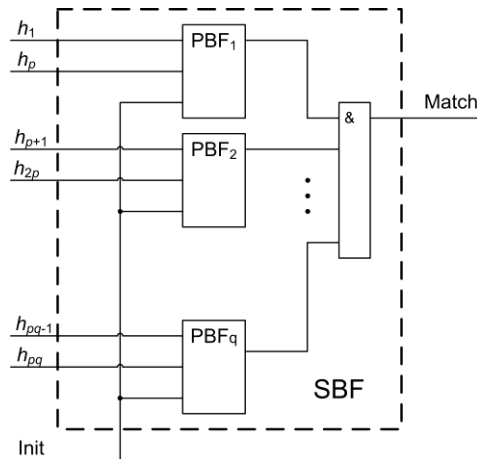


Рис. 3. Схема SBF регістра бітів для спрощеного фільтра Блума

Тут PBF (Partial Bloom Filter) – схема РБ для часткового ФБ, реалізованого на одному блоці BRAM (Рис. 4). Мультиплексор MUX дозволяє в режимі програмування завантажувати у BRAM через один з портів заздалегідь підготовлену комбінацію бітів (керує процесом завантаження інтерфейсний блок IU). В той час як в режимі функціонування обидва входи працюють на прийом значень двох геш-функцій [20].

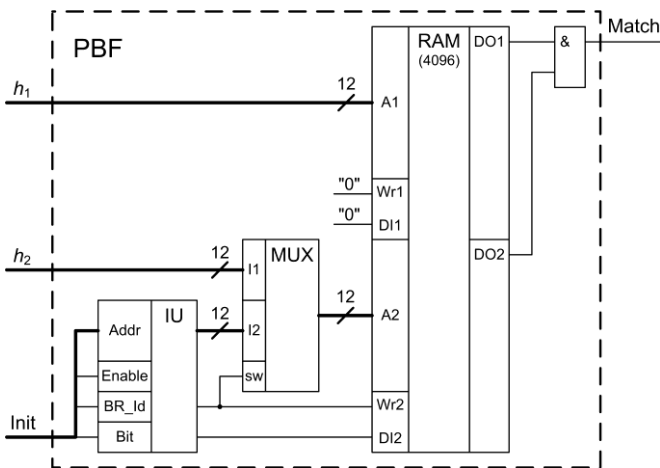


Рис. 4. Схема PBF регістра бітів для часткового фільтра Блума

Якщо ФБ повинен розпізнавати більшу кількість патернів, ніж здатен містити SBF, потрібна повнорозмірна схема РБ. Відповідна структура має назву

Повнорозмірний фільтр Блума – LBF (Large Bloom Filter) і наведена на Рис. 5.

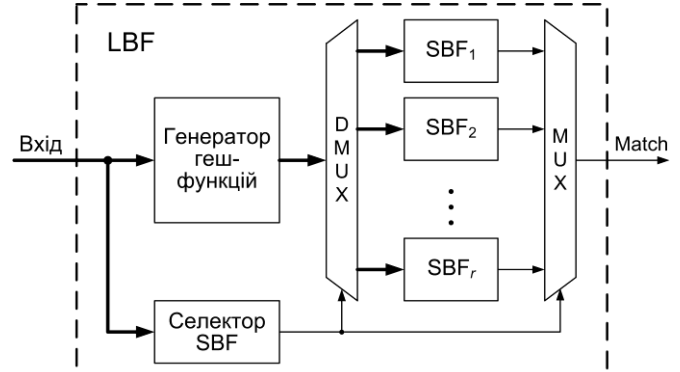


Рис. 5. Схема повнорозмірного фільтра Блума LBF

Тут поданням виходів ГГФ на входи потрібних спрощених схем регістрів SBF керує спеціальний селектор SBF (згідно алгоритму розподілу, про який йшлося наприкінці підрозділу "Особливості реалізації фільтра Блума на ПЛІС").

Розглянемо, як зміняться компоненти ФБ при реалізації рішення з використанням лічильників.

Схеми ГГФ, як і узагальнені схеми спрощеного SBF (рис. 3) та повнорозмірного LBF (Рис. 5) фільтрів Блума залишаться без змін. Модифікація торкнеться лише схеми регістра бітів для часткового фільтра Блума – PBF (Рис. 4). Назвемо цю схему CPBF (Counting Partial Bloom Filter). Оскільки додавати патерни тепер буде потрібно під час

функціонування ФБ, до схеми РБ додається ще один мультиплексор MUX (Рис. 6). Інтерфейсний блок IU замінюється блоком керування CU, який тепер керує не

тільки процесом завантаження РБ перед початком роботи, але також доданням та вилученням патернів.

