

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

ІЛЬІН МАКСИМ ОЛЕКСАНДРОВИЧ

УДК 004.9:004.4

ДИСЕРТАЦІЯ

АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ ТА
ПРОГНОЗУВАННЯ ЕКОЛОГІЧНИХ ЗАГРОЗ НА ОСНОВІ АНАЛІЗУ
ГЕТЕРОГЕННИХ ПОТОКОВИХ ДАНИХ

121 – Інженерія програмного забезпечення

12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

_____ М.О. Ільїн

Науковий керівник: Олещенко Любов Михайлівна, к.т.н., доцент

Київ – 2026

АНОТАЦІЯ

Ільїн М. О. Алгоритмічне та програмне забезпечення для виявлення та прогнозування екологічних загроз на основі аналізу гетерогенних потокових даних – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення. – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, 2026.

Оброблення та аналіз гетерогенних потокових даних екологічного моніторингу є складною задачею через різноманітність форматів, асинхронність надходження, багатомовність та змінюваність форм слів у текстах, невідповідність часових баз, різні одиниці виміру, пропуски та шуми в рядах сенсорних даних, а також вимоги до часу відгуку обчислень. Стандартні програмні конвеєри з відсутністю єдиної просторово-часової основи не забезпечують стабільного геоприв'язування та узгоджених агрегатів, що призводить до помилок у локалізації подій, дублювання супутникових виявлень та нестабільної швидкодії.

У зв'язку з цим виникає потреба в інтегрованому підході, який поєднує нормалізацію форматів та часових поясів, геокодування зі збереженням коректності для різних словоформ, просторово-часове вирівнювання на шестикутній сітці та двошарову агрегацію добових індикаторів ризику.

Упродовж останніх років такі підходи успішно застосовуються у задачах кризового моніторингу та геоаналітики, що підтверджує доцільність створення уніфікованого програмного засобу для високопродуктивного оброблення великих потоків даних та адаптації до мінливих умов.

Метою дисертаційної роботи є підвищення швидкодії оброблення гетерогенних потокових даних, точності геоприв'язування та забезпечення відтворюваності просторово-часових показників екологічних ризиків шляхом створення алгоритмічного та програмного забезпечення для реалізації конвеєра уніфікації, геокодування та агрегації гетерогенних потокових даних.

У **першому розділі** дисертації проведено аналіз сучасного стану досліджень з програмної аналітики потокових даних для моніторингу та прогнозування екологічних загроз. Розглянуто основні підходи до оброблення гетерогенних екологічних даних, зокрема, OSINT-повідомлень, сенсорних вимірювань та геопросторової інформації. У межах аналізу програмних рішень для моніторингу екологічних загроз було розглянуто як міжнародні програмні системи (ACLED, GDELT, Liveuamap), так і національні системи моніторингу (EcoZagroza, Ecodozor, SaveEcoBot), виявлено їх переваги та обмеження, зокрема, у сфері інтеграції мультимодальних OSINT-джерел, оперативного геокодування та прогнозної аналітики. Аналіз наявних програмних систем моніторингу екологічних загроз підтвердив наукову та практичну актуальність розроблення алгоритмічних та програмних засобів для аналізу та прогнозування екологічних загроз на основі гетерогенних потокових даних, що важливо в умовах воєнних дій та підвищених екологічних ризиків, коли оперативна інформація та її точна інтерпретація можуть рятувати життя. Окрему увагу приділено архітектурним шаблонам сучасних програмних систем та їх структурним обмеженням.

На основі проведеного аналізу було сформульовано функціональні та нефункціональні вимоги до програмного забезпечення.

У **другому розділі** розроблено методи проєктування програмного забезпечення для аналізу гетерогенних потокових даних, орієнтовані на підтримку аналізу даних в реальному часі та прийняття рішень в умовах невизначеності. Запропоновані методи збору та попередньої обробки потокових даних з різних джерел, а також методи розпізнавання подій на основі оброблення природної мови для аналізу текстових OSINT-повідомлень.

Розроблено метод геокодування зі збереженням коректності для різних словоформ у багатомовних повідомленнях. Запропоновано метод багатоджерельної перевірки та фільтрації якості подій, а також ймовірнісну геопросторову модель прийняття рішень, яка інтегрує результати аналізу гетерогенних потоків даних у єдиному формалізованому просторі.

Запропонований архітектурний шаблон H3-DSMV призначений для обробки геопросторових потокових даних та поєднує принцип єдиного джерела істини, ієрархічну узгодженість просторових агрегацій та мінімальну інфраструктурну складність. Фіксація подій на дрібнішій сітці H3 з автоматичним формуванням матеріалізованих подань на вищому рівні роздільності забезпечує коректну та швидку аналітику без дублювання даних та втрати точності. Основними перевагами підходу є висока просторово-часова точність, детермінована агрегація, автоматична синхронізація подань, зменшення накладних витрат на інфраструктуру та підвищена надійність і масштабованість системи в задачах реального часу.

У **третьому розділі** розроблено архітектуру та здійснено програмну реалізацію модульної системи оброблення потокових даних на основі мікропакетного конвеєра. Запропоновано модульну архітектуру з використанням H3-агрегації та матеріалізованих подань, що забезпечує просторово-часову агрегацію гетерогенних потоків даних. Реалізовано модулі оброблення мовних компонент, геопросторової підсистеми з підтримкою морфологічно коректного геокодування, а також спеціалізований модуль аналізу потокових даних про стан довкілля. Інтегровано прогнозування на основі LSTM-моделі для аналізу часових залежностей екологічних показників.

У **четвертому розділі** проведено експериментальні дослідження запропонованого програмного забезпечення. Описано експериментальні сценарії та використані джерела OSINT-подій, сформульовано методологію оцінювання та обрано відповідні метрики якості. Проаналізовано результати геопросторового аналізу та прогнозування екологічних загроз, а також виконано порівняння запропонованої системи з наявними програмними рішеннями.

Узагальнюючи результати проведених експериментальних досліджень, можна стверджувати, що розроблене програмне забезпечення демонструє системну перевагу над наявними рішеннями екологічного моніторингу завдяки

поєднанню потокового режиму оброблення, багатоджерельної верифікації подій та просторово-часової агрегації на основі шестикутної сітки НЗ.

На відміну від систем Safecast та SaveEcoBot, які орієнтовані відповідно на сенсорні або OSINT-дані та не забезпечують повноцінної автоматичної валідації, запропоноване програмне забезпечення інтегрує текстові повідомлення, супутникові спостереження та фізичні вимірювання, що істотно зменшує похибки та кількість хибних спрацювань. Порівняння з Copernicus EMS показало, що хоча супутникові офлайн-системи і забезпечують дещо вищу просторову точність, проте вони не придатні для оперативного реагування через значні часові затримки.

Експериментальні результати підтвердили, що розроблене програмне рішення забезпечує зіставну з еталонними системами точність геопросторової локалізації подій у потоковому режимі (85% коректних прив'язувань) при суттєво меншому середньому часі отримання результату (близько 6 хв). Запровадження інтегрального показника, який поєднує точність та оперативність, дозволило обґрунтовано показати, що саме розроблене програмне забезпечення є найбільш збалансованим для задач оперативного екологічного моніторингу, перевершуючи альтернативні системи за сукупною ефективністю. Додатково було встановлено, що використання багатоджерельної перевірки та ймовірнісної моделі прийняття рішень дозволяє істотно зменшити похибку агрегованої оцінки ризику за умов неповних або зашумлених даних (до 8%), а також знизити частоту хибних спрацювань до 4%. Це є критично важливим для практичних сценаріїв, пов'язаних з воєнними та техногенними загрозами, де дані часто є фрагментованими та несинхронізованими. Оцінка експлуатаційних характеристик показала, що розроблене програмне забезпечення має значно меншу затримку оброблення та знижене ресурсоспоживання порівняно з аналізованими аналогами, що підтверджує його придатність для розгортання в умовах обмежених обчислювальних та енергетичних ресурсів. У сукупності отримані результати підтверджують наукову обґрунтованість та практичну доцільність запропонованих методів та програмної реалізації для систем оперативного моніторингу стану

довкілля і раннього попередження про екологічні та техногенні загрози. Також було розглянуто переваги та особливості практичного впровадження розробленого програмного забезпечення.

Наукова новизна одержаних результатів полягає в наступному.

Уперше розроблено метод геокодування зі збереженням коректності для різних словоформ багатомовних OSINT-повідомлень, характерними рисами якого є поєднання морфологічної нормалізації змінюваних форм, каскаду точного, нечіткого та фонетичного зіставлення з реєстром географічних назв та усунення контекстуальної неоднозначності в локальному оточенні НЗ, що, на відміну від існуючих методів геокодування, орієнтованих на англomовні тексти, дає змогу підвищити частку коректних прив'язувань топонімів в середньому на 21% та скоротити середній час пошуку координат в середньому на 51 %.

Уперше розроблено ймовірнісну геопросторову програмну модель прийняття рішень на основі аналізу гетерогенних потокових даних, характерними рисами якої є інтеграція результатів геокодування, багатоджерельної перевірки та доменної агрегації на шестикутній сітці НЗ з підтримкою сценарних обмежень, що, на відміну від відомих моделей, дає змогу оцінювати небезпечні зони, формувати рекомендації щодо планування евакуації та розміщення нових укриттів з кореляцією 0,78 з офіційними повітряними тривогами.

Уперше розроблено архітектуру уніфікованого мікропакетного конвеєра гармонізації гетерогенних потоків даних, характерними рисами якого є детерміноване перетворення подій до єдиної часової бази, нормалізація одиниць вимірювання та уніфікація схем подій у межах одного процесу, що, на відміну від стандартних багатоетапних ETL-конвеєрів, дає змогу скоротити затримку уніфікації в середньому на 80%.

Уперше розроблено архітектурний шаблон НЗ-DSMV (Dual Scale Materialized Views) для аналізу гетерогенних потоків даних, характерними рисами якого є фіксація подій на дрібнішій сітці r10 з автоматичною побудовою та підтримкою матеріалізованих подань на вищій сітці r7, що, на відміну від

розподілених архітектур з окремими компонентами агрегації, дає змогу в 5 разів скоротити середній час відповіді системи.

Запропоновані методи та програмне рішення забезпечують безперервний моніторинг екологічних ризиків, підтвердження подій за кількома джерелами, формування добових просторово-часових індикаторів для підтримки прийняття рішень органами цивільного захисту та екологічного нагляду, а також застосування на обмежених обчислювальних ресурсах у режимі автономної роботи.

Удосконалені процедури доменної агрегації ризику для гетерогенних поточкових даних екологічного моніторингу, на відміну від існуючих процедур, поєднують FRP-агрегацію, метеокорекцію для врахування вітру та метеофакторів, нормування радіаційних показників та єдину часову базу.

Запропонований метод інтеграції LSTM-прогнозування у єдиний просторово-часовий конвеєр екологічного моніторингу, на відміну від реактивних методів моніторингу, забезпечує перехід до проактивного режиму підтримки рішень на основі короткострокових прогнозів.

Основні наукові результати дисертаційної роботи опубліковано у 11 наукових працях, зокрема, у 6 наукових статтях, опублікованих у фахових виданнях, включених до переліку наукових фахових видань України з присвоєнням категорії «Б», та у 5 матеріалах науково-технічних конференцій.

Ключові слова: інженерія програмного забезпечення, архітектура програмного забезпечення, системи моніторингу, візуалізація даних в реальному часі, алгоритм, попередня обробка даних, гетерогенні поточкові дані, прогнозування, нейронні мережі, екологічні ризики, виявлення, прийняття рішень, програмування.

SUMMARY

Ilin M. O. Algorithmic and software support for detection and forecasting of environmental threats based on the analysis of heterogeneous streaming data – a qualifying scholarly work (manuscript).

Dissertation for the degree of Doctor of Philosophy in the field of knowledge 12 Information Technologies, specialty 121 Software Engineering. – National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, 2026.

Processing and analysis of heterogeneous streaming data in environmental monitoring constitute a complex task due to the diversity of data formats, asynchronous data arrival, multilinguality and morphological variability of textual content, inconsistencies in temporal reference systems, heterogeneous measurement units, missing values and noise in sensor time series, as well as strict requirements for low computational latency. Conventional software pipelines that lack a unified spatio-temporal foundation fail to provide stable georeferencing and consistent aggregation, which leads to errors in event localization, duplication of satellite detections, and unstable system performance. In this context, there is a need for an integrated approach that combines data format and time zone normalization, morphology-aware geocoding, spatio-temporal alignment using a hexagonal grid, and multi-level aggregation of daily risk indicators.

In recent years, such approaches have been successfully applied in crisis monitoring and geospatial analytics tasks, confirming the feasibility and relevance of developing a unified solution for high-performance processing of large-scale data streams and adaptation to dynamically changing conditions.

The aim of the dissertation is improving the processing speed of heterogeneous streaming data, increase the accuracy of geospatial referencing, and ensure the reproducibility of spatio-temporal indicators of environmental risks by developing algorithmic and software solutions for implementing a pipeline for unification, geocoding, and aggregation of heterogeneous streaming data.

In the **first chapter** of the dissertation, an analysis of the current state of research in the field of software analytics of streaming data for environmental threat monitoring and forecasting is conducted. The chapter examines existing approaches to the processing of heterogeneous environmental data, including OSINT reports, sensor measurements, and geospatial information. Within the framework of the analysis of algorithmic and software solutions for monitoring environmental threats, both international software systems (ACLED, GDELT, Liveuamap) and national monitoring systems (EcoZagroza, Ecodozor, SaveEcoBot) were considered, their advantages and limitations were identified, in particular, in the field of integration of multimodal OSINT sources, operational geocoding and predictive analytics.

The analysis of existing software systems for monitoring environmental threats confirmed the scientific and practical relevance of developing algorithmic and software tools for analyzing and predicting environmental threats based on heterogeneous streaming data, which is important in conditions of military operations and increased environmental risks. Particular attention is paid to architectural patterns employed in modern systems and the structural constraints associated with them. Based on the conducted analysis, functional and non-functional requirements.

In the **second chapter**, methods for designing a software for the analysis of heterogeneous streaming data are developed, with an emphasis on real-time operation and decision-making under uncertainty. Enhanced methods for data acquisition and preprocessing from multiple heterogeneous sources are proposed, along with event recognition methods based on natural language processing for analyzing textual OSINT reports. A geocoding method that preserves correctness across different word forms in multilingual messages is developed, ensuring improved accuracy of spatial event localization. Furthermore, a multi-source event verification and quality filtering method is proposed, as well as a probabilistic geospatial decision-making model that integrates the results of heterogeneous data stream analysis within a unified formalized framework. The proposed H3-DSMV architectural template is designed for processing geospatial streaming data and combines the principle of a single source of truth,

hierarchical consistency of spatial aggregations and minimal infrastructure complexity. Event capture on a finer H3 grid with automatic generation of materialized views at a higher resolution level provides correct and fast analytics without data duplication and loss of accuracy. The main advantages of the approach are high spatiotemporal accuracy, deterministic aggregation, automatic synchronization of views, reduced infrastructure overhead and increased reliability and scalability of the system in real-time tasks.

In the **third chapter**, the architecture and software implementation of a modular streaming data processing system based on a micro-batch pipeline are presented.

A generalized modular architecture employing H3 spatial aggregation and materialized views is proposed, enabling efficient spatiotemporal aggregation of heterogeneous data streams. Software modules for processing linguistic components, a geospatial subsystem utilizing morphologically robust geocoding, and a specialized module for processing environmental state streaming data are implemented. Output data formats and mechanisms for ensuring result reproducibility are defined. In addition, algorithmic support for time-series forecasting based on LSTM models is developed and integrated to capture temporal dependencies in environmental indicators.