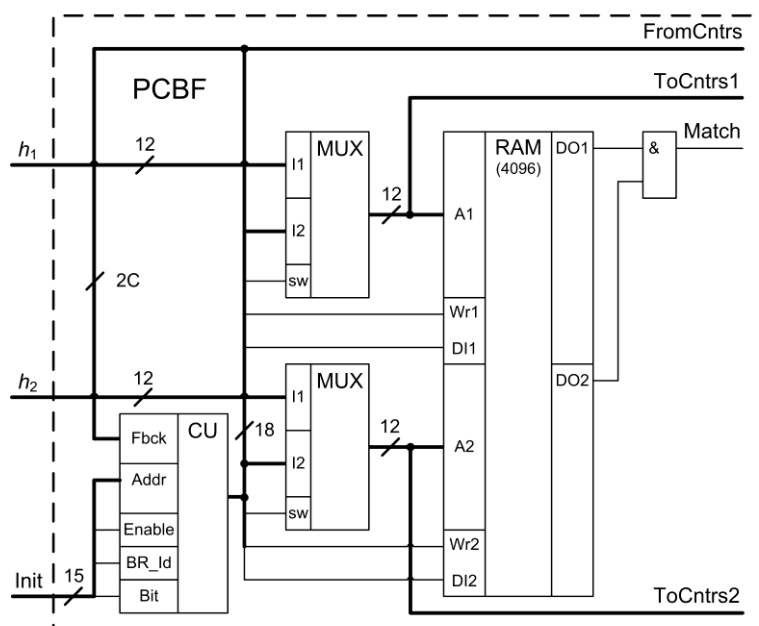


Рис. 6. Схема регістра бітів для часткового СBF

Оскільки кількість лічильників у СBF дорівнює числу M бітів у РБ, значення якого може сягати кількох тисяч, синтезувати їх з конфігураційних блоків CLB мікросхеми ПЛІС не є доцільним. В даному дослідженні пропонується задіяти для цього вбудовану швидкодіючу блокову пам'ять BRAM. Комірки ОЗП, що створюється на базі BRAM, будуть використовуватися як D-тригери, які за рахунок зворотного зв'язку функціонуватимуть як лічильні Т-тригери. Логіка з'єднань між каскадами лічильників, яка створює з Т-тригерів реверсивний асинхронний лічильник, зосереджена в блоках переносів $CL_1 \dots CL_{C-1}$ (Рис. 7). Тут C – кількість розрядів лічильників. Перемикання між режимами додавання/віднімання, а також занесення нуля або одиниці у відповідну комірку РБ також здійснює CU.

Сумістити РБ з наймолодшим розрядом лічильників неможливо, оскільки в такому випадку не буде змоги відслідковувати події переходів лічильників в нульовий стан та – з нульового стану, в результаті яких в комірки РБ повинні заноситися нулі та одиниці відповідно.

ФУНКЦІЇ ОЦІНКИ ДЛЯ ФІЛЬТРА БЛУМА З ЛІЧИЛЬНИКАМИ

Як було згадано вище, метод швидкої оцінки кількісних характеристик реконфігурованих сигнатурних систем захисту інформації базується на створенні та використанні

функцій оцінки [3, 18]. Ці функції описують залежність від набору патернів, що розпізнаватимуться, таких параметрів синтезованих в ПЛІС цифрових схем, як апаратні (ресурсні) витрати та швидкодія.

Принцип побудови ФО апаратних витрат базується на підрахунку всіх апаратних ресурсів, які задіяні при створенні відповідного компонента або системи в цілому. При цьому, щоб звести підрахунки до якоїсь універсальної одиниці, наприклад, "умовних LUT" – логічних таблиць (ЛТ) мікросхеми ПЛІС, різні ресурси додаються з відповідними коефіцієнтами. Тоді загальну кількість апаратних витрат на синтез певної схеми можна підрахувати за формулою:

$$R = L + \alpha F + \beta B + \gamma S, \quad (4)$$

де L – кількість ресурсів логіки ПЛІС (кількість ЛТ),

F – кількість ресурсів розподіленої пам'яті ПЛІС (кількість тригерів),

B – кількість ресурсів блокової пам'яті ПЛІС (кількість блоків BRAM),

S – кількість ресурсів зовнішньої пам'яті – об'єм бортової пам'яті реконфігурованого прискорювача (γ Мб),

α, β, γ – коефіцієнти нормалізації відносно ЛТ ресурсів різного типу, відповідно: кількості тригерів, числа блоків BRAM та об'єму бортової пам'яті RA.

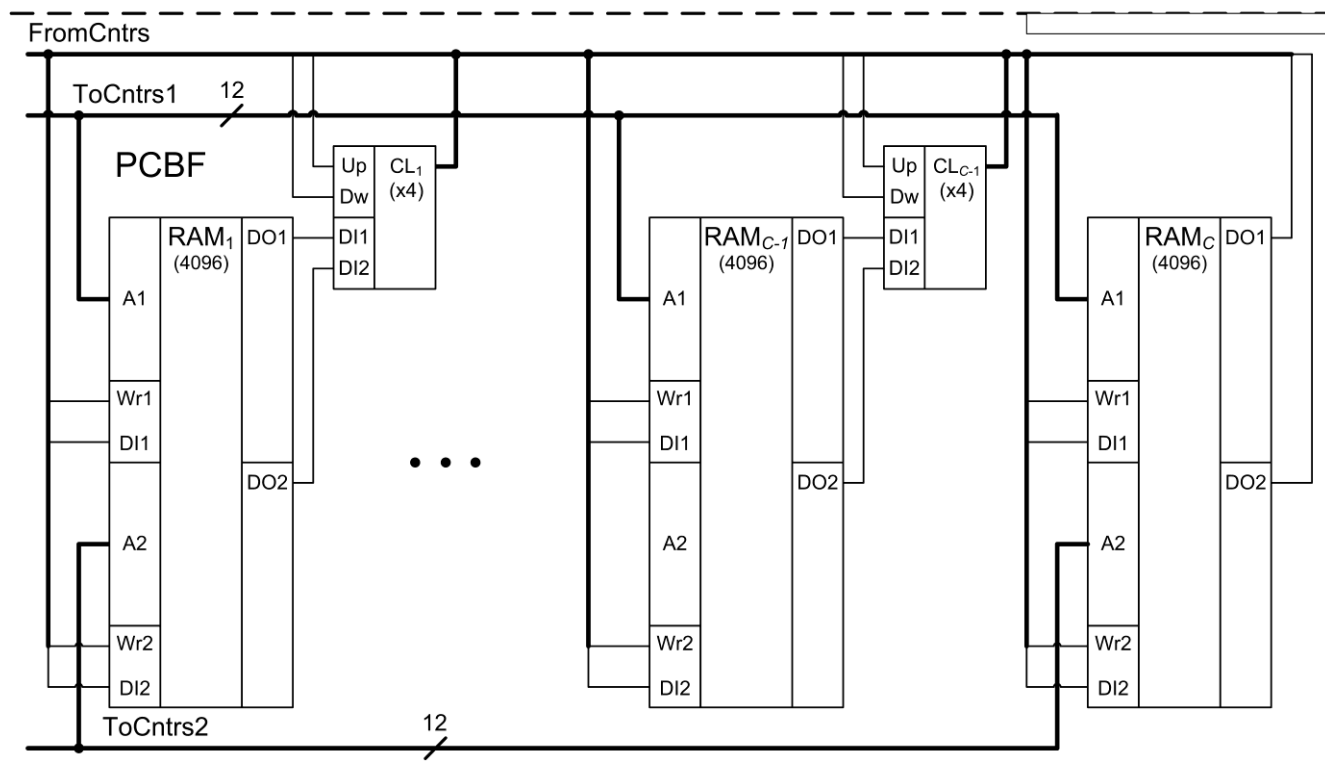


Рис. 7. Схема лічильників для часткового CBF

Слід зауважити, що формула (4) лише демонструє основну ідею підходу до оцінки апаратних витрат. Її нескладно розвинути шляхом додавання інших ресурсів ПЛІС та RA з відповідними коефіцієнтами.