In the **fourth chapter**, experimental studies are conducted to evaluate the effectiveness of the proposed software. The experimental scenarios and OSINT event sources are described, and an evaluation methodology along with appropriate performance metrics is defined.

The results of geospatial analysis and environmental threat forecasting are analyzed, and a comparative assessment with existing systems is performed. Summarizing the results of the conducted experimental studies, it can be stated that the developed software demonstrates a systematic advantage over existing environmental monitoring solutions due to the combination of streaming data processing, multi-source event verification, and spatio-temporal aggregation based on the H3 hexagonal grid.

A comparison with the Copernicus EMS system showed that although satellite-based offline solutions provide slightly higher spatial accuracy, they are not suitable for operational response due to substantial time delays.

The experimental results confirmed that the developed software achieves spatial localization accuracy comparable to reference systems in a streaming mode (85% correct geospatial bindings) while providing a significantly shorter average time to obtain results (approximately 6 minutes). The introduction of an integral indicator combining accuracy and responsiveness made it possible to convincingly demonstrate that the proposed solution is the most balanced for operational environmental monitoring tasks, outperforming alternative systems in overall effectiveness.

Additionally, it was established that the use of multi-source verification and a probabilistic decision-making model significantly reduces the error of aggregated risk assessment under conditions of incomplete or noisy data (down to 8%) and lowers the false alarm rate to 4%. The evaluation of operational characteristics showed that the developed software has substantially lower processing latency and reduced resource consumption compared to the analyzed analogues, confirming its suitability for deployment under limited computational and energy resources.

Taken together, the obtained results confirm the scientific validity and practical feasibility of the proposed methods and software implementation for operational environmental monitoring systems and early warning of environmental and technogenic hazards. The advantages and features of practical implementation of the software are considered.

The **scientific novelty** of the results obtained is as follows.

For the first time, a geocoding method has been developed that preserves correctness for different word forms in multilingual OSINT messages; its distinctive features include the combination of morphological normalization of inflected forms, a cascade of exact, fuzzy, and phonetic matching with a registry of geographic names, and the elimination of contextual ambiguity within the local H3 neighborhood, which unlike existing geocoding methods oriented toward English-language texts makes it possible to

increase the share of correctly linked toponyms by an average of 21% and reduce the average coordinate lookup time by approximately 51%.

For the first time, a probabilistic geospatial software decision-making model based on the analysis of heterogeneous streaming data has been developed; its distinctive features include the integration of geocoding results, multi-source event verification, and domain-specific aggregation on the H3 hexagonal grid with support for scenario constraints, which unlike existing models enables the assessment of hazardous zones and the generation of recommendations for evacuation planning and the placement of new shelters with a correlation of 0.78 with official air-raid alerts.

For the first time, an architecture of a unified micro-batch pipeline for harmonizing heterogeneous data streams has been developed; its distinctive features include deterministic transformation of events to a unified time base, normalization of measurement units, and unification of event schemas within a single process, which unlike multi-stage ETL pipelines reduces unification latency by an average of 80%.

For the first time, the H3-DSMV (Dual Scale Materialized Views) architectural pattern for analyzing heterogeneous data streams has been developed; its distinctive features include fixing events on a finer r10 grid with automatic construction and maintenance of materialized views on a higher r7 grid, which unlike distributed architectures with separate aggregation components reduces the average system response time by a factor of five.

The proposed methods and software solution provide continuous monitoring of environmental risks, confirmation of events from multiple sources, formation of daily spatiotemporal indicators to support decision-making by civil protection and environmental supervision bodies, as well as application on limited computing resources in the autonomous mode.

Improved domain risk aggregation procedures for heterogeneous streaming data of environmental monitoring, unlike existing procedures, combine FRP aggregation, meteorological correction to take into account wind and meteorological factors, normalization of radiation indicators and a single time base.

The proposed method of integrating LSTM forecasting into a single spatiotemporal pipeline of environmental monitoring, unlike reactive monitoring methods, provides a transition to a proactive mode of decision support based on short-term forecasts.

The main scientific results of the dissertation have been published in 11 scientific works, including 6 scientific articles published in professional journals included in the list of scientific journals of Ukraine in category “Б”, as well as 5 materials of scientific and technical conferences.

Keywords: software engineering, software architecture, monitoring systems, real-time data visualization, algorithm, data preprocessing, heterogeneous streaming data, prediction, neural networks, environmental risks, detection, decision making, programming.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Статті у фахових виданнях, включених до переліку наукових фахових видань України з присвоєнням категорії «Б»:

1. Ilin M. O., Oleshchenko L. M. Current state and development prospects of heterogeneous streaming data processing methods. Information Technology: Computer Science, Software Engineering and Cyber Security, Вип. 1, 2023. С. 100-106.

Внесок Ільїна М.О.: виконання аналізу підходів до програмної обробки гетерогенних потокових даних, узагальнення вимог до уніфікації схем, форматів та часових узгоджень даних, підготовка тексту та висновків статті.

Внесок Олещенко Л.М.: постановка наукової проблеми, формулювання цілей та завдань дослідження.

2. Oleshchenko L. M., Ilin M. O. Software analysis of radiation air pollution streaming data. Вісник Херсонського національного технічного університету. № 2(85), 2023. С. 187-195.

Внесок Олещенко Л.М.: постановка задачі, визначення методичних підходів та критеріїв оцінювання якості даних та отриманих результатів.

Внесок Ільїна М.О.: збір та підготовка відкритих часових рядів даних радіаційного моніторингу, розроблення програмних методів виявлення аномалій та збоїв датчиків, аналіз результатів, підготовка візуалізацій та висновків.

3. Ilin M. O., Oleshchenko L. M. Enhancing distributed stream processing through artificial intelligence: a review and proposal for LSTM integration. Інфокомунікаційні та комп'ютерні технології. Том 2. № 06, 2023. С. 66-73.

Внесок Ільїна М.О.: аналіз архітектур програмних систем розподіленої потокової обробки даних, обґрунтування сценаріїв інтеграції LSTM у CEP-контури, підготовка моделі взаємодії компонентів, опис методології використання моделі LSTM для прогнозної аналітики.

Внесок Олещенко Л.М.: формулювання дослідницьких завдань та концептуальних обмежень дослідження.

4. Олещенко Л. М., Ільїн М. О. Програмний метод аналізу поточкових даних соціальних мереж за допомогою гібридної моделі нейронної мережі. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 36 (75). № 1, 2025. С. 163-170.

Внесок Олещенко Л.М.: постановка задачі подієвого аналізу соціальних потоків, науковий контроль отриманих результатів.

Внесок Ільїна М.О.: розроблення архітектури, програмна реалізація гібридної моделі та експериментального контуру (симульованого потокового середовища), підготовка даних, проведення експериментів, порівняльний аналіз з базовою моделлю LSTM, інтерпретація результатів.

5. Ilin M. O., Oleshchenko L. M. Development of a modular data unification pipeline for real-time environmental threat analytics. Вісник Херсонського національного технічного університету. Том 2 № 3(94) 2025. С. 211-217.

Внесок Ільїна М.О.: проєктування та програмна реалізація модульного конвеєра уніфікації, реалізація НЗ-агрегації та механізмів узгодження добових шарів, інтеграція джерел даних та експериментальна перевірка.

Внесок Олещенко Л.М.: постановка задачі побудови модульного конвеєра та формулювання вимог до узгодження часових шарів та агрегацій.

6. Ilin M.O., Oleshchenko L.M. Software-driven data analysis of propaganda narratives and social media engagement. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 36 (75). № 4, 2025. С. 155-162.

Внесок Ільїна М.О.: побудова та наповнення високонавантаженої БД та конвеєра оброблення даних, реалізація процедур вилучення та нормалізації сутностей, інженерія інтерпретованих ознак, виконання аналізу та формулювання робочих гіпотез, підготовка матеріалів та висновків статті.

Внесок Олещенко Л.М.: постановка завдань дослідження, науковий контроль отриманих результатів.

Публікації у матеріалах наукових конференцій:

7. Ilin M. O., Oleshchenko L. M. RNN implementation for streaming data analysis and prediction. International scientific conference «The latest science and technology achievements and their significance for society». December 6–7, 2023. Częstochowa, the Republic of Poland, 2023. P. 6-10.

Внесок Ільїна М.О.: підготовка даних, програмна реалізація RNN-підходу для потокового аналізу та прогнозування, проведення експериментів, аналіз отриманих результатів та підготовка висновків.

Внесок Олещенко Л.М.: постановка задачі потокового прогнозування та критеріїв оцінювання результатів дослідження.

8. Oleshchenko L., Ilin M. Artificial intelligence methods integrating in streaming data processing systems. X International scientific and practical conference «Modern Trends in the Development of Scientific Space» February 14-16, 2024. Dresden, Germany, 2024. P. 96-101.

Внесок Олещенко Л.М.: формування теоретичних засад інтеграції ШІ в СЕР-системи, постановка дослідницьких завдань.

Внесок Ільїна М.О.: систематизація інтеграційних сценаріїв, підготовка прикладних рекомендацій щодо суміщення аналітики та СЕР, узагальнення переваг та обмежень підходів, підготовка матеріалів для публікації.

9. Ільїн М.О., Олещенко Л.М. Метод та програмне забезпечення для виявлення подій у реальному часі та аналізу трендів з використанням HybridLSTM. Прикладна математика та комп'ютинг. ПМК, 2024: Сімнадцята наук. конф. магістрантів та аспірантів, Київ, 20-22 лист. 2024 р. Київ: «Просвіта», 2024. С. 201-205.

Внесок Ільїна М.О.: програмна реалізація запропонованого методу, підготовка даних, проведення дослідження, підготовка висновків.

Внесок Олещенко Л.М.: постановка задачі та науковий контроль проведення дослідження та інтерпретації отриманих результатів.

10. Ilin M., Oleshchenko L. Software pipeline development for OSINT-based geocoding of missile and drone strikes in Ukraine. International Scientific Conference «Science, Technology, and Society in the 21st Century», August 6, 2025, Amsterdam, Netherlands. P. 116-119.

Внесок Ільїна М.О.: проектування та програмна реалізація конвеєра, побудова локального реєстру географічних назв, реалізація морфологічної нормалізації та правил розв'язання неоднозначностей, інтеграція процедур валідації, тестування на прикладних даних та підготовка висновків.

Внесок Олещенко Л.М.: постановка задачі та контроль якості отриманих результатів дослідження.

11. Ilin M., Oleshchenko L. Real-time radiation anomaly mapping using Kp/GOES X-ray correction and H3 spatial aggregation. International Scientific Conference «Trends, Issues, and Challenges in Modern Science», September 5, 2025, Cambridge, UK. P. 86-89.

Внесок Ільїна М.О.: розроблення та програмна реалізація конвеєра, реалізація H3-агрегації та побудови карт аномалій, експериментальна перевірка та аналіз результатів, підготовка матеріалів публікації.

Внесок Олещенко Л.М.: постановка задачі, визначення засад корекції за геокосмічними індикаторами та вимог до просторової агрегації.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	20
ВСТУП.....	22
РОЗДІЛ 1. АНАЛІЗ ПРОГРАМНИХ СИСТЕМ МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ ЕКОЛОГІЧНИХ ЗАГРОЗ.....	29
1.1. Стан досліджень з програмної аналітики потокових даних для прогнозування екологічних загроз.....	29
1.2. Порівняльний аналіз програмних рішень для моніторингу екологічних загроз та їх обмеження.....	34
1.3. Методи збору та попередньої обробки гетерогенних потокових даних.....	42
1.4. Методи розпізнавання подій на основі NLP.....	44
1.5. Аналіз вимог до програмного забезпечення.....	48
1.6. Висновки до розділу 1.....	52
РОЗДІЛ 2. МЕТОДИ ТА ПРОГРАМНІ ЗАСОБИ АНАЛІЗУ ГЕТЕРОГЕННИХ ПОТОКОВИХ ДАНИХ ДЛЯ ВИЯВЛЕННЯ І ПРОГНОЗУВАННЯ ЕКОЛОГІЧНИХ ЗАГРОЗ.....	54
2.1. Метод геокодування зі збереженням коректності для різних словоформ багатомовних OSINT-повідомлень.....	54
2.2. Метод багатоджерельної перевірки та фільтрації якості подій.....	59
2.3. Ймовірнісна геопросторова модель прийняття рішень на базі гетерогенних потоків даних.....	65
2.4. Процедури доменної агрегації ризику для гетерогенних потокових даних.....	71
2.5. Вагові коефіцієнти системи.....	73
2.6. Архітектурний шаблон H3-Dual Scale Materialized Views.....	77
2.7. Висновки до розділу 2.....	83

РОЗДІЛ 3. АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ МОДУЛЬНОЇ ПРОГРАМНОЇ СИСТЕМИ ОБРОБЛЕННЯ ПОТОКОВИХ ДАНИХ НА ОСНОВІ МІКРОПАКЕТНОГО КОНВЕЄРА.....	86
3.1. Узагальнена модульна архітектура мікропакетного конвеєра з НЗ-агрегацією та матеріалізованими поданнями для оброблення гетерогенних поточкових даних.....	86
3.2. Програмна реалізація модуля оброблення мовних компонент.....	100
3.3. Програмна реалізація геопросторової підсистеми з використанням геокодування зі збереженням коректності для різних словоформ.....	113
3.4. Модуль оброблення поточкових даних про стан довкілля.....	121
3.5. Формати вихідних даних та забезпечення відтворюваності результатів...	129
3.6. Алгоритмічне забезпечення та інтеграція LSTM-прогнозування.....	136
3.7. Висновки до розділу 3.....	143
РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	145
4.1. Експериментальні сценарії та джерела OSINT-подій.....	147
4.2. Методологія оцінювання та метрики.....	152
4.3. Результати геопросторового аналізу.....	163
4.4. Порівняння з наявними програмними системами.....	170
4.5. Особливості впровадження програмного забезпечення.....	181
4.6. Висновки до розділу 4.....	187
ВИСНОВКИ.....	189
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	192
ДОДАТКИ.....	205

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API – програмний інтерфейс застосунків (Application Programming Interface).

CER – оброблення складних подій (Complex Event Processing).

CRS – система координат (Coordinate Reference System).

CSV – текстовий формат табличних даних з роздільниками (Comma-separated Values).

ES – журналювання подій (Event Sourcing).

ETL – вилучення, перетворення та завантаження даних (Extract, transform, load).

EURDEP – європейська мережа обміну даними про радіаційний моніторинг (EUropean Radiological Data Exchange Platform).

F1 – F1-міра (гармонійне середнє precision та recall).

FIRMS – Інформація про пожежі для системи управління ресурсами (Fire Information for Resource Management System), створено NASA.

FRP – радіаційна потужність пожеж (Fire Radiative Power).

FRP_sum – добова сума FRP (узгоджена метрика агрегації).

GDPR – Загальний регламент про захист даних (General Data Protection Regulation).

GIS – геоінформаційна система (Geographic Information System).

H3 – ієрархічний шестикутний просторовий індекс.

H3-DSMV – двошарова схема матеріалізованих зображень на H3 (Dual-Scale Materialized Views).

ID – ідентифікатор (Identifier).

JSON – текстовий формат обміну даними (JavaScript Object Notation).

LLM – велика мовна модель (Large Language Model).

LSTM – довга короткочасна пам'ять (Long Short-Term Memory).

MCLP – задача максимального покриття (Maximal Coverage Location Problem).

ML – машинне навчання (Machine Learning).

MV – матеріалізоване зображення (Materialized View).

NLP – оброблення природної мови (Natural Language Processing).

NER – розпізнавання іменованих сутностей (Named Entity Recognition).

OSINT – розвідка на основі відкритих джерел (Open-Source Intelligence).