Для складної цифрової схеми кількість використаних ресурсів дорівнює сумі ресурсів кожного компонента, з яких ця схема складається. Використовуючи такий підхід можна знайти ресурсні витрати R_{BF} на синтез ФБ, які містять витрати на генератор геш-функцій R_{GF} , на регістр бітів R_{PB} та на допоміжні схеми $R_{доп}$:

$$R_{BF} = R_{GF} + R_{PB} + R_{доп} = L_{GF} + \alpha F_{GF} + \beta B_{PB} + L_{доп} + \alpha F_{доп} = L_{GF} + L_{доп} + \alpha(F_{GF} + F_{доп}) + \beta B_{PB}, \quad (5)$$

де L_{GF} і $L_{доп}$ – витрати ЛТ відповідно на ГГФ та на допоміжні схеми,

F_{GF} і $F_{доп}$ – витрати тригерів відповідно на ГГФ та на допоміжні схеми,

B_{PB} – витрати на блокову пам'ять для РБ, які вимірюються кількістю задіяних блоків BRAM кристалу ПЛІС.

Якщо підрахувати всі види ресурсів, наприклад, для схеми спрощеного фільтра Блума SBF (див. докладні розрахунки у [20]), отримуємо наступну ресурсну функцію оцінки:

$$R_{SBF} = EG \left(\left\lceil \frac{8j}{x} \right\rceil + (\alpha + 1) \left\lceil \frac{\left\lceil \frac{8j}{x} \right\rceil - 1}{x - 1} \right\rceil - \alpha \right) + q \left(\alpha G + \beta + \left\lceil \frac{G}{\left\lfloor \frac{x-1}{2} \right\rfloor} \right\rceil + 4 \right) + \left\lceil \frac{q-1}{x-1} \right\rceil, \quad (6)$$

де $E = -\lceil \log_2 \rho_{доzv} \rceil$ – фактор хибного розпізнавання, який чисельно дорівнює кількості геш-функцій у ФБ і обернено пропорційний логарифму ймовірності хибного спрацьовування, прийнятого для конкретного застосування фільтра Блума;

$G = \left\lceil \log_2 \frac{E \cdot \delta_j}{\ln 2} \right\rceil$ – розрядність геш-функцій (дорівнює довжині адресу блоків BRAM), де δ_j – функція розподілу довжин патернів (кількість патернів довжиною j символів в наборі);

j – довжина патернів, які розпізнає даний ФБ;

x – число входів ЛТ конкретної ПЛІС, що використана в RA;

$q = \left\lceil \frac{E}{p} \right\rceil$ – кількість PBF в складі SBF, де p – число портів блокової пам'яті BRAM.

Слід зауважити, що змінними у виразі (6) є: фактор хибного розпізнавання E , функція розподілу довжин патернів δ_j та довжина патернів j , тоді як решта змінних, які залежать від типу застосованої ПЛІС, є константами.

При формуванні ресурсної ФО для схеми з лічильниками, побудованій на основі спрощеного ФБ – SCBF, деякі складові виразу (6) зміняться. У підрахунок витрат логіки на допоміжні схеми в складі часткового ФБ з лічильниками PCBF в схему CU додається $2(C-1)$ входів, кількість мультиплексорів MUX подвоюється, витрачаються логічні ресурси також на блоки переносів CL (по 4 ЛТ на кожний):

$$L_{\text{допPCBF}} = 3 + \left\lfloor \frac{2(C-1)}{x} \right\rfloor + 2 \left\lfloor \frac{G}{\left\lfloor \frac{x-1}{2} \right\rfloor} \right\rfloor + 1 + \varepsilon + 4(C - 1) = \left\lfloor \frac{2(C-1)}{x} \right\rfloor + \varepsilon + 2 \left\lfloor \frac{G}{\left\lfloor \frac{x-1}{2} \right\rfloor} \right\rfloor + 4C. \quad (7)$$

Кількість тригерів в складі схеми CU збільшиться на 4:

$$F_{\text{допPCBF}} = G + 4. \quad (8)$$

Нарешті, у загальній схемі збільшиться кількість задіяних ресурсів вбудованої пам'яті – блоків BRAM:

$$B_{\text{SCBF}} = q(1 + C). \quad (9)$$

Тоді ресурсна ФО для схеми спрощеного ФБ з лічильниками матиме вигляд:

$$\begin{aligned} R_{\text{SCBF}} &= EG \left(\left\lfloor \frac{8j}{x} \right\rfloor + \left\lfloor \frac{\left\lfloor \frac{8j}{x} \right\rfloor - 1}{x-1} \right\rfloor \right) \\ &\quad + q \left(\left\lfloor \frac{2(C-1)}{x} \right\rfloor + 2 \left\lfloor \frac{G}{\left\lfloor \frac{x-1}{2} \right\rfloor} \right\rfloor + 4C \right) \\ &\quad + \left\lfloor \frac{q-1}{x-1} \right\rfloor + \\ &\quad + \alpha \left(EG \left(\left\lfloor \frac{\left\lfloor \frac{8j}{x} \right\rfloor - 1}{x-1} \right\rfloor - 1 \right) + q(G+4) \right) + \beta q(1+C) = \\ &= EG \left(\left\lfloor \frac{8j}{x} \right\rfloor + (\alpha+1) \left\lfloor \frac{\left\lfloor \frac{8j}{x} \right\rfloor - 1}{x-1} \right\rfloor - \alpha \right) + \alpha q(G+4) + \\ &\quad + q \left(\left\lfloor \frac{2(C-1)}{x} \right\rfloor + 2 \left\lfloor \frac{G}{\left\lfloor \frac{x-1}{2} \right\rfloor} \right\rfloor + 4C + \beta(1+C) \right) + \left\lfloor \frac{q-1}{x-1} \right\rfloor. \quad (10) \end{aligned}$$

Аналізуючи вираз (10), можна звернути увагу на відсутність в ньому, як і у виразі (6), коефіцієнта γ . Це означає, що схеми фільтра Блума, що описуються даними ФО, не споживають ресурсів зовнішньої по відношенню до ПЛІС пам'яті (бортового ОЗП прискорювача), а тільки логічні таблиці, тригери та блоки вбудованої пам'яті BRAM. Це позитивно впливає на швидкодію схем пристроїв.

ОБГОВОРЕННЯ РЕЗУЛЬТАТІВ

Порівнюючи вирази (10) та (6) можна зробити висновок, що додавання лічильників у схему ФБ не призвело до драматичного збільшення апаратних витрат. Для порівняння конкретних значень, потрібно використати реальні набори патернів з баз даних сигнатур, наприклад, розповсюджених систем MCSBV.

Отримана функція оцінки дозволяє для будь-якого набору патернів розрахувати кількість апаратних ресурсів, потрібних для синтезу фільтра Блума з лічильниками, здатного розпізнавати даний набір. При цьому не потрібно кожного разу виконувати складну і часоємну процедуру компіляції проекту для ПЛІС. Наведений приклад спрощеної схеми ФБ демонструє, яким чином можна формувати ФО для інших схем фільтрів Блума. В роботі [8] наведено декілька модифікацій ФБ з лічильниками для різних застосувань.

ВИСНОВКИ

В роботі розглянуто особливості фільтра Блума, які слід враховувати при використанні даної схеми в реконфігурованих засобах захисту інформації безпеки. До переваг даного абстрактного пристрою відноситься економне споживання ресурсів та висока пропускна здатність у випадку його апаратної реалізації. ФБ добре масштабується за швидкодією, розміру словнику патернів та довжині патернів.

Регулярна та логічна структура обумовлює непогану здатність до реалізації на ПЛІС. Головна складність, втім, полягає в одночасному зверненні множини блоків генерації геш-функцій до багаторозрядного масиву бітів, якщо останні реалізовані у вигляді ОЗП. Але ця проблема вирішується шляхом розбиття запам'ятовуючого пристрою на декілька окремих модулів та незначної корекції схематики обчислювачів геш-функцій.

До недоліків ФБ можна віднести здатність розпізнавати патерни тільки одного розміру.

Іншим обмеженням є неможливість змінювати склад патернів в процесі функціонування. З метою подолання цього обмеження використовуються модифікації ФБ з лічильниками. В роботі проаналізовані принципи побудови таких схем, розглянутий можливий варіант реалізації ФБ з лічильниками на базі ПЛІС.

Для швидкої оцінки апаратних витрат на синтез схеми, сформована функція оцінки, яка описує залежність значення ресурсів ПЛІС від набору патернів, що розпізнаватимуться. Подібним чином можна створити функції оцінки і для інших варіантів схеми ФБ з лічильниками.

Отримані в даному дослідженні результати дозволять розробникам цифрових пристроїв створювати більш ефективні реконфігуровані засоби інформаційної безпеки, в тому числі на об'єктах критичної інфраструктури, які висувають підвищені вимоги до їх продуктивності та швидкодії.

В подальшому планується проведення низки обчислювальних експериментів з метою оцінки збільшення апаратних витрат у схем з лічильниками на відкритих наборах патернів з баз даних сигнатур відомих MCSBV.