RNN – рекурентна нейронна мережа (Recurrent Neural Network).

RPS – операцій за секунду (подій/рядків на секунду).

r7, r10 – рівні роздільності сітки НЗ.

S-BERT – семантичне кодування речень (Sentence-BERT).

SEB – SaveEcoBot (система екологічного моніторингу).

UTC – всесвітній координований час.

UHP – уніфікований конвеєр гармонізації (Unified Harmonization Pipeline).

WGS84 – геодезична референс-система World Geodetic System 1984.

XLM-R – XLM-RoBERTa (багатомовна трансформерна модель).

mDeBERTa-v3 – багатомовна модель DeBERTa v3.

ГІС – геоінформаційна система.

ШІ – штучний інтелект.

ВСТУП

Актуальність роботи. В сучасних умовах зростання техногенного навантаження та кліматичних змін проблема своєчасного виявлення та прогнозування екологічних загроз набуває стратегічного значення. Попри значний прогрес у сфері моніторингу довкілля спостерігається відсутність комплексного програмного забезпечення, здатного здійснювати оброблення гетерогенних потокових даних у реальному часі. Наявні рішення орієнтовані на аналіз окремих типів даних (метеорологічних, радіаційних, супутникових тощо) або працюють у відкладеному режимі, не забезпечуючи динамічної інтеграції різнорідних джерел. Відсутність уніфікованих алгоритмічних підходів та модульних архітектур, здатних до масштабування та адаптації під змінні умови, створює розрив між накопиченням даних та формуванням оперативних управлінських рішень. Це зумовлює потребу у створенні алгоритмічного та програмного забезпечення, орієнтованого на виявлення, оцінювання та прогнозування екологічних ризиків у режимі реального часу, з урахуванням високої варіативності, нестабільності та неоднорідності вхідних потоків інформації. Оброблення гетерогенних потокових даних екологічного моніторингу є складним завданням через різнорідність форматів та часових баз, багатомовність та змінюваність форм слів у текстах, асинхронність надходження сигналів, пропуски та шуми в сенсорних рядах, а також вимоги до низького часу відгуку. Традиційні конвеєри з відсутністю єдиної просторово-часової основи не забезпечують стабільного геоприв'язування та узгоджених агрегатів, що призводить до помилок локалізації подій, дублювання супутникових виявлень та непрогнозованих затримок. Необхідний інтегрований підхід, який поєднує уніфікацію форматів та часових поясів, геокодування зі збереженням коректності для різних словоформ, просторово-часове вирівнювання на шестикутній сітці та двошарову агрегацію добових індикаторів ризику.

На основі проведеного аналізу було виявлено протиріччя між вимогою оброблення великого обсягу подій у наближеному до реального часу режимі та неоднорідністю джерел (OSINT, супутникові FRP, радіаційні сенсори, метеополі);

необхідністю точного геоприв'язування при багатомовності та змінюваності словоформ текстів та відсутності єдиного реєстру географічних назв; потребою швидкого оновлення агрегатів та вимогою відтворюваності розрахунків за історією подій; високими вимогами до точності індикаторів ризику та обмеженими інфраструктурними ресурсами. Ці протиріччя зумовлюють наукове завдання створення нових методів та програмних засобів для уніфікації, геокодування та просторово-часової агрегації гетерогенних потокових даних з метою виявлення та прогнозування екологічних загроз.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася в рамках ініціативної НДР «Методи оптимізації продуктивності програмного забезпечення та використання технологій штучного інтелекту для вдосконалення програмних систем аналітики великих даних» (державний реєстраційний номер 0124U001790, 2024-2026), яка здійснювалася у Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського» на кафедрі програмного забезпечення комп'ютерних систем (див. Додаток А).

Мета і завдання дослідження. Метою дисертаційної роботи є підвищення швидкодії оброблення гетерогенних потокових даних, точності геоприв'язування та забезпечення відтворюваності просторово-часових показників екологічних ризиків шляхом створення алгоритмічного та програмного забезпечення для реалізації конвеєра уніфікації, геокодування та агрегації гетерогенних потокових даних. Для досягнення мети були сформульовані такі **наукові завдання**:

- виконати аналіз сучасних програмних систем моніторингу та прогнозування екологічних загроз, а також визначити вимоги до програмного забезпечення для оброблення гетерогенних потокових даних;
- розробити метод геокодування з урахуванням змінюваності словоформ та його програмну реалізацію, оцінити якість прив'язування;

- реалізувати програмний модуль для прийняття рішень на основі інтелектуального аналізу гетерогенних потокових даних;
- спроектувати архітектуру програмного забезпечення з двошаровою НЗ-агрегацією, матеріалізованими добовими поданнями та подійним журналом для відтворення станів;
- розробити модульний конвеєр уніфікації для потоків даних OSINT, FRP, радіаційних та метеорологічних потоків з єдиною часовою базою UTC та нормалізованими одиницями виміру;
- визначити правила доменної агрегації: FRP, метеокорекцію коефіцієнтом вирівнювання за вітром та нормування радіаційних рядів;
- провести експериментальну перевірку розробленого програмного забезпечення за метриками часу відгуку, пропускну здатності, повноти та точності геокодування, а також частки дублікатів FRP.

Об’єкт дослідження – процеси аналізу вимог, розроблення, забезпечення якості та впровадження програмних засобів для оброблення і аналізу гетерогенних потокових даних у системах екологічного моніторингу стану довкілля.

Предмет досліджень – методи, алгоритми та програмні засоби уніфікації, геокодування та просторово-часової агрегації неоднорідних потокових даних незалежно від їхньої структури.

Методи дослідження. Використано методи інженерії програмного забезпечення; методи інженерії даних та статистичного аналізу; методи оброблення природної мови; просторову агрегацію на шестикутній сітці НЗ; аналіз часових рядів та оброблення подій мікропакетами; експериментальне моделювання з побудовою матеріалізованих подань, методи машинного навчання для розроблення модуля прийняття рішень на основі інтелектуального аналізу гетерогенних потокових даних.

Наукова новизна одержаних результатів полягає в наступному.

Уперше розроблено метод геокодування зі збереженням коректності для різних словоформ багатомовних OSINT-повідомлень, характерними рисами якого

є поєднання морфологічної нормалізації змінюваних форм, каскаду точного, нечіткого та фонетичного зіставлення з реєстром географічних назв та усунення контекстуальної неоднозначності в локальному оточенні НЗ, що, на відміну від існуючих методів геокодування, орієнтованих на англomовні тексти, дає змогу підвищити частку коректних прив'язувань топонімів в середньому на 21% та скоротити середній час пошуку координат в середньому на 51 %.

Уперше розроблено ймовірнісну геопросторову програмну модель прийняття рішень на основі аналізу гетерогенних потокових даних, характерними рисами якої є інтеграція результатів геокодування, багатоджерельної перевірки та доменної агрегації на шестикутній сітці НЗ з підтримкою сценарних обмежень, що, на відміну від відомих моделей, дає змогу оцінювати небезпечні зони, формувати рекомендації щодо планування евакуації та розміщення нових укриттів з кореляцією 0,78 з офіційними повітряними тривогами.

Уперше розроблено архітектуру уніфікованого мікропакетного конвеєра гармонізації гетерогенних потоків даних, характерними рисами якого є детерміноване перетворення подій до єдиної часової бази, нормалізація одиниць вимірювання та уніфікація схем подій у межах одного процесу, що, на відміну від стандартних багатоетапних ETL-конвеєрів, дає змогу скоротити затримку уніфікації в середньому на 80%.

Уперше розроблено архітектурний шаблон НЗ-DSMV (Dual Scale Materialized Views) для аналізу гетерогенних потоків даних, характерними рисами якого є фіксація подій на дрібнішій сітці r10 з автоматичною побудовою та підтримкою матеріалізованих подань на вищій сітці r7, що, на відміну від розподілених архітектур з окремими компонентами агрегації, дає змогу в 5 разів скоротити середній час відповіді системи.

Практичне значення одержаних результатів. Запропоновані методи та програмні засоби забезпечують безперервний моніторинг екологічних ризиків, підтвердження подій за кількома джерелами, формування добових просторово-часових індикаторів для підтримки прийняття рішень органами

цивільного захисту та екологічного нагляду, а також застосування на обмежених обчислювальних ресурсах у режимі автономної роботи. Удосконалені процедури доменної агрегації ризику для гетерогенних потокових даних екологічного моніторингу, на відміну від існуючих процедур, поєднують FRP-агрегацію, метеокорекцію для врахування вітру та метеофакторів, нормування радіаційних показників та єдину часову базу. Запропонований метод інтеграції LSTM-прогнозування у єдиний просторово-часовий конвеєр екологічного моніторингу, на відміну від реактивних методів моніторингу, забезпечує перехід до проактивного режиму підтримки рішень на основі короткострокових прогнозів.

Впровадження одержаних результатів. Результати наукових досліджень були впроваджені в навчальний процес Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» при викладанні дисципліни «Великі дані та аналітика» для студентів освітньо-кваліфікаційного рівня «Магістр» спеціальності 121 «Інженерія програмного забезпечення» (див. Додаток Б).

Особистий внесок здобувача. Всі результати, наведені в основній частині дисертаційної роботи, отримані здобувачем самостійно (див. Додаток В).

У наукових працях, опублікованих у співавторстві: [1] – у статті автором виконано аналіз сучасних підходів до програмної обробки гетерогенних потокових даних, узагальнено вимоги до уніфікації схем, форматів та часових узгоджень; [2] – у статті автором здійснено програмний аналіз відкритих рядів радіаційного моніторингу, розроблено методи нормування сигналів, виявлення аномалій датчиків та виконано експериментальну перевірку результатів; [3] – у статті автором проаналізовано архітектури розподіленої потокової обробки та обґрунтовано інтеграцію LSTM у CEP-контури для реалізації прогнозної аналітики; [4] – у статті автором запропоновано гібридну нейронну модель для подієвого аналізу потоків соціальних мереж та експериментально підтверджено зниження MAE порівняно з базовою LSTM; [5] – у статті автором розроблено модульний конвеєр уніфікації гетерогенних потокових даних для аналітики

екологічних ризиків у реальному часі з використанням НЗ-агрегації; [6] – у статті автором побудовано відтворюваний конвеєр аналізу поточкових даних, реалізовано інженерію інтерпретованих ознак та виконано структурно-поведінковий аналіз взаємодій; [7] – у матеріалах конференції автором реалізовано RNN-підхід до поточкового аналізу та прогнозування даних у режимі, наближеному до реального часу; [8] – у матеріалах конференції автором систематизовано сценарії інтеграції методів штучного інтелекту в системи обробки гетерогенних поточкових даних та сформульовано практичні рекомендації щодо їх застосування; [9] – у матеріалах конференції автором розроблено метод та програмне забезпечення на основі моделі HybridLSTM для виявлення подій та аналізу трендів у потоках даних соціальних мереж; [10] – у матеріалах конференції автором створено програмний конвеєр OSINT-геокодування з локальним реєстром географічних назв та морфологічною нормалізацією для автономної обробки поточкових даних; [11] – у матеріалах конференції автором розроблено програмний конвеєр картування радіаційних аномалій у реальному часі з корекцією за Kp-індексом, GOES X-ray та просторовою НЗ-агрегацією.

Апробація результатів дисертації. Основні положення роботи доповідалися та обговорювалися на міжнародних наукових конференціях: «The latest science and technology achievements and their significance for society» (Częstochowa, Republic of Poland, 2023); «Modern Trends in the Development of Scientific Space» (Dresden, Germany, 2024); «Science, Technology, and Society in the 21st Century» (Amsterdam, Netherlands, 2025); «Trends, Issues, and Challenges in Modern Science» (Cambridge, UK, 2025), а також на науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг. ПМК, 2024» (Київ, Україна, 2024).

Публікації. За матеріалами дисертації опубліковано **11** наукових праць, з яких **6** – статті у фахових виданнях категорії «Б» [1–6] та **5** – публікації у матеріалах наукових конференцій [7–11].

Структура роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, що включає 115 найменувань та сім додатків. Загальний обсяг роботи становить 234 сторінки, у тому числі 172 сторінок основного тексту, 34 рисунків та 35 таблиць.

РОЗДІЛ 1. АНАЛІЗ ПРОГРАМНИХ СИСТЕМ МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ ЕКОЛОГІЧНИХ ЗАГРОЗ

1.1. Стан досліджень з програмної аналітики потокових даних для прогнозування екологічних загроз

Аналіз гетерогенних потокових даних у реальному часі є критично важливим у контексті техногенних та природних катастроф, оскільки сучасні загрози мають складний, багатofакторний характер і часто проявляються одночасно в різних фізичних та інформаційних середовищах. Системи моніторингу спираються на велику кількість сенсорів та джерел даних: радіаційні датчики, сейсмічні станції, метеорологічні супутники, магнітометри, мережі спостереження за космічною погодою, а також інформаційні та телекомунікаційні системи. Кожне з цих джерел окремо може зафіксувати аномалію, однак лише їх комплексний аналіз дозволяє коректно інтерпретувати ситуацію та уникнути хибних висновків. Показовим прикладом є фіксація підвищеного рівня радіаційного фону. Така подія може свідчити про техногенну аварію (наприклад, витік радіоактивних речовин на промисловому об'єкті або АЕС), але водночас аналогічні показники можуть бути зумовлені природними факторами. Зокрема, потужні сонячні спалахи та корональні викиди маси здатні викликати інтенсивні потоки заряджених частинок, які досягають магнітосфери Землі та призводять до зростання радіаційного фону, збоїв у роботі супутників, навігаційних систем та навіть наземної електроніки. Відомими прикладами таких явищ є подія Керрінгтона 1859 року, яка спричинила масштабні збої телеграфних мереж, а також сильні сонячні спалахи 1989 та 2003 років, що призвели до відключень енергосистем та порушень зв'язку. Інший приклад – за даними NOAA, 6 травня 2023 року спостерігалася аномальна сонячна активність з кількома активними областями на Сонці (області 3293, 3294, 3296, 3297, 3299). Найбільша область, 3297, мала площу 0,460 мільйонних часток сфери та складну магнітну

конфігурацію. Ця активність спричинила підвищення рівня радіації, що підтверджується даними – рекордне значення 2627 (рис. 1.1, 1.2).