ФІНАНСУВАННЯ

Статтю підготовлено за матеріалами дослідження, яке фінансується Національною академією наук України в рамках виконання науково-дослідної роботи "Розвиток методів і засобів підвищення рівня кіберзахистності цифрових підстанцій (шифр: МОД-К)" протягом 2024-2026 років. Номер державної реєстрації: 0124U002384.

ЛІТЕРАТУРА

- [1] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors", *Communications of the ACM*, Article vol. 13, no. 7, pp. 422-426, 1970, doi: 10.1145/362686.362692.
- [2] D. V. Pryor, M. R. Thistle, and N. Shirazi, "Text searching on Splash 2", in *IEEE Workshop on FPGAs for Custom Computing Machines*, 1993, pp. 172-177.
- [3] С. Я. Гільгурт, "Метод прискореної кількісної оцінки компонентів реконфігурованих сигнатурних систем кіберзахисту", *Електронне моделювання*, т. 44, № 5, с. 3-24, 2022. doi: 10.15407/emodel.44.05.003.
- [4] S. Dharmapurikar, M. Attig, and J. Lockwood, "Design and Implementation of a String Matching System for Network Intrusion Detection using FPGA-based Bloom Filters", in *All Computer Science and Engineering Research*, Washington University in St. Louis, 2004, WUCSE-2004-12.
- [5] R. Patgiri, S. Nayak, and N. B. Muppalaneni, "Is Bloom Filter a Bad Choice for Security and Privacy? ", *2021 International Conference on Information Networking (ICOIN)*, Jeju Island, Korea (South), pp. 648-653, 2021, doi: 10.1109/ICOIN50884.2021.9333950.
- [6] S. Ya. Hilgurt, "Pattern Handling for Quantifying Hardware Components of Signature-Based Cybersecurity Systems", *Proceedings of the 2nd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2022)*, Ternopil, Ukraine, Nov. 22-24, 2022. – CEUR Workshop Proceedings, vol. 3309, pp. 83-93, 2022, Available online: <https://ceur-ws.org/Vol-3309/paper7.pdf>.
- [7] R. Patgiri, S. Nayak, and S. K. Borgohain, "Hunting the Pertinency of Bloom Filter in Computer Networking and Beyond: A Survey", *Journal of Computer Networks and Communications*, 2019, 2712417, 10 pages, doi: 10.1155/2019/2712417.
- [8] L. Luo, D. Guo, R. T. B. Ma, O. Rottenstreich, and X. Luo, "Optimizing Bloom Filter: Challenges, Solutions, and Comparisons", in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, 2019, pp. 1912-1949, doi: 10.1109/COMST.2018.2889329.
- [9] S. Geravand, and M. Ahmadi, "Bloom filter applications in network security: A state-of-the-art survey", *Computer Networks*, Article vol. 57, no. 18, pp. 4047-4064, Dec 2013, doi: 10.1016/j.comnet.2013.09.003.
- [10] M. A. Owaid, and O. A. Dawood, "A survey in privacy-preserving by bloom filters", in *Proceedings of the 4th international computer sciences and informatics conference (Icsic 2022)*, 28-29 June 2022, Amman, Jordan, 2022, doi: 10.1063/5.0174813.
- [11] M. A. Ferrag, M. Babaghayou, and M. A. Yazici, "Cyber security for fog-based smart grid SCADA systems: Solutions and challenges", *Journal of Information Security and Applications*, Article vol. 52, 2020, Art no. 102500, doi: 10.1016/j.jjsa.2020.102500.
- [12] С. Гільгурт, "Побудова фільтрів Блума реконфігурованими засобами для вирішення задач інформаційної безпеки", *Безпека інформації*, т. 25, № 1, с. 53-58, 2019, doi: 10.18372/2225-5036.25.13594.
- [13] J. Harwayne-Gidansky, D. Stefan, and I. Dalal, "FPGA-based SoC for Real-Time Network Intrusion Detection using Counting Bloom Filters", in *Proceedings of the IEEE Southeastcon 2009*, 2009.
- [14] Y. Chen, A. Kumar, and J. Xu, "A new design of bloom filter for packet inspection speedup", in *IEEE Global Telecommunications Conference (GLOBECOM 07)*, Washington, DC, Nov 26-30 2007, NEW YORK: IEEE, in IEEE Global Telecommunications Conference (Globecom), 2007, pp. 1-5.
- [15] N. S. Artan, K. Sinkar, J. Patel, and H. J. Chao, "Aggregated bloom filters for intrusion detection and prevention hardware", in *IEEE Global Telecommunications Conference (GLOBECOM 07)*, Washington, DC, Nov 26-30 2007, NEW YORK: IEEE, in IEEE Global Telecommunications Conference (Globecom), 2007, pp. 349-354.
- [16] L. Carter, and M. Wegman, "Universal Classes of Hashing Functions", *Computer and System Sciences*, vol. 18, no. 2, pp. 143-154, 1979.
- [17] С. Я. Гільгурт, "Порівняльний аналіз підходів до побудови компонентів реконфігурованих засобів технічного захисту інформації", *Проблеми інформатизації та управління*, т. 2, № 66, с. 17-26, 2021, doi: 10.18372/2073-4751.66.15712.
- [18] S. Y. Hilgurt, A. M. Davydenko, T. V. Matovka, and M. P. Prygara, "Tools for Analyzing Signature-Based Hardware Solutions for Cyber Security Systems", *JCSANDM*, vol. 12, no. 03, pp. 339-366, 2023, doi: 10.13052/jcsm2245-1439.123.5.
- [19] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: A scalable wide-area Web cache sharing protocol", *IEEE ACM Transactions on Networking*, Article vol. 8, no. 3, pp. 281-293, 2000, doi: 10.1109/90.851975.
- [20] С. Я. Гільгурт, О. А. Чемерис, *Реконфігуровні сигнатурні засоби захисту інформації комп'ютерних систем*. Київ, Україна: Академперіодика, 2022, 297 с., doi: 10.15407/akademperiodyka.458.297.

ESTIMATION OF COUNTING BLOOM FILTER HARDWARE COSTS FOR FPGA-BASED CYBERSECURITY SYSTEMS

Serhii Hilgurt

The objective of the study is to research possible solutions and ways of practical implementation regarding the organization of dynamic changing the list of patterns that are matched by the hardware (FPGA-based) Bloom filter in the signature cybersecurity systems, such as network intrusion detection systems, antiviruses, spam filters, etc. Changing the list of patterns means both adding and removing elements from the list of patterns that the signature system searches in the incoming data stream, in particular in the bodies of network traffic packets.

To achieve the goal, the principles of construction and functioning of the Bloom filter, its advantages and disadvantages compared to other matching schemes in terms of known performance indicators are analyzed in the work. The modifications and variations proposed by developers to improve its characteristics during the entire period of use in computer network cybersecurity systems are considered. The features of the hardware implementation of the Bloom filter on the FPGAs are analyzed.

The one of two discovered approaches to solve the problem of dynamic readjustment, that provides greater speed due to changing the hardware structure of the device, namely the Bloom filter with counters, is considered in the work. On the example of one of the hardware scheme modifications, the so-called simplified Bloom filter, a possible option for building its digital structure is considered. To increase the efficiency of the process of developing cybersecurity systems using this scheme, a hardware resource evaluation function has been created, which allows you to find the quantitative characteristics of the cost for synthesizing FPGA-based digital devices without performing the time-consuming procedure of full project compilation. A preliminary comparison of the obtained expression with the evaluation function for the scheme of the simplified Bloom filter without counters was carried out.

Keywords: NIDS, multi-pattern matching, FPGA, Bloom filter with counters, hardware costs.