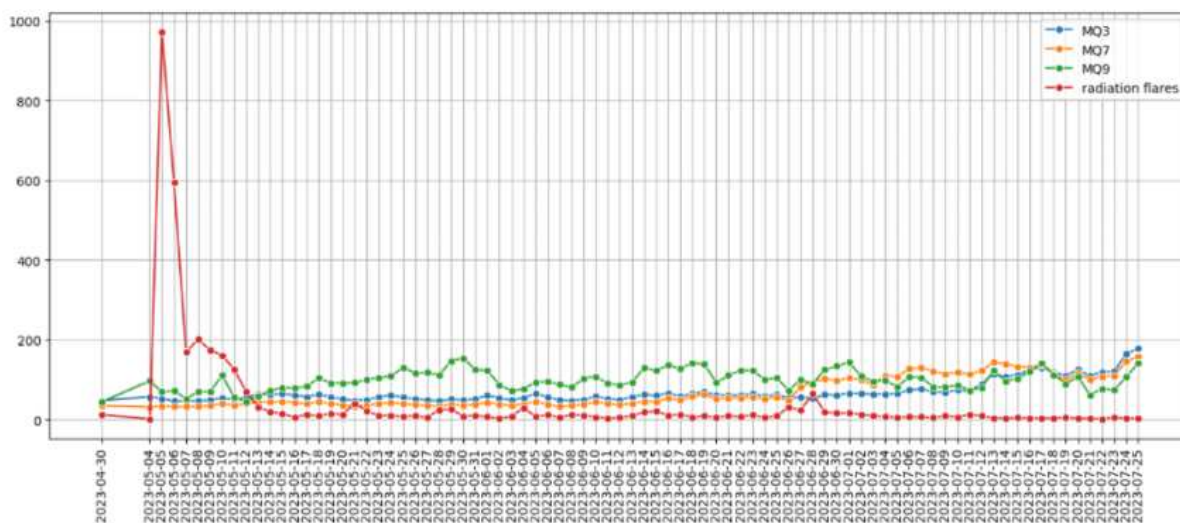


Рис. 1.1. Дані про показники та аномальний сплеск радіації, отримані з сенсорів M03, M07, M09 (травень-липень 2023)

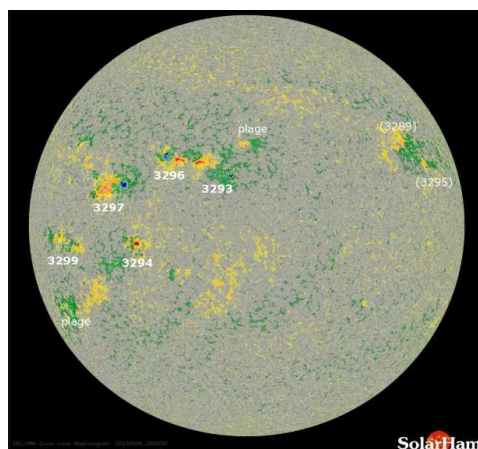


Рис. 1.2. SDO/HMI магнітограма (06.05.2023) [12]

Поєднання інформації з радіаційних сенсорів, спостережень за сонячною активністю, даних супутників космічної погоди та наземних інфраструктурних систем дозволяє відрізнити техногенну загрозу від природної.

Багатоджерельний аналіз у реальному часі дозволить забезпечувати більш точну ідентифікацію причин аномалій, зменшити ризик паніки або неправильних

управлінських рішень та своєчасно впроваджувати превентивні заходів – від коригування режимів роботи критичної інфраструктури до оперативного інформування населення та служб реагування.

Сучасні конфлікти породжують не лише військові, а й екологічні ризики, такі як викиди небезпечних речовин, пожежі, руйнування інфраструктури, радіаційні та хімічні інциденти. Сучасні дослідники акцентують увагу на необхідності оперативного моніторингу та прогнозування екологічних загроз з використанням програмних інструментів аналітики даних у реальному часі [13-37], оскільки значна частина вхідних даних для такого моніторингу – це потоки відкритої інформації (інформаційні канали соціальних мереж, місцеві новини, звіти громадських ініціатив, датчики якості повітря тощо).

В Україні після 2022 року масові руйнування, прориви гідротехнічних споруд та удари по енергетичній та промисловій інфраструктурі створили складну екологічну ситуацію. Руйнування Новокаховської греблі спричинило гуманітарну та екологічну катастрофу в Україні. Було затоплено природні парки, річка Дніпро забруднена машинним маслом, відходами та мінами, частина яких потрапила в Чорне море (рис. 1.3).

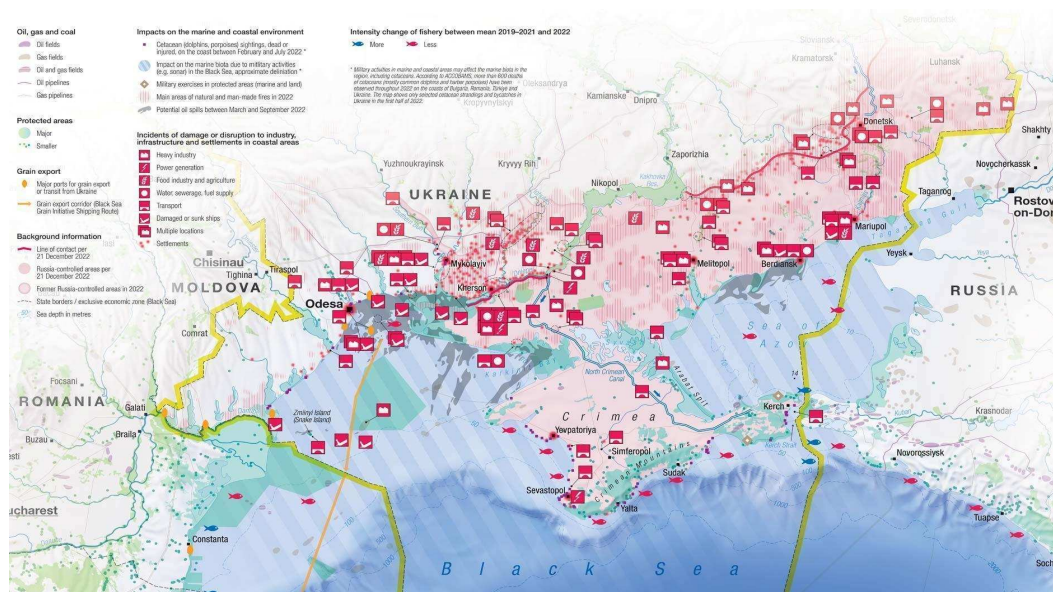


Рис. 1.3. Мапа загроз екологічно важливим районам з цінними рибальськими ресурсами морського узбережжя Чорного моря [37]

Останні наукові роботи (зокрема, публікації українських авторів та звіти CEOBS та Harvard-групи) надають узагальнену картину масштабів екологічних збитків – від локальних хімічних ризиків до масштабних викидів CO₂ та руйнування заповідних територій. Ці роботи систематизують джерела ризиків (руйнування ТЕЦ, ОПЗ, дамб) та підкреслюють потребу у синтезі супутникових, польових та соціальних джерел даних для оцінки екологічних небезпек [38].

У сучасних дослідженнях екологічних наслідків воєнних дій переважають аналітичні огляди та вимірювальні дослідження, що базуються на супутникових даних та польових спостереженнях. У роботі [39] показано ефективність комбінування даних Sentinel, MODIS та SAR для виявлення масштабних подій (пожежі, руйнування, зміни покриття), однак відсутні алгоритми автоматичної триангуляції подій між OSINT та дистанційним зондуванням у режимі реального часу. У дослідженні [40] обґрунтовано необхідність програмних інструментів моніторингу екологічних ризиків, проте не запропоновано стандартизованих механізмів інтеграції радіаційних, повітряних та текстових OSINT-даних. Платформа SaveEcoBot [41] забезпечує відкритий доступ до екологічних показників через API, однак не реалізує алгоритмічну оцінку достовірності повідомлень та автоматизоване прогнозування ризиків. Робота [42] демонструє потенціал поєднання супутникових, сенсорних та польових даних, але окреслює потребу у стандартизованих конвеєрах збору й оброблення потокових даних у реальному часі. Розвиток ГІС-підходів [43–45] та програмних рішень для аналізу супутникових знімків [46], а також ГІС-систем моніторингу промислових об'єктів [47] та тривимірного моделювання підземних вод [48] підтверджує потенціал просторової інтеграції різнорідних джерел. Водночас відсутні уніфіковані програмні рішення, що забезпечують об'єднання супутникових, сенсорних та OSINT-потоків у межах єдиної масштабованої системи з формалізованою оцінкою невизначеності. Міжнародні ініціативи (зокрема, Safecast) демонструють можливості краудсорсингових сенсорних мереж, однак їх інтеграція з текстовими OSINT-потокими ускладнюється проблемами узгодження часових міток, джерел

походження даних та просторової інтерполяції. У працях з кризової інформатики та гуманітарного NLP [49–51] визначено три ключові виклики: фільтрація великих потоків мультимедійних повідомлень, точне геоприв'язування (NER, нормалізація, геокодування) та синтез OSINT з фізичними вимірюваннями з явною оцінкою довіри. Класичні алгоритми виявлення подій у потокових даних (ключові слова, порогові методи) забезпечують оперативність, але чутливі до шуму, тоді як ML-підходи з людиною в циклі [49] залежать від англomовних ресурсів та зовнішніх API, що обмежує їх застосовність до локальних мов та закритих платформ. Проблема геоприв'язування залишається одним із центральних вузьких місць: роботи [52], [53], [89], [98], [88] описують етапи розв'язання топонімічної неоднозначності, проте наголошують на нестачі кризових корпусів, складності морфологічної нормалізації та необхідності врахування адміністративних та часових змін назв [49–53]. Це свідчить про потребу в алгоритмах геопарсингу, адаптованих до морфологічно складних мов та воєнного контексту.

Дослідження мультиджерельної інтеграції [54] підтверджують, що поєднання соціальних медіа та дистанційного зондування підвищує точність карт уражень, однак стикається з асинхронністю даних та різною просторово-часовою роздільною здатністю. Багатоджерельні каркаси об'єднання (соціальні медіа, SAR, мобільні дані, моделі поширення) демонструють ефективність, але потребують стандартизованих протоколів перевірки та еталонних наборів даних. Проблема переносимості моделей ML при зміні контексту загроз [55] вказує на необхідність адаптивного та контекстно-орієнтованого навчання.

Роботи [56], [57], [59] доводять доцільність інтеграції соціальних медіа з супутниковими даними для реагування на пожежі та катастрофи, проте підкреслюють потребу в автоматизованому очищенні шуму, верифікації повідомлень та мовно-адаптованому геопарсингу. Оглядові дослідження [58], [60] акцентують на проблемах обмежених мовних ресурсів та достовірності даних, набір даних [61] розширює емпіричну базу OSINT-досліджень, проте не супроводжується рішеннями для прогнозування екологічних загроз.

Аналіз наукової літератури [39–61] засвідчив, що наявні підходи здебільшого фрагментарно розв’язують окремі задачі (моніторинг, геоприв’язування, інтеграція, виявлення подій), відсутні алгоритмічні та програмні рішення, які б забезпечували відтворюваний потоковий конвеєр оброблення гетерогенних даних у реальному часі, реалізовували автоматичну триангуляцію OSINT та сенсорних сигналів, підтримували морфологічно адаптивний геопарсинг для кризових доменів, інтегрували моделі прогнозування з механізмами контекстної адаптації та інтерфейс для оперативного реагування.

1.2. Порівняльний аналіз програмних рішень для моніторингу екологічних загроз та їх обмеження

Сучасні програмні рішення для моніторингу екологічних загроз поєднують методи збору, оброблення та візуалізації даних. Вони базуються на інтеграції відкритих джерел інформації, супутникових знімків, даних з безпілотних літальних апаратів, сенсорних мереж та польових звітів. Міжнародних стандартів оцінки екологічної шкоди під час війни немає, але Україна, ЄС, США та ООН застосовують методи моніторингу з використанням супутників, дронів, фото та свідчень очевидців. Для збору та картографування даних з метою планування відновлення та майбутніх воєнних репарацій використовуються такі інформаційні системи екологічного моніторингу, як EcoZagroza [62], Ecodozor [63], SaveEcoBot [64], Ecoaction [65] та Environment People Law [66].

Safecast Radiation Monitoring Platform – глобальна система громадського радіаційного моніторингу, що забезпечує збір даних з розподілених сенсорів, їх просторову прив’язку та візуалізацію.

EcoZagroza є офіційною вебплатформою Міністерства захисту довкілля України, її головне призначення – автоматизований збір повідомлень про екологічні загрози з географічною прив’язкою, включно з показниками хімічного складу води та рівня радіації. Платформа пропонує інтерактивну карту,

функціональність соціальної взаємодії, систему сповіщень та англomовний інтерфейс з «особистим кабінетом» користувача.

Ecodozor – це платформа, створена завдяки співпраці з UNEP, ОБСЄ та гуманітарної ініціативи REACH, що виконує картографування екологічних інцидентів, включаючи наслідки військових дій, забруднення, пожежі та повені, містить дані про понад 29 000 подій, що становлять загрозу довкіллю. Система моніторингу обробляє інформацію, отриману з дронів, супутникових знімків, проте потребує масштабування для роботи з великими обсягами даних та глибшої аналітики [63] (рис. 1.4).

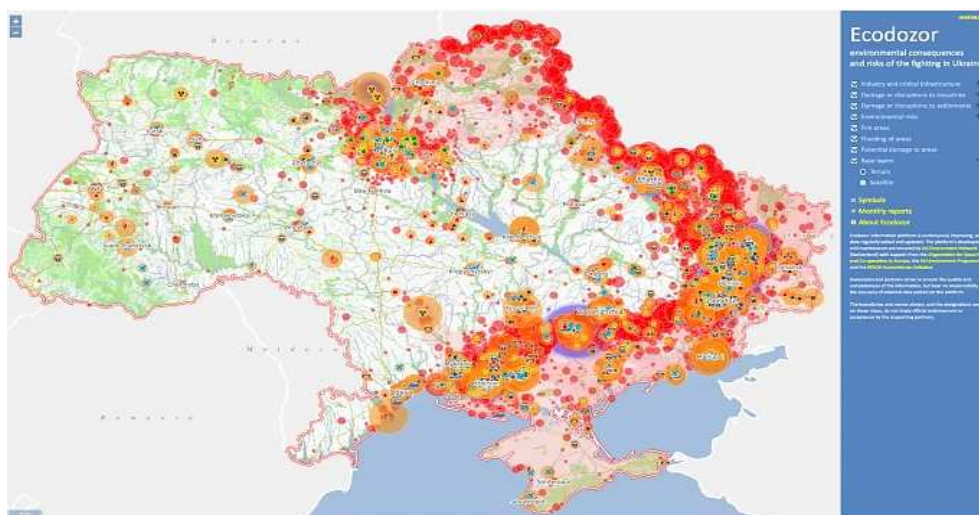


Рис. 1.4. Платформа Ecodozor [63]

SaveEcoBot – це платформа моніторингу екологічних даних, що агрегує карти якості повітря, рівнів радіації, пожеж та інших екологічних показників. Вона надає дані про рівень забруднення та екологічні порушення у форматі інтерактивних карт, чат-ботів, API для відкритих даних, а також аналітичних інструментів та віджетів. Технічно система інтегрує відкриті дані з десятків джерел та підтримує роботу через Telegram- та Viber-боти, а також API-інтерфейси для розробників [64].

Ecoaction, або *Центр екологічних ініціатив*, є громадською організацією, яка веде вебплатформу для публікації аналітики, висвітлює позиції щодо екологічного законодавства, кліматичної політики, зеленого відновлення та інтеграції стандартів ЄС. Сайт функціонує на базі CMS з можливістю розміщення аналітичних матеріалів та інтерактивних звернень [65].

Copernicus Emergency Management Service (EMS) – це супутниково орієнтована система моніторингу надзвичайних ситуацій, що використовує дистанційне зондування Землі та просторову агрегацію [66].

З точки зору інженерії програмного забезпечення, розглянуті платформи реалізують комплексний підхід до управління екологічною інформацією, що включає модулі для прийому повідомлень від користувачів, автоматизовану перевірку та нормалізацію даних, геопросторове оброблення та інтерактивну візуалізацію результатів. З архітектурної точки зору такі системи будуються за принципами багаторівневих вебдодатків, у яких клієнтська частина реалізує інтерфейс користувача з можливістю доступу через браузер та мобільні застосунки, тоді як серверна частина забезпечує бізнес-логіку, керування базами даних та взаємодію з зовнішніми API. Важливим компонентом є картографічний модуль, що використовує бібліотеки Leaflet або OpenLayers для відображення геопросторових даних, а також інтеграцію зі супутниковими сервісами та датчиками IoT для отримання оперативної екологічної інформації. Розгортання таких систем вимагає комплексної інфраструктури, що включає вебсервери з підтримкою HTTPS, геосервери для оброблення просторових запитів, бази даних з підтримкою географічних функцій (PostgreSQL/PostGIS), а також модулі для оброблення великих обсягів даних (Hadoop, Spark) при високій інтенсивності надходження інформації. Необхідною умовою є забезпечення безперервного резервного копіювання та захисту каналів передачі даних, оскільки ці системи працюють з чутливою інформацією, зокрема, щодо розташування потенційно небезпечних об'єктів. Водночас, розглянуті платформи мають низку обмежень.

З технічної точки зору, критичною є залежність від стабільного інтернет-з'єднання та працездатності зовнішніх сервісів (наприклад, супутникових постачальників даних).

При роботі з повідомленнями від користувачів виникає проблема перевірки та фільтрації застарілої інформації, що вимагає впровадження додаткових алгоритмів машинного навчання або модерації. У контексті масштабування особливої уваги потребує оптимізація картографічної візуалізації за великої кількості об'єктів та запитів. З організаційної точки зору обмеження пов'язані з недостатнім рівнем технічної грамотності користувачів або нерівномірним охопленням територій, що призводить до фрагментарності даних.

Типова архітектура розглянутих екологічних інформаційних платформ передбачає модульну побудову, що забезпечує інтеграцію збору, зберігання, оброблення та візуалізацію даних (рис. 1.5).

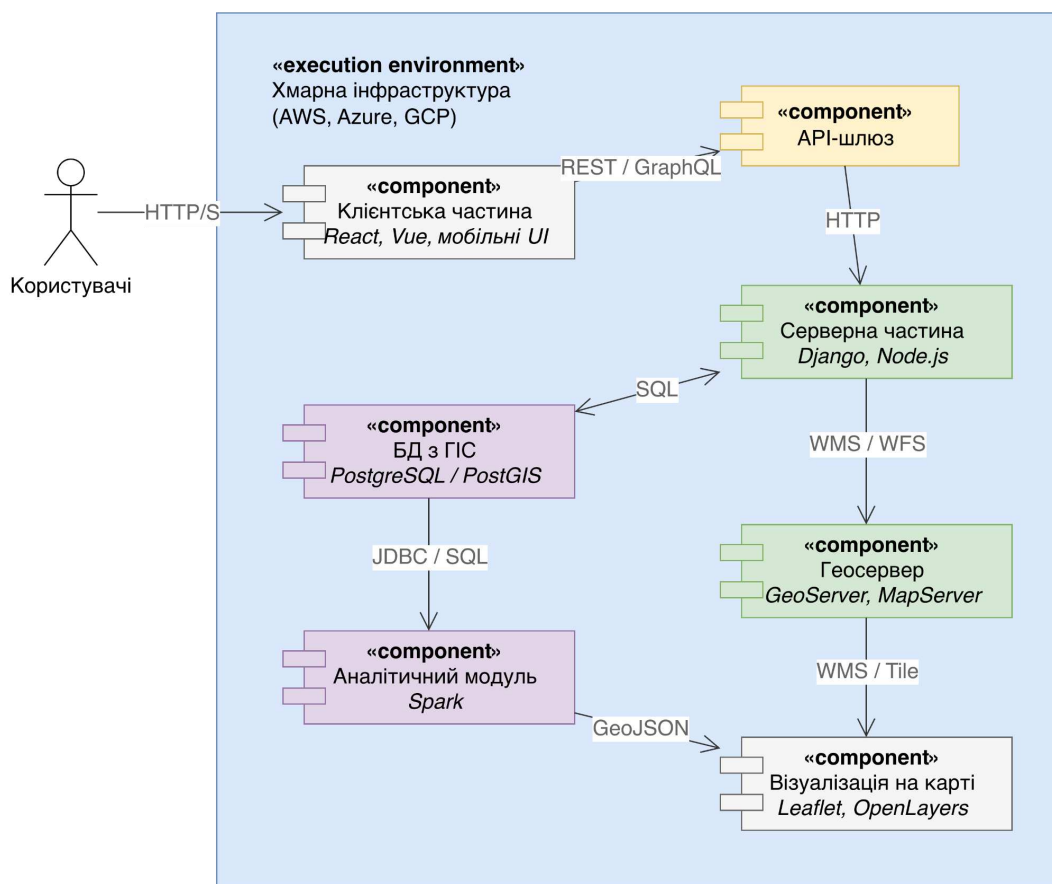


Рис. 1.5. Узагальнена UML- діаграма компонентів екологічних систем моніторингу

Основою таких програмних систем є клієнтська частина, реалізована на сучасних вебфреймворках (React, Vue.js) або у вигляді мобільних застосунків, що забезпечує доступ користувачів до даних. Передача запитів та обмін інформацією здійснюються через API-шлюз, який виконує роль посередника між користувачем та серверною частиною, забезпечуючи контроль доступу, кешування та балансування навантаження. Серверна частина зазвичай побудована на основі Django, Node.js або аналогічних платформ та відповідає за бізнес-логіку, взаємодію з базою даних та геоінформаційним сервером. У якості сховища даних використовується PostgreSQL з розширенням PostGIS, що дає змогу працювати з просторовими даними. Геосервери (GeoServer, MapServer) виконують рендеринг картографічної інформації та інтеграцію з візуалізаційними бібліотеками (Leaflet, OpenLayers). Хмарна інфраструктура (AWS, Azure, GCP) забезпечує масштабованість, резервне копіювання та географічно розподілену роботу сервісу. Розгортання розглянутих систем потребує хмарного або гібридного середовища з підтримкою контейнеризації (Docker, Kubernetes), систем моніторингу (Prometheus, Grafana) та інструментів безперервної інтеграції та доставки CI/CD. До критичних нефункціональних вимог належать висока доступність, стійкість до збоїв, інформаційна безпека та відповідність стандартам захисту персональних даних GDPR. Серед основних недоліків таких платформ – значні витрати на підтримку та оновлення системи, необхідність високої кваліфікації персоналу для адміністрування, а також ризик втрати даних у разі кібератак або збоїв хмарних провайдерів. Наявні екологічні платформи, такі як EcoZagroza та Ecodozor вимагають постійної оптимізації архітектури, адаптації до зростання обсягів даних та впровадження інструментів підвищення достовірності отриманої інформації. SaveEcoBot та подібні проекти збирають показники якості повітря та надають API та візуалізації для регіонів України. Даними платформами виконується збір та публікація оперативних даних про забруднення повітря, відкритий доступ до мап та API. Основною проблемою цієї системи моніторингу є слабка інтеграція з OSINT - повідомленнями про обстріли чи пожежі, відсутність

автоматичної кореляції сплесків забруднення з інцидентами; також бракує стандартизованих інтерфейсів для потокового підключення до конвеєрів моніторингу. Бази даних ACLED та GDELT є прикладами масштабного подієвого кодування, що поєднує формалізовану методологічну класифікацію з просторово-часовими атрибутами подій. Їх підходи передбачають використання показників точності та рівня впевненості, а також обов'язкове збереження інформації про джерело походження даних [67-68].

ACLED (Armed Conflict Location & Event Data Project) застосовує стандартизовану методологію ручного та напівавтоматичного кодування подій, що включає перевірку джерел, класифікацію за типами інцидентів та геоприв'язування з точністю до населеного пункту. У контексті екологічних загроз та воєнних дій такий підхід дозволяє інтегрувати події, пов'язані з пошкодженням інфраструктури, хімічними викидами чи пожежами, у єдину базу даних для подальшого прогнозування.

Технічна реалізація ACLED базується на REST API з підтримкою великих структурованих баз подій, що оновлюються щотижня у стандартизованих форматах CSV та JSON з можливістю інтеграції з ГІС [67], [105].

GDELT (Global Database of Events, Language, and Tone) реалізує масштабний автоматизований збір новинних повідомлень з понад 100 мов світу з щохвилинним оновленням, використовуючи алгоритми ML, автоматичний переклад, категоризацію подій за CAMEO-кодами та геокодування. GDELT репрезентує підхід, орієнтований на вебскрейпінг з використанням обробки природної мови, емоційного аналізу та класифікації подій за CAMEO-кодами [68], [106].

Liveuamap – це інтерактивна карта конфліктів, яка об'єднує повідомлення з медіа, соціальних мереж та офіційних джерел, фокусуючись на швидкому геоприв'язуванні та класифікації подій [80]. Платформа включає екологічно релевантні інциденти (пожежі чи руйнування об'єктів хімічної промисловості), але алгоритмічне оброблення є обмеженим; саме тут можна додати модулі

автоматичного аналізу зображень, прогнозування розповсюдження забруднень та інтеграції з сенсорними даними [69].

Автоматизований збір OSINT-даних здійснюється через API та потокові канали (Twitter Streaming API, RSS) [49] з подальшим попереднім обробленням: нормалізацією, дедуплікацією, перекладом та тематичною фільтрацією [54]. Семантичний аналіз на основі NLP забезпечує виявлення сутностей та ключових атрибутів подій (час, місце, тип інциденту) [58], [60], тоді як геопарсинг і геокодування пов'язують текст з координатами; для зменшення помилкових прив'язувань застосовують ML-підходи [52–53], [55]. Інтеграція з супутниковими даними реалізується засобами комп'ютерного зору (сегментація, розпізнавання об'єктів) [57], [59], [60]. Платформи AIDR [49], CrisisLex [51], Big Crisis Data [54], ACLED [67], GDELT [68], Liveuamap [69] демонструють потокову класифікацію та візуалізацію подій, однак відрізняються рівнем автоматизації та перевірки. Методи оброблення OSINT формують повний цикл: збір (API, Kafka, сенсорні джерела), попереднє оброблення (NLTK, spaCy, OpenCV), семантичний аналіз (NER, LDA/BERTopic, BERT), геопросторове моделювання (GeoNames API, ST-DBSCAN, Leaflet), прогнозування (ARIMA, LSTM/GRU, Random Forest) та перевірку (Google Earth Engine, Copernicus). Для систем виявлення та прогнозування екологічних загроз необхідна інтеграція цих компонентів у масштабований програмний конвеєр із підтримкою поточкових платформ (Kafka, Flink), супутникових і сенсорних API (SaveEcoBot, Copernicus). Оброблення включає фільтрацію шуму, лематизацію, розв'язання топонімічної неоднозначності [52], [53], тематичне моделювання та використання кризових лексиконів [51]. Прогнозування базується на поєднанні статистичних, стохастичних та ML-моделей, а мультимодальна інтеграція реалізується на рівні ознак або рішень із баєсівською оцінкою невизначеності [96]. Верифікація передбачає перехресну перевірку джерел та супутникове підтвердження геолокації [57], [59], що підвищує достовірність результатів.

Українські платформи екологічного моніторингу демонструють специфічні підходи до роботи з національними даними. EcoZagroza як GovTech-рішення забезпечує автоматизований збір повідомлень з геоприв'язкою, перевірку та валідацію через стандартизовані форми, базові аналітичні модулі та інтеграцію з державними реєстрами. Платформа надає API для картографування та мобільний додаток з інтерактивною картою, що дозволяє використовувати її для збору структурованих екологічних подій в Україні з можливістю масштабування для потокових OSINT-даних. Ecodozor спеціалізується на обробці даних з супутників, дронів та польових звітів, забезпечуючи виявлення та маркування зон ризику через багат шарову карту. SaveEcoBot демонструє інтеграційний підхід, об'єднуючи відкриті екологічні дані з десятків джерел через автоматичну публікацію у чат-ботах, фільтрацію та агрегацію. Платформа Ecoaction фокусується на аналітичній обробці даних про екологічну політику та наслідки через тематичну категоризацію та формування звітів.

На основі проведеного аналізу програмних рішень обґрунтовано доцільність підходу, що поєднує методи стандартизації та багатомовного аналізу текстових даних, архітектурні рішення для інтерактивної картографічної візуалізації подій, API-орієнтований збір та верифікацію екологічних показників, автоматизований аналіз супутникових та аерофотознімків, а також інтеграцію аналітичних та експертних матеріалів.

Реалізація такого підходу має забезпечувати автономність функціонування та мінімізацію зовнішніх залежностей, що є критичним в умовах воєнного часу. Такий підхід створює підґрунтя для побудови платформи, здатної здійснювати збір, оброблення, перевірку, прогнозування та візуалізацію екологічних загроз у реальному часі.

Оброблення потокових повідомлень з відкритих джерел охоплює методи збору, фільтрації, аналізу та візуалізації інформації з соціальних мереж, новинних стрічок, супутникових знімків та сенсорних мереж.

1.3. Методи збору та попередньої обробки гетерогенних потокових даних

Гетерогенні потоки даних OSINT формуються з дискретних текстових повідомлень різної довжини, стилю та мови, що надходять з соціальних мереж та новинних каналів. Такі потоки характеризуються нерівномірністю надходження, високим рівнем шуму та частими повтореннями [51], [101]. Точність виявлення подій підвищується завдяки перехресному підтвердженню текстових повідомлень з даними фізичних сенсорів та супутниковими термічними спостереженнями (FIRMS MODIS/VIIRS), що забезпечують високочастотну фіксацію «гарячих точок», корельованих з пожежною активністю. Просторово-часове зіставлення виконується в обмежених часових вікнах та радіусних околах навколо геокодованих згадок з урахуванням орбітального покриття, хмарності та відмінностей денних та нічних каналів, що знижує ризик хибних відповідностей. Для узгодження різнорідних виявлень застосовується двоетапне агрегування на сітці НЗ: первинне картування з короткими інтервалами та подальше укрупнення осередків, що поєднує оперативність із надійністю добових індикаторів ризику [102], [79]. Дедуплікація між супутниками здійснюється в осередках та часових вікнах з вибором максимального FRP та подальшим добовим сумуванням і проєкцією на сітку НЗ $r=7$, що зменшує вплив артефактів та формує узгоджену пожежну компоненту для подальших розрахунків [79]. Дані мереж іонізуючого випромінювання, які характеризуються різною частотою вимірювань та телеметричними розривами, усереднюються на добовому кроці, нормалізуються відносно ковзної 14-денної базової лінії та подаються як z-аномалія γ_z з обмеженням діапазону, що забезпечує міжстанційну зіставність та стійкість до збоїв; просторове узгодження виконується через інтерполяцію або агрегацію на НЗ [77], [78]. Текстові потоки проходять очищення (видалення службових маркерів), мовну та морфологічну нормалізацію, автоматичне визначення мови та географічну фільтрацію з використанням реєстрів топонімів та глобальних баз (зокрема, GeoNames), що спрощує подальший геопарсинг та просторово-часову агрегацію [91]. Після нормалізації формується єдиний журнал подій для модулів

класифікації, NER та геокодування; стабілізація потоків забезпечується дедуплікацією (контрольні суми, близькість векторних представлень), уніфікацією часових міток і «м'якою» агрегацією атрибутів, що мінімізує фрагментацію при зіставленні з сенсорними сигналами. Таблиця 1.1. узагальнює ключові характеристики гетерогенних потоків даних (соціальні мережі, новини, супутникові та сенсорні джерела) та відповідні методи їх попередньої обробки, демонструючи відмінності у часово-просторовій роздільності, форматах доступу, процедурах нормалізації та практичному призначенні для виявлення та оцінювання екологічних загроз.