REFERENCES

- [1] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors", *Communications of the ACM*, Article vol. 13, no. 7, pp. 422-426, 1970, doi: 10.1145/362686.362692.
- [2] D. V. Pryor, M. R. Thistle, and N. Shirazi, "Text searching on Splash 2", in *IEEE Workshop on FPGAs for Custom Computing Machines*, 1993, pp. 172-177.
- [3] S. Ya. Hilgurt, "Accelerated Quantitative Evaluation of Components of FPGA-Based Security Systems", *Electronic Modeling*, vol. 44, no. 5, pp. 3-24, 2022, doi: 10.15407/emodel.44.05.003. (In Ukrainian).
- [4] S. Dharmapurikar, M. Attig, and J. Lockwood, "Design and Implementation of a String Matching System for Network Intrusion Detection using FPGA-based Bloom Filters", in *All Computer Science and Engineering Research*, Washington University in St. Louis, 2004, WUCSE-2004-12.
- [5] R. Patgiri, S. Nayak, and N. B. Muppalaneni, "Is Bloom Filter a Bad Choice for Security and Privacy? ", *2021 International Conference on Information Networking (ICOIN)*, Jeju Island, Korea (South), pp. 648-653, 2021, doi: 10.1109/ICOIN50884.2021.9333950.
- [6] S. Ya. Hilgurt, "Pattern Handling for Quantifying Hardware Components of Signature-Based Cybersecurity Systems", *Proceedings of the 2nd International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP 2022)*, Ternopil, Ukraine, Nov. 22-24, 2022. – CEUR Workshop Proceedings, vol. 3309, pp. 83-93, 2022, Available online: <https://ceur-ws.org/Vol-3309/paper7.pdf>.
- [7] R. Patgiri, S. Nayak, and S. K. Borgohain, "Hunting the Pertinency of Bloom Filter in Computer Networking and Beyond: A Survey", *Journal of Computer Networks and Communications*, 2019, 2712417, 10 pages, doi: 10.1155/2019/2712417.
- [8] L. Luo, D. Guo, R. T. B. Ma, O. Rottenstreich, and X. Luo, "Optimizing Bloom Filter: Challenges, Solutions, and Comparisons", in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, 2019, pp. 1912-1949, doi: 10.1109/COMST.2018.2889329.
- [9] S. Geravand, and M. Ahmadi, "Bloom filter applications in network security: A state-of-the-art survey", *Computer Networks*, Article vol. 57, no. 18, pp. 4047-4064, Dec 2013, doi: 10.1016/j.comnet.2013.09.003.
- [10] M. A. Owaid, and O. A. Dawood, "A survey in privacy-preserving by bloom filters", in *Proceedings of the 4th international computer sciences and informatics conference (ICSIC 2022)*, 28-29 June 2022, Amman, Jordan, 2022, doi: 10.1063/5.0174813.
- [11] M. A. Ferrag, M. Babaghayou, and M. A. Yazici, "Cyber security for fog-based smart grid SCADA systems: Solutions and challenges", *Journal of Information Security and Applications*, Article vol. 52, 2020, Art no. 102500, doi: 10.1016/j.jisa.2020.102500.
- [12] S. Hilgurt, "Constructing Bloom filters by reconfigurable means for solving information security tasks", *Ukrainian Scientific Journal of Information Security*, vol. 25, no. 1, pp. 53-58, 2019, doi: 10.18372/2225-5036.25.13594. (In Ukrainian).
- [13] J. Harwayne-Gidansky, D. Stefan, and I. Dalal, "FPGA-based SoC for Real-Time Network Intrusion Detection using Counting Bloom Filters", in *Proceedings of the IEEE Southeastcon 2009*, 2009.
- [14] Y. Chen, A. Kumar, and J. Xu, "A new design of bloom filter for packet inspection speedup", in *IEEE Global Telecommunications Conference (GLOBECOM 07)*, Washington, DC, Nov 26-30 2007, NEW YORK: IEEE, in IEEE Global Telecommunications Conference (Globecom), 2007, pp. 1-5.
- [15] N. S. Artan, K. Sinkar, J. Patel, and H. J. Chao, "Aggregated bloom filters for intrusion detection and prevention hardware", in *IEEE Global Telecommunications Conference (GLOBECOM 07)*, Washington, DC, Nov 26-30 2007, NEW YORK: IEEE, in IEEE Global Telecommunications Conference (Globecom), 2007, pp. 349-354.
- [16] L. Carter, and M. Wegman, "Universal Classes of Hashing Functions", *Computer and System Sciences*, vol. 18, no. 2, pp. 143-154, 1979.
- [17] S. Ya. Hilgurt, "Comparative analysis of approaches to the building of reconfigurable security tools components", *Problems of informatization and management*, vol. 2, no. 66, pp. 17-26, 2021, doi: 10.18372/2073-4751.66.15712.
- [18] S. Y. Hilgurt, A. M. Davydenko, T. V. Matovka, and M. P. Prygara, "Tools for Analyzing Signature-Based Hardware Solutions for Cyber Security Systems", *JCSANDM*, vol. 12, no. 03, pp. 339-366, 2023, doi: 10.13052/jcsm2245-1439.123.5.
- [19] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: A scalable wide-area Web cache sharing protocol", *IEEE ACM Transactions on Networking*, Article vol. 8, no. 3, pp. 281-293, 2000, doi: 10.1109/90.851975.
- [20] S. Ya. Hilgurt, O. A. Chemerys, *Reconfigurable signature-based information security tools of computer systems*. Kyiv, Ukraine: Akadempriodyka, 2022, p. 297, doi: 10.15407/akademperiodyka.458.297. (In Ukrainian).