Таблиця 1.1

Порівняння потоків та відповідних методів попередньої обробки

Джерело	Сигнал	Часова розд.	Просторова роздільність	Доступ/формат	Ключова нормалізація	Призначення
Соціальні мережі/ Новини	Опис подій	секунди–хв.	текст → H3	API/ JSON	лексичні тригери, мова [51]	Класифікація / геокодування [101]
FIRMS (MODIS/ VIIRS)	FRP (пожежі)	хвилини –год.	~375–1000 м → H3	API/ CSV	15-хв дедуплікація $r=10$, добове сумування $r=7$	пожежна компонента ризику [79, 104]
SaveEcoBot	γ -фон (nSv/h)	хвилини –год. → доба	точки → H3	API/ JSON	добові медіани, γ_z з 14-денною базою	техногенні аномалії [77, 79]

У контексті супутникових даних виконується оброблення пропусків (наприклад, через хмарність) шляхом збільшення часових вікон перевірки або застосування ковзних накопичувальних сум, що підвищує ймовірність перетину з релевантним текстовим епізодом. З урахуванням описаних операцій формується конвеєр попереднього оброблення, що збалансовує ранню фільтрацію на основі лексики, часово-просторову нормалізацію для сенсорних потоків та уніфікацію текстових записів для модулів NLP. Отримані артефакти забезпечують сталу

якість даних на етапі класифікації й геопарсингу та зменшують частку хибних позитивів у процесах підтвердження. Таким чином закладається основа для подальшої оцінки ризику та розрахунку показників попиту, де необхідні стабільні добові агрегати та сумісні ключі H3 для просторового згортання.

1.4. Методи розпізнавання подій на основі NLP

Розпізнавання подій у багатомовних потоках формулюється як поєднання класифікації повідомлень та видобування сутностей з подальшою аргументацією. Для вирішення цієї задачі застосовуються багатомовні трансформери XLM-R та mDeBERTa-v3, що демонструють високі показники на міжмовних тестових наборах та підтримують українську та споріднені мови [81], [82], [95].

Ключовою проблемою при навчанні таких моделей є дисбаланс класів: деякі типи подій зустрічаються значно рідше за інші, що може призвести до ігнорування рідкісних, але критично важливих категорій.

Для подолання цієї проблеми використовується фокальна функція втрат, яка концентрує навчання на складних прикладах:

$$FL(p_t) = -\alpha(1 - p_t)^\gamma \cdot \log(p_t), \quad (1.1)$$

де p_t – ймовірність істинного класу, α – ваговий коефіцієнт, γ – параметр фокусування. Ця функція знижує вплив негативних прикладів, які модель вже впевнено класифікує та концентрує навчання на складних випадках, де класифікація є неоднозначною [83].

Для підвищення узагальнюваності додатково застосовуються техніки текстового «збагачення» – зворотний переклад та парафразування, які генерують синтетичні приклади з близьким змістом, але відмінним формулюванням [84].

Модель видобування сутностей виявляє у текстах локації, об'єкти інфраструктури, небезпечні речовини та числові наслідки подій. Подальший бінарний зв'язувальний класифікатор визначає, чи належить виявлена сутність відповідній події, застосовуючи парне кодування на трансформерах [86]. Такий

підхід дозволяє не просто виявити згадки об'єктів у тексті, а встановити їхній зв'язок з конкретною подією, описаною в повідомленні [85], [87].

Для морфологічно насичених мов, таких як українська, попередня токенизація та лематизація здійснюється інструментами Stanza та UDPipe; інколи використовуються вбудовувачі (fastText та flair) у ролі ознак для простіших моделей [90], [95]. Результатом є вектор ознак події: (тип події, час, геоприв'язування, аргументи/наслідки, P(event)). Формалізація задачі класифікації з багатьма мітками передбачає, що кожному повідомленню x відповідає бінарний вектор міток $y \in \{0, 1\}^K$, де K – кількість класів подій. Модель з параметрами θ оцінює ймовірності $\hat{y} = \sigma(f_{\theta}(x)) \in [0, 1]^K$ через сигмоїдну активацію для кожного класу незалежно; базова функція втрат без урахування фокусування еквівалентна сумі бінарних крос-ентропій по класах. Для дисбалансних колекцій доцільним є поєднання фокальної втрати з вагуванням позитивних та негативних прикладів. Порогове рішення $\tilde{y}_k = 1[\hat{y}_k \geq \tau_k]$ може використовувати як єдиний τ для всіх класів, так і індивідуальні τ_k , оптимізовані за валідаційною кривою точність–повнота на перевіірчній вибірці. Це дозволяє керувати співвідношенням хибнопозитивних та хибнонегативних рішень у класах з різною поширеністю. Для стабільності швидкодії у змінних потоках застосовується постоброблення оцінок, включно з узгодженням температурою, ізотонною чи плоскою регресією, коли це виправдано якістю перевіірочної вибірки. Оцінювання якості у сценарії з багатьма мітками доцільно проводити за мікро- та макро-агрегованими метриками.

Нехай TP_k, FP_k, FN_k – статистики для класу k . Тоді:

$$Precision_{micro} = \frac{\sum_k TP_k}{\sum_k TP_k + \sum_k FP_k}, \quad (1.2)$$

$$Recall_{micro} = \frac{\sum_k TP_k}{\sum_k TP_k + \sum_k FN_k}, \quad (1.3)$$

$$F1_{micro} = 2 \cdot \frac{Precision_{micro} \cdot Recall_{micro}}{Precision_{micro} + Recall_{micro}}. \quad (1.4)$$

Для макро-агрегації обчислюються $F1_k$ для кожного класу окремо та усереднюються: $F1_{macro} = \frac{1}{K} \sum_{k=1}^K F1_k$. У застосуваннях з суттєвим дисбалансом класів $F1_{macro}$ відображає поведінку на рідкісних подіях та реагує на покращення *recall* у «важких» класах, тоді як $F1_{micro}$ віддзеркалює загальний вклад частих класів. Коли існують залежності між мітками (наприклад, «удар» і «пожежа» часто спільно активуються), доцільним є розширення кодування виходів через штраф за несумісні комбінації або використання простих ланцюжків класифікаторів, проте базовий незалежний декодер з індивідуальними порогами часто забезпечує достатню якість при нижчій складності.

Архітектурно модуль класифікації реалізується як енкодер трансформера з пулінгом [CLS]-представлення у фіксовану розмірність h , після чого застосовується багатокласовий лінійний шар $W \in R^{K \times h}$ з незалежними вихідними значеннями. Для зменшення перенавчання використовуються випадкове вимикання нейронів на вбудованих представленнях та в проєкційних шарах, усереднення ваг на етапі прогнозування, а також використання стратифікованих міні-пакетів для стабілізації градієнтів у класах з низькою поширеністю.

У довгих повідомленнях застосовується усікання за токенами з евристичними збереження початкових та фінальних сегментів, або ковзне вікно з «увагою» до фраз-тригерів. Модель видобування сутностей орієнтується на токен-класифікацію зі схемою тегів BIO/BILOU, де категорії LOC, FAC, ORG, HAZARD, VICTIM визначаються на рівні проміжків. Для числових параметрів наслідків (жертви, масштаби, одиниці виміру) додається компонент регулярних виразів та нормалізатор одиниць для приведення до канонічної форми. Окремо застосовується перевірка згадок за шаблонами контекстних тригерів (наприклад, для VICTIM – числівник у сусідстві дієслів «загинуло/поранено»). В результаті формується множина кандидатів сутностей $E = e_i$. Зв'язування сутностей з подіями формулюється як бінарна класифікація пар (m, e_i) , де m – представлення

повідомлення або його релевантного фрагмента, e_i – контекстно-залежне представлення сутності (наприклад, через mean-pooling токенів сутності у просторі енкодера). Класифікатор оцінює $s_i = g(m, e_i)$ та приймає рішення $r^i = 1[s_i \geq \tau_{rel}]$; альтернативно використовується підхід парного кодування з конкатенацією «[CLS] повідомлення [SEP] текст сутності», що забезпечує доступ моделі до повного контексту при мінімальних змінах у графі обчислень.

З огляду на багатомовність та розмаїття стилів, для підвищення стійкості NER та класифікатора застосовуються текстові «збагачення». Зворотний переклад дозволяє генерувати перефрази з близьким смислом, а заміна лексем на синонімічні або морфологічно варіативні форми підвищує лексичне різноманіття навчальних прикладів. Для нерегулярних форм імен власних доцільним є зашумлення символів (видалення/вставка/перестановка) у межах контрольованого радіусу редагувань, що симулює опечатки та транслітераційні варіанти у реальних потоках. Такі процедури зменшують чутливість до поверхневих змін формулювань і покращують узагальнення на джерелах поза тренувальним корпусом [84]. Окрему увагу приділяється контролю якості та пояснюваності. На рівні класифікації вихідні значення лінійного шару перетворюються у узгоджені ймовірності та аналізуються на предмет зсувів даних; на рівні NER перевіряється узгодженість тегів (недопустимість послідовностей I без попереднього B тощо). Для повнотекстового аудиту зберігаються проміжні атрибути: топ- k тригерів, що зумовили позитивне рішення, карта уваги для пояснення локалізації, а також рівні впевненості для кожного модуля. Такі записи сприяють прозорості, полегшують розв'язання конфліктних випадків (наприклад, одночасна присутність HAZARD та відсутність підтвердження події) та забезпечують підґрунтя для подальшого активного добору прикладів для розмітки. Для досягнення цільових затримок у режимі мікропакетів застосовується об'єднання повідомлень у пакети з лімітом токенів, попереднє кешування вбудованих представлень частих фраз та квантизація ваг енкодера на етапі розгортання. Використання GPU/CPU

регулюється за рахунок зміни розміру пакетів, тоді як для обмежених середовищ застосовується спрощення моделі до меншої конфігурації через дистиляцію, зберігаючи при цьому єдині інтерфейси для підміни ваг без змін у конвеєрі.

Таблиця 1.2 узагальнює порівняльний аналіз моделей розпізнавання подій за типом архітектури, їхніми перевагами та недоліками, а також характерними сферами застосування в задачах багатокласової класифікації та NER.

Таблиця 1.2

Порівняльний аналіз моделей розпізнавання подій

Модель	Тип	Переваги	Недоліки	Застосування
XLM-R (base/ large)	Encoder	Міжмовне узагальнення, UA/RU підтримка	Велика затримка	Багатокласова класифікація, NER [81, 96]
mDeBER Ta-v3	Encoder	Покращені представлення, точніший NER	Потребує більше пам'яті	Багатокласова класифікація, складні задачі для NER [82]
fastText + bi-LSTM	Seq2Seq	Швидкість, підходить для малих наборів	Нижча точність на складних текстах	Використання у якості моделі для оцінки інших моделей, ресурсно обмежені сценарії

1.5. Аналіз вимог до програмного забезпечення

Аналіз наукової літератури вказує прогалини в наявних програмних системах екологічного моніторингу:

(1) нестача перевірених методів та корпусів для морфологічно-насичених мов у кризовому домені, особливо для формату месенджера Telegram;

(2) відсутність відтворюваних конвеєрів, що поєднують морфологічно адаптоване розпізнавання іменованих сутностей, розв'язання топонімічної неоднозначності та об'єднання з сенсорними або супутниковими даними із збереженням походження у стандартизованому вигляді;

(3) потреба у формальних підходах до оцінки невизначеності та усунення дублювання повідомлень у мультиджерельних потоках даних;

(4) потреба в архітектурах для автономної роботи та механізмах автоматичного поповнення локальних автономних реєстрів географічних назв історичними та адміністративними варіантами назв.

Ці прогалини визначають фронт дослідження для розробки програмного забезпечення, що поєднуватиме морфологічно адаптоване NLP, геокодування та об'єднання з SaveEcoBot-подібними сенсорами.

Для розроблення алгоритмічного та програмного забезпечення для виявлення та прогнозування екологічних загроз на основі аналізу гетерогенних поточних даних доцільно поєднати кілька підходів з проаналізованих платформ.

Розроблюване ПЗ повинно забезпечувати уніфіковане подання та геопросторове тегування подій, що дає змогу інтегрувати гетерогенні джерела даних у спільному просторово-часовому контексті. Система має реалізовувати автоматизований збір та багатомовний NLP-аналіз поточних даних, включно з OSINT-повідомленнями та мультимедійними свідченнями, з метою оперативного виявлення, фільтрації та семантичного збагачення подій.

Програмне забезпечення повинно підтримувати верифікацію та контроль якості подій шляхом поєднання автоматичних перевірок та формалізованих структур даних, а також інтеграцію сенсорних і дистанційних спостережень для підвищення достовірності виявлення загроз та зменшення хибних спрацювань.

Оброблення даних має здійснюватися в межах єдиного асинхронного мікропакетного конвеєра з детермінованою уніфікацією часових міток, одиниць вимірювання та схем подій без використання зовнішніх брокерів повідомлень, що забезпечує низький час відгуку та високу пропускну здатність. Аналітичні модулі повинні забезпечувати геопросторову агрегацію та прогнозування ризиків з використанням ієрархічних сіток та методів ML, поєднуючи поточні та прогнозні оцінки у межах єдиного просторово-часового представлення для переходу до проактивної підтримки прийняття рішень.

На основі проведеного аналізу наявних програмних систем моніторингу було сформульовано функціональні та нефункціональні вимоги до програмного

забезпечення, призначеного для виявлення та прогнозування екологічних загроз на основі аналізу гетерогенних потокових даних.

Функціональні вимоги

FR-1. Збір гетерогенних потокових даних.

Система повинна забезпечувати приймання потокових даних з відкритих та офіційних джерел, зокрема, текстових OSINT-повідомлень, супутникових та безпілотних спостережень, а також даних сенсорних IoT-мереж (радіаційних та метеорологічних), шляхом інтеграції через API.

FR-2. Мікропакетне оброблення та уніфікація даних.

Система повинна виконувати мікропакетне оброблення подій з приведенням часових міток до єдиної часової бази UTC, просторових координат до шестикутної сітки НЗ визначеного рівня роздільності та значень показників до уніфікованих одиниць вимірювання.

FR-3. Багатомовний аналіз текстових даних.

Система повинна здійснювати автоматичний багатомовний аналіз текстових повідомлень, включаючи класифікацію подій, виявлення сутностей, оброблення заперечень та нормалізацію словоформ, з формуванням структурованого подання події.

FR-4. Геопросторове прив'язування подій.

Система повинна реалізовувати каскадний механізм геокодування, що поєднує точне, нечітке, транслітераційне та семантичне зіставлення топонімів із подальшою просторовою прив'язкою до осередків НЗ та узгодженням із супутниковими та сенсорними спостереженнями.

FR-5. Аналітика та прогнозування екологічних ризиків.

Система повинна забезпечувати розрахунок агрегованих просторово-часових показників екологічного ризику та формування прогнозних оцінок розвитку загроз на основі методів машинного навчання.

FR-6. Візуалізація та доступ до результатів.

Система повинна надавати засоби візуалізації результатів аналізу у вигляді інтерактивної карти з просторовими агрегатами, підтримувати фільтрацію та пошук, забезпечувати експорт даних та доступ до них через відкриті формати та програмні інтерфейси.

Нефункціональні вимоги (згідно ISO/IEC 25010)

NFR-1. Сумісність.

Система повинна підтримувати інтеграцію з зовнішніми інформаційними ресурсами через стандартизовані API та відкриті формати обміну даними. Метрика: частка успішно інтегрованих інтерфейсів $\geq 90\%$.

NFR-2. Захист даних і доступу.

Система повинна забезпечувати мінімізацію оприлюднених полів, розділення публічних та внутрішніх подань, контроль доступу до конфіденційних даних, аудит-лог доступу та криптографічний захист даних під час передавання та (за потреби) зберігання. Метрика: 100% шифрування даних; відсутність критичних вразливостей відповідно до OWASP Top 10.

NFR-3. Доступність інтерфейсу.

Інтерфейс користувача повинен бути адаптивним та відповідати вимогам доступності.

Метрики:

- коректне відображення на різних типах пристроїв $\geq 99\%$;
- повна локалізація інтерфейсу;
- відповідність WCAG 2.1 AA $\geq 90\%$.

NFR-4. Доступність та відмовостійкість.

Система повинна забезпечувати відтворюваність результатів оброблення та можливість відновлення стану шляхом збереження повної історії подій, партиціювання журналу та формування періодичних знімків агрегованих даних.

Метрики: uptime $\geq 99,5\%$; MTTR ≤ 15 хв; RPO ≤ 15 хв.

NFR-5. Швидкодія та масштабованість.

Система повинна забезпечувати оброблення великих обсягів подій з обмеженою затримкою та підтримувати горизонтальне масштабування.

Метрики:

- ≥ 1 млн. подій/добу;
- 95-й перцентиль затримки ≤ 5 с;
- коефіцієнт масштабованості $\geq 0,9$.

NFR-6. Модульність та супроводжуваність.

Архітектура системи повинна забезпечувати модульне розширення через уніфіковані контракти модулів (адаптери джерел, гармонізація, аналітика), наявність документації та автоматизовані процедури розгортання.

Метрики: час інтеграції нового модуля ≤ 8 год; 100% опису API; автоматизований CI/CD-процес.

NFR-7. Правова, ліцензійна та ресурсна відповідність.

Система повинна відповідати чинним правовим нормам, вимогам ліцензування даних та принципам раціонального використання обчислювальних ресурсів. Метрики: відсутність зафіксованих порушень; середнє завантаження CPU/GPU $\leq 70\%$.

Наведені вимоги сформульовані відповідно до міжнародного стандарту ISO/IEC 25010:2011 (SQuaRE) з урахуванням принципів специфікації вимог ISO/IEC/IEEE 29148:2018, що забезпечує їх структурованість, вимірюваність та перевірюваність.

1.6. Висновки до розділу 1

Виконаний аналіз наукових робіт та програмних рішень показав, що для екологічних загроз існують підходи, які охоплюють використання ГІС та автоматизованих засобів перевірки інформації. У межах аналізу програмних рішень для моніторингу екологічних загроз було розглянуто як міжнародні програмні системи (ACLED, GDELT, Liveuamap), так і національні системи моніторингу (EcoZagroza, Ecodozor, SaveEcoBot), було виявлено їх переваги та

обмеження, зокрема, у сфері інтеграції мультимодальних OSINT-джерел, оперативного геокодування та прогнозої аналітики.

Оцінка методів та технологій оброблення OSINT-джерел показала, що найбільш результативними є комбіновані підходи, які поєднують NLP-аналіз, геопросторове моделювання та гібридні моделі, здатні обробляти великі гетерогенні потоки даних у реальному часі.

Аналіз наявних програмних систем моніторингу екологічних загроз підтвердив наукову та практичну актуальність розроблення алгоритмічних та програмних засобів для аналізу та прогнозування екологічних загроз на основі гетерогенних поточкових даних, що важливо в умовах воєнних дій та підвищених екологічних ризиків, коли оперативна інформація та її точна інтерпретація можуть рятувати життя.

Було сформульовано функціональні та нефункціональні вимоги до розроблення програмного забезпечення. Аналіз вимог виявив необхідність створення модульної, масштабованої та безпечної системи з підтримкою інтеграції зовнішніх API, автоматизованою перевіркою, багатомовним інтерфейсом та дотриманням міжнародних стандартів обміну геоданими.

Програмне забезпечення має забезпечувати мікропакетне приймання поточкових даних, уніфікований конвеєр гармонізації та двошарову просторову агрегацію на основі сітки НЗ для перетворення гетерогенних OSINT-даних в узгоджені просторово-часові події з оцінюванням їх достовірності та рівня небезпеки та інтеграцією наземних сенсорних спостережень з метою підтримки превентивних рішень.

РОЗДІЛ 2. МЕТОДИ ТА ПРОГРАМНІ ЗАСОБИ АНАЛІЗУ ГЕТЕРОГЕННИХ ПОТОКОВИХ ДАНИХ ДЛЯ ВИЯВЛЕННЯ І ПРОГНОЗУВАННЯ ЕКОЛОГІЧНИХ ЗАГРОЗ

2.1. Метод геокодування зі збереженням коректності для різних словоформ багатомовних OSINT-повідомлень

Актуальність геокодування текстових згадок локацій у морфологічно насичених мовах, зокрема, українській, зумовлена необхідністю забезпечення коректної просторової прив'язки повідомлень з гетерогенних поточкових джерел під час виявлення та прогнозування екологічних загроз, оскільки словозмінні форми топонімів у природномовних повідомленнях істотно ускладнюють їх однозначне зіставлення з канонічними записами географічних реєстрів, таких як GeoNames, без попередньої морфологічної нормалізації.

За відсутності такої нормалізації зростає кількість помилок просторової агрегації даних, що призводить до спотворення геоінформаційного аналізу, зниження точності локалізації потенційних джерел екологічних загроз і, як наслідок, до помилкових управлінських рішень у системах раннього попередження, які функціонують у режимі реального часу та інтегрують дані з текстових повідомлень, сенсорних мереж і зовнішніх інформаційних сервісів.

Геокодування текстових згадок локацій у морфологічно насичених мовах, таких як українська, стикається з проблемою словозмінних форм: назви населених пунктів у різних відмінках можуть суттєво відрізнятися від канонічної форми у реєстрі географічних назв.

Наприклад, згадки «у Києві», «Харківського району» або «на Львівщині» потребують морфологічної нормалізації перед зіставленням з базою GeoNames.

Для вирішення цієї задачі було розроблено метод, який поєднує лексичне, фонетичне та семантичне зіставлення у послідовності з наростаючою обчислювальною складністю (рис. 2.1).

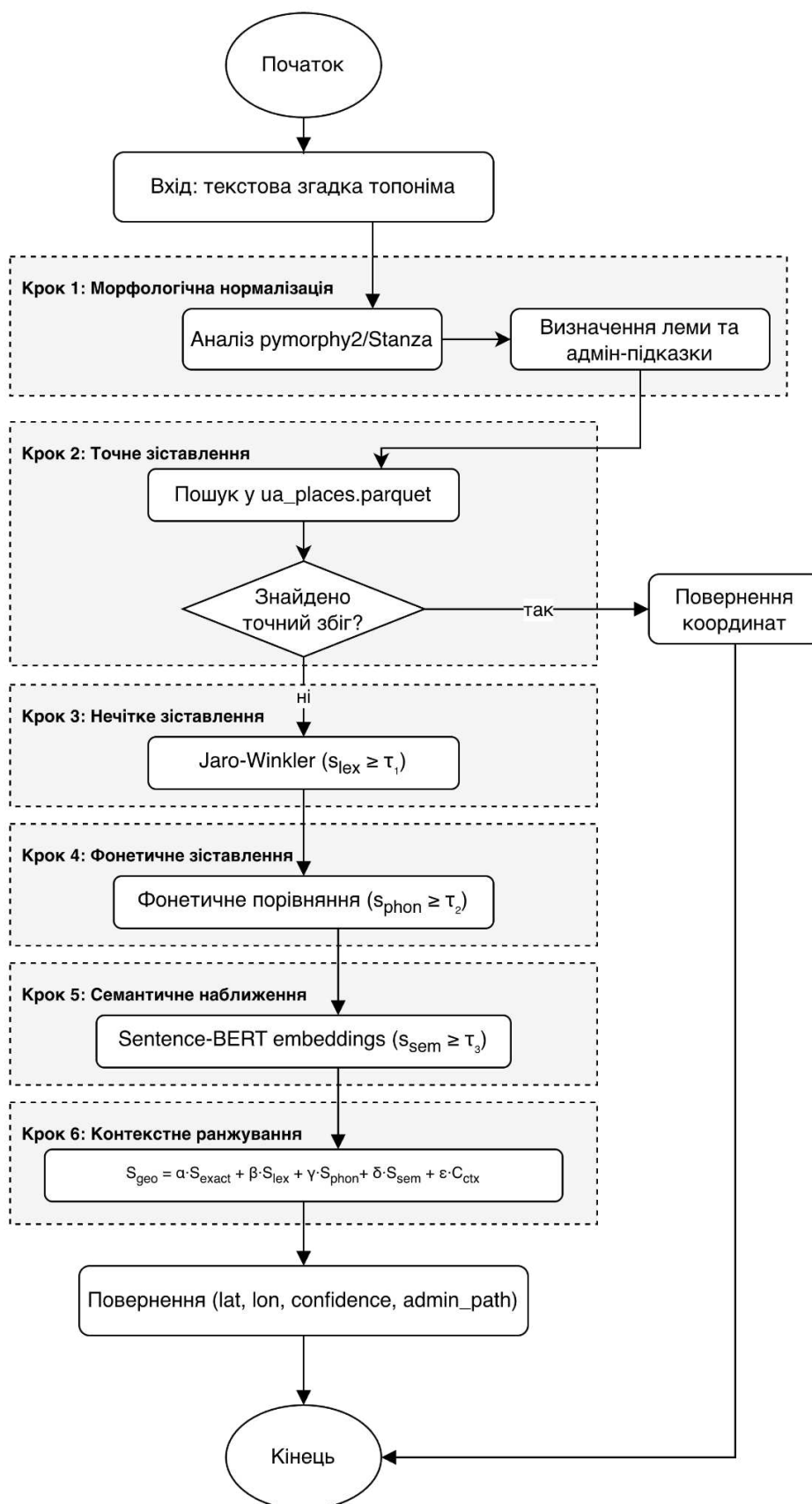


Рис. 2.1. Схема запропонованого методу геокодування

Розглянемо основні кроки виконання запропонованого методу.

Крок 1. Морфологічна нормалізація. На вхід подається текстова згадка топоніма. Виконується приведення поверхневої форми до нормалізованої леми засобами морфологічного аналізатора східнослов'янської групи мов (у даній роботі використовуються аналізатори для української та ворожого мови: `rumorphy2` та `Stanza/UDPipe`). Визначається адміністративна підказка (`city`, `region`, `district`) зі скорочень типу «м.», «обл.», «р-н».

Крок 2. Точне зіставлення. Виконується пошук ідентичного збігу нормалізованої леми із записами у реєстрі географічних назв `ua_places.parquet` (понад 60 000 записів). Якщо знайдено єдиний точний збіг, процес завершується успішно з поверненням координат.

Крок 3. Нечітке зіставлення. При відсутності точного збігу активується нечітка відповідність з використанням метрики фонетичної подібності Джаро-Вінклера. Обчислюється оцінка нечіткої подібності s_{lex} з порогом $\tau_1 \approx 0,85$. Знаходяться кандидати, близькі за написанням.

Крок 4. Фонетичне зіставлення. Застосовується фонетичне порівняння s_{phon} з порогом $\tau_2 \approx 0,8$ для захоплення подібності у звучанні, що важливо для усних джерел або транслітераційних варіантів.

Крок 5. Семантичне наближення. Для складних випадків застосовується Sentence-BERT: генеруються векторні представлення для згадки та кандидатів, обчислюється косинусна подібність [97] s_{sem} з порогом $\tau_3 \approx 0,6-0,7$.

Крок 6. Контекстне ранжування. Обчислюється загальна оцінка геокодування.

Крок 7. Повернення результату. Повертається кортеж (`lat`, `lon`, `confidence`, `admin_path`), де `confidence` $\in [0;1]$ – рівень впевненості, `admin_path` – адміністративний шлях («Україна/Київська область/Київ»).

Каскад починається з лематизації – приведення поверхневих форм до нормалізованої леми за допомогою морфологічних аналізаторів (`Stanza`, `UDPipe`).

Після лематизації виконується точна відповідність: спроба знайти точний збіг нормалізованої лема із записами у реєстрі географічних назв. Якщо точний збіг знайдено і він є унікальним, процес завершується успішно. Однак у багатьох випадках точної відповідності не вистачає через історичні назви, альтернативні аліаси або помилки у написанні.

Наступним етапом є нечітке зіставлення за допомогою метрик редагувальної відстані Левенштейна та фонетичної подібності Джаро-Вінклера. Ці метрики дозволяють знаходити кандидатів, які близькі до нормалізованої лема за написанням або звучанням (табл. 2.1), що особливо корисно при опечатках або транслітераційних варіантах.

Обчислюється оцінка нечіткої подібності s_{lex} на основі комбінації цих метрик. Додатково застосовується фонетичне зіставлення s_{phon} для захоплення подібності у звучанні, що важливо для усних джерел або неформальних згадок.

Таблиця 2.1

Приклади інфлексій та нормалізації

Поверхнева форма	Нормалізована лема	Адмін-підказка	Тип інфлексії / нормалізації
Києві	Київ	місто	місцевий відмінок (називний)
в Одесі	Одеса	місто	прийменникова конструкція
Харківського району	Харківський район	район	узгодження з адмін-шляхом
м. Івано-Франківськ	Івано-Франківськ	місто	розгортання «м.»
Львівщині	Львівська область	область	регіональна назва (офіційна)

Для складних випадків, де лексичне та фонетичне зіставлення не дають впевненого результату, застосовується семантичне наближення через Sentence-BERT вбудовані представлення. Модель генерує вектори для згадки та кандидатів з реєстру, після чого обчислюється косинусна подібність s_{sem} .

Це дозволяє захопити контекстну близькість навіть за відсутності поверхневої схожості – наприклад, «столиця» може бути семантично пов'язана з «Київ» у відповідному контексті.

Фінальний етап включає контекстне ранжування s_{ctx} , яке враховує часову та просторову близькість до підтверджених подій, адміністративну ієрархію та частотність згадок у потоці. Загальна оцінка геокодування обчислюється як зважена сума всіх компонентів:

$$score_{geo} = w_{lex} \cdot s_{lex} + w_{phon} \cdot s_{phon} + w_{sem} \cdot s_{sem} + w_{ctx} \cdot s_{ctx}, \quad (2.1)$$

де s_{lex} – лексична схожість, s_{phon} – фонетична, s_{sem} – семантична, s_{ctx} – контекстна узгодженість (адміністративні підказки, близькість до підтверджених подій/сенсорів), а ваги w_{lex} , w_{phon} , w_{sem} , w_{ctx} нормалізовані до одиниці і відображають відносний внесок кожного етапу.

Це дозволяє гнучко налаштовувати баланс між швидкістю та точністю: для більшості згадок достатньо лексичних перевірок на ранніх етапах, тоді як складні випадки проходять через семантичне та контекстне ранжування.

Каскадна стратегія дає змогу послідовно зменшувати простір пошуку: від високопродуктивних лексичних перевірок до дорожчих семантичних порівнянь лише для складних випадків (табл. 2.2).

Таблиця 2.2

Операційні пороги та результати відсіювання

Етап	Міра	Типовий поріг	Очікуване відсіювання
Лексичний (чіткий/частковий)	нормалізований збіг	—	60–80%
Нечіткий (Левенштейн/Джаро–Вінклер)	s_{lex}	$\tau_1 \approx 0,85$	+10–20%
Фонетичний	s_{phon}	$\tau_2 \approx 0,8$	+3–8%
Семантичний	s_{sem}	$\tau_3 \approx 0,6–0,7$	+5–10%
Контекстний	s_{ctx}	τ_{ctx} сценарний	«тонке» ранжування/ перевірка

Це зберігає пропускну здатність у режимах мікропакетів та забезпечує стабільну затримку, адже більшість згадок розв'язуються на ранніх етапах. Крім

того, прозорий формат оцінки дозволяє оперативно вводити доменні правила (наприклад, пріоритет офіційних перейменувань чи фільтр історичних назв поза релевантним часовим контекстом) без перепідготовки семантичної моделі.

Приклад нормалізації контекстної оцінки (без зміни основної формули): s_{ctx} може бути розкладений на зважену суму індикаторів:

$$s_{ctx} = \alpha_{admin} \cdot I(admin_match) + \alpha_{time} \cdot \exp\left(-\frac{\Delta t}{\lambda}\right) + \alpha_{dist} \cdot \exp\left(-\frac{d}{R}\right) + \alpha_{sens} \cdot I(sensor_hit), \quad (2.2)$$

де $I(*)$ – індикатор (0/1), Δt – часовий зсув від останньої підтвердженої активності, d – відстань до сенсорної події, λ та R – масштаби згасання, α_* – вагові коефіцієнти. Така декомпозиція не змінює визначення балу $score_{geo}$, але робить внесок контексту керованим та тлумачним.

Оцінювання якості доцільно проводити на кількох рівнях допуску: «@5 км», «@10 км» та «@адмін-одинаця». Для унікальних назв, присутніх у реєстрі географічних назв, очікується вища точність на малих допусках; для непрямих або історичних згадок доцільно оцінювати успіх на рівні адміністративної одиниці (район/область), що краще відповідає практичним сценаріям використання агрегованих НЗ-тайлів. Важливим етапом є аудит неоднозначних випадків і накопичення складних прикладів для подальшого вдосконалення порогів τ_1, τ_2, τ_3 та ваг $w_{lex}, w_{phon}, w_{sem}, w_{ctx}$.

2.2. Метод багатоджерельної перевірки та фільтрації якості подій

Надійність системи моніторингу критично залежить від її здатності відокремлювати достовірні події від шуму та дезінформації.

Запропонований метод багатоджерельної перевірки інтегрує текстові OSINT-повідомлення з фізичними сенсорними даними та застосовує складені оцінки достовірності для фільтрації подій перед їх публікацією у просторово-часових поданнях.

Розглянемо основні кроки запропонованого методу.

Крок 1. Приймання події. На вхід надходить OSINT-повідомлення або сенсорне спостереження. Визначається тип джерела та його профіль надійності $Reliability(src) \in [0;1]$ [92].

Крок 2. NLP-класифікація. Для текстових повідомлень виконується класифікація типу події (strike, fire, alert тощо) з отриманням ймовірності $P(event)$ від моделі XLM-RoBERT.

Крок 3. Геокодування. Застосовується каскадний метод геокодування (п.2.1) з отриманням $geocode_confidence \in [0;1]$.

Крок 4. Просторово-часове узгодження. Пошук сенсорних підтверджень у 15-хвилинних вікнах на сітці НЗ r10. Якщо знайдено FRP-детекцію або γ -аномалію у тому ж шестикутнику та часовому вікні, встановлюється $evidence_match = 1$.

Крок 5. Обчислення складеної оцінки.

Крок 6. Перевірка кворуму. Для важливих подій застосовується правило незалежного кворуму: 2 текстові джерела АБО 1 текст та сенсор.

Крок 7. Усунення дублювання. Виявляються дублікати в радіусі та інтервалі близькості (Δt , Δr), виконується об'єднання атрибутів.

Крок 8. Публікація. Події з $score \geq 0,5$ та $geocode_confidence \geq 0,3$ публікуються зі статусом verified; інші переводяться у стан pending.

База знань структурується як інтегрований набір довідників, профілів джерел та агрегованих фізичних ознак, уніфікованих на решітці НЗ. До її складу входять: багатомовний реєстр географічних назв з адміністративними ієрархіями, аліасами та центроїдами; тематичні словники інфраструктурних об'єктів та небезпек; профілі джерел з емпіричними оцінками надійності; просторово-часові шари сенсорних індикаторів (щільність FRP, аномалії γ), що забезпечують можливість перехресного підтвердження з текстовими подіями.

Єдині ключі НЗ дозволяють зберігати та обчислювати всі показники в узгодженій системі координат для подальшої фільтрації, агрегування ризику та планувальних задач.

Склад і роль компонентів бази знань підсумовано в табл. 2.3.

Таблиця 2.3

Компоненти бази знань та їх використання у фільтрації

Компонента	Зміст	Ключі/ ідентифікатори	Роль у фільтрації
Реєстр географічних назв	Топоніми, аліаси, адмін-шляхи, центроїди	geoname_id / local_id, НЗ	Кандидати геокодування, нормалізація
Профілі джерел	Тип, мова, емпірична достовірність	source_id	Reliability(src) та кворум
Сенсорні шари	FRP (FIRMS), γ -аномалії (z)	НЗ, час	evidence_match, FRP_sum, γ_z
Словники домену	Інфраструктура, небезпеки	категорії	Контекстні ознаки, фільтри

Фільтрація подій базується на складеному балі достовірності, який поєднує властивості джерела, вихідні ймовірності з NLP-моделей, впевненість геокодування та факт наявності сенсорного підтвердження у відповідному часово-просторовому вікні.

Така інтегративна оцінка узгоджується з підходами багатоканальної перевірки, де перехресні свідчення знижують ризик хибних спрацювань та підвищують відтворюваність рішень.

Складена оцінка обчислюється за формулою:

$$\begin{aligned}
 score = & w_{src} \cdot Reliability(src) + w_{nlp} \cdot P(event) + \\
 & + w_{geo} \cdot geocode_confidence + w_{phys} \cdot evidence_match
 \end{aligned}
 \tag{2.3}$$

де $Reliability(src) \in [0;1]$ характеризує довіру до джерела, $P(event)$ – прогноз ймовірності події з NLP-класифікатора, $geocode_confidence \in [0;1]$ –

«впевненість» геокодування, $evidence_match \in \{0,1\}$ – індикатор наявності сенсорного підтвердження у заданому вікні.

Для важливих подій використовується правило незалежного кворуму (два текстові джерела або комбінація тексту та даних з сенсора).

При виявленні дублікатів у радіусі та інтервалі близькості застосовується «м'яке» об'єднання атрибутів, зберігаючи повний журнал джерел та пояснювальну інформацію.

Використані величини для розрахунку ризику і попиту наведено в табл. 2.4.

Таблиця 2.4

Критерії фільтрації та публікації подій

Критерій	Правило/поріг	Дія
Складений бал score	$score \geq 0,5$	Дозволити публікацію
“Впевненість” геокодування	$geocode_confidence < 0,3$	Статус pending (не впливає на ризик)
Кворум підтверджень	2 текстові або 1 текст + сенсор	Статус verified і включення в n_hits
Усунення дублювання	ближче заданого $\Delta t, \Delta r$	«М'яке» об'єднання атрибутів, єдина подія

Добова оцінка ризику на НЗ $r=7$ визначається як адитивна комбінація компонент OSINT та сенсорних сигналів. Для пожежної компоненти застосовується логарифмування FRP після усунення дублювання, з коефіцієнтом $w(h, t)$ ($wind_align$), що враховує напрям та швидкість вітру.

Компонента γ -випромінювання обмежується до додатних аномалій:

$$risk = 1,0 \cdot N_{hits} + 0,6 \cdot \ln(1 + FRP_{sum}) + 0,3 \cdot \max(\gamma_{poz}, 0), \quad (2.4)$$

де N_{hits} – кількість підтверджених подій (удар/пожежа) в осередку за добу, FRP_{sum} – спочатку береться $\max(FRP)$ у 15-хвилинних вікнах на НЗ $r10$ з подальшою добовою сумою на НЗ $r7$ (FRP_{sum}), γ_{poz} – позитивний z-індекс γ (γ_{pos}) з 14-денною базою. Попит визначається як добуток ризику на чисельність населення в тайлі: $demand = population(h3) \times risk$.

Для вибору місць інтервенцій використовується задача максимального покриття MCLP, яка максимізує сумарний покритий попит у заданому радіусі доступності при обмеженні на кількість нових об'єктів p [100].

Використані величини для розрахунку ризику та попиту наведено в табл.2.5.

Таблиця 2.5

Компоненти ризику та попиту для агрегування на НЗ $r=7$

Величина	Опис	Джерело/обчислення	Використання
N_{hits}	Підтверджені події за добу	verified-журнал	Формула 2.4
FRP_{sum}	Добова сума пожежної потужності	FIRMS $\rightarrow$ НЗ $r=7$	Формула 2.4
γ_{poz}	Нормалізована γ -аномалія	добові z-оцінки $\rightarrow$ НЗ $r=7$	Формула 2.4
population(h3)	Населення в тайлі	демографічний шар $\rightarrow$ НЗ $r=7$	demand = population $\times$ risk

Операційні політики публікації та контролю якості задаються порогами на складеній оцінці, рівні «впевненості» геокодування та кворумі підтверджень (табл. 2.6).

Таблиця 2.6

Типові помилки обробки та заходи нейтралізації

Симптом	Потенційна причина	Наслідок	Міра протидії
Висока оцінка без сенсорів	Надійне джерело та високе значення $P(event)$	Хибнопозитивна класифікація	Вимога кворуму або підвищення w_{phys} у чутливих сценаріях
Низька оцінка при сенсорному збігу	Слабке джерело або низьке значення $P(event)$	Втрата подій	Мінімальна підстраховка сенсорами (evidence_match)
Помилки геокодування	Низьке значення geocode confidence	Хибні n_{hits}	Pending до уточнень; контекстна перевірка
Дублювання подій	Близькі час/місце	Завищення n_{hits}	«М'яке» об'єднання і унікалізація кластерів

Події з недостатнім значенням *score* або *geocode_confidence* переводяться у стан очікування (*pending*) до появи нових свідчень; *verified*-події потрапляють до

агрегатів ризику. Така дисципліна забезпечує надійність карти ризику та знижує чутливість до інформаційних вкидів, водночас допускаючи «реабілітацію» слабо підтверджених повідомлень у разі сенсорного підтвердження.

Рис. 2.2 ілюструє наскрізний конвеєр підтвердження події: від приймання повідомлень (даних від сенсорів) та розрахунку складеної оцінки через усунення дублювання в часово-просторовому вікні $\{\Delta t, \Delta r\}$, перевірку кворуму (незалежні текстові джерела або текст та сенсорні дані), присвоєння статусів *pending/verified* та публікацію на тайловому рівні H3 $r=7$ до подальшої агрегації за добу та обчислення ризику.

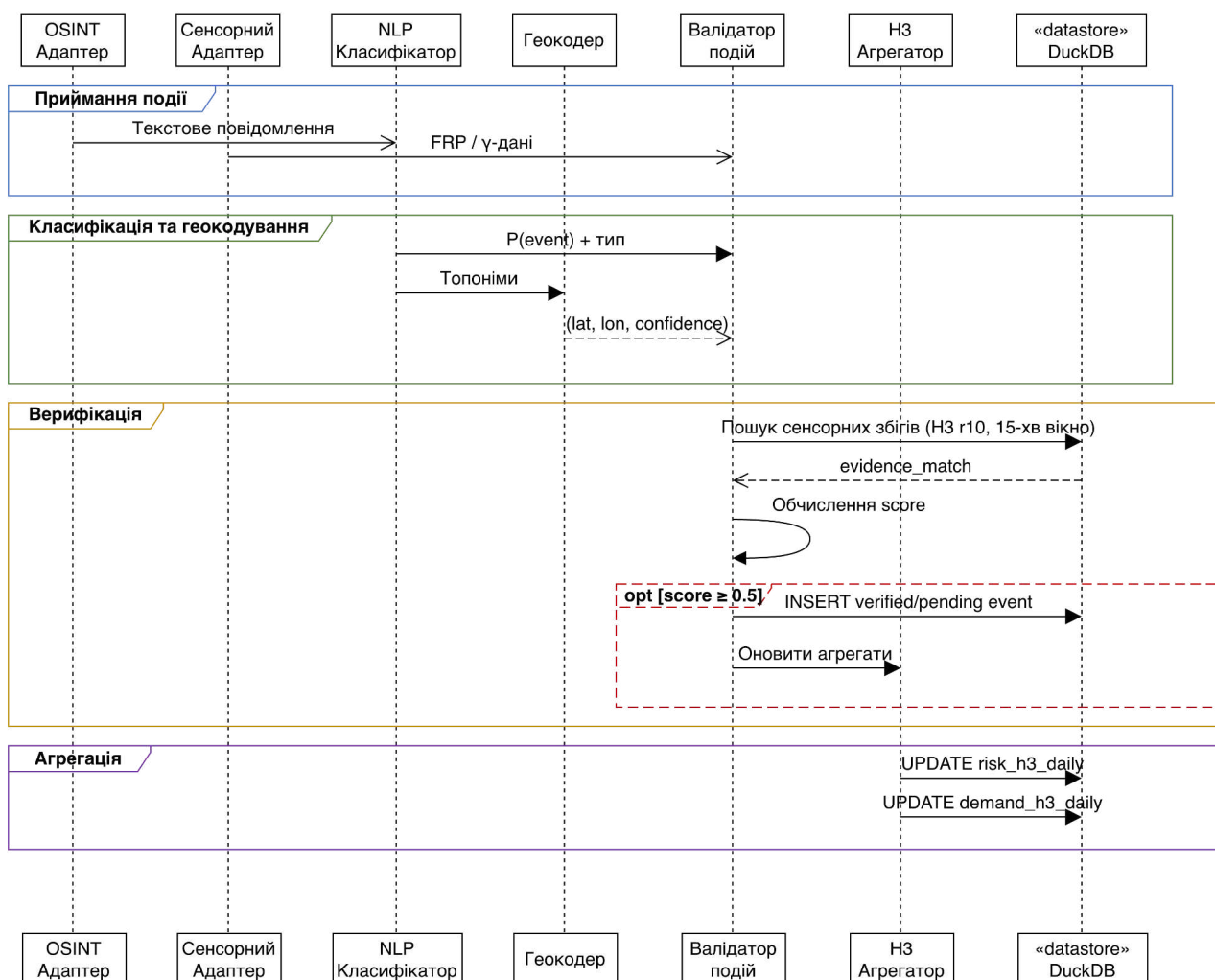


Рис 2.2. Діаграма послідовності підтвердження події запропонованого методу багатоджерельної перевірки та фільтрації якості подій

Діаграма рис. 2.2 відображає точки контролю якості («впевненість» геокодування, *evidence_match*), зворотні зв'язки для оновлення профілів джерел та ваг, а також шляхи затриманого підтвердження (подія отримує сенсорне підтвердження пізніше та підвищує свій статус без повторного приймання). Відбувається узгодження логіки прийняття рішень з журналом у DuckDB (*events/events_live*), сформовані *verified*-події переходять у добові НЗ-агрегати ризику та далі в розрахунок попиту та планувальні задачі.

2.3. Ймовірнісна геопросторова модель прийняття рішень на базі гетерогенних потоків даних

Під час аналізу наявних програмних систем моніторингу стану довкілля стало зрозуміло, що лише реактивне відображення поточного стану ризику недостатнє для оперативного управління.

Потрібно передбачати просторово-часову еволюцію загроз та перетворювати прогнози на конкретні рекомендації для рішень щодо укриттів та евакуації. Було поєднано дві взаємодоповнювальні ідеї: рекурентне нейронне прогнозування індикаторів ризику на основі послідовностей гетерогенних джерел (OSINT, FRP, γ-фон) та ймовірнісну геопросторову інтеграцію цих прогнозів з поточним ризиком на сітці НЗ. Така інтеграція дозволяє зменшити реактивність та перейти до проактивної логіки, у якій рішення ґрунтуються не лише на тому, що вже сталося, а й на тому, що з високою ймовірністю відбудеться найближчим часом.

Ймовірнісна геопросторова модель інтегрує результати геокодування, багатоджерельної перевірки та доменної агрегації на шестикутній сітці НЗ з підтримкою сценарних обмежень та оптимізації розміщення укриттів.

Розглянемо основні кроки моделі прийняття рішень (рис. 2.3).

Крок 1. Формування вхідного вектора ознак. Для кожного дня d та шестикутника h формується вхідний вектор ознак $x_{t,h}$.

Крок 2. LSTM-прогнозування. Застосовується LSTM-модель для прогнозування ризику на горизонті $\tau = 1-7$ діб: $risk_{pred}(h, d + \tau) = f_{LSTM}(x(h, d - W:d))$, де W – розмір вікна історії.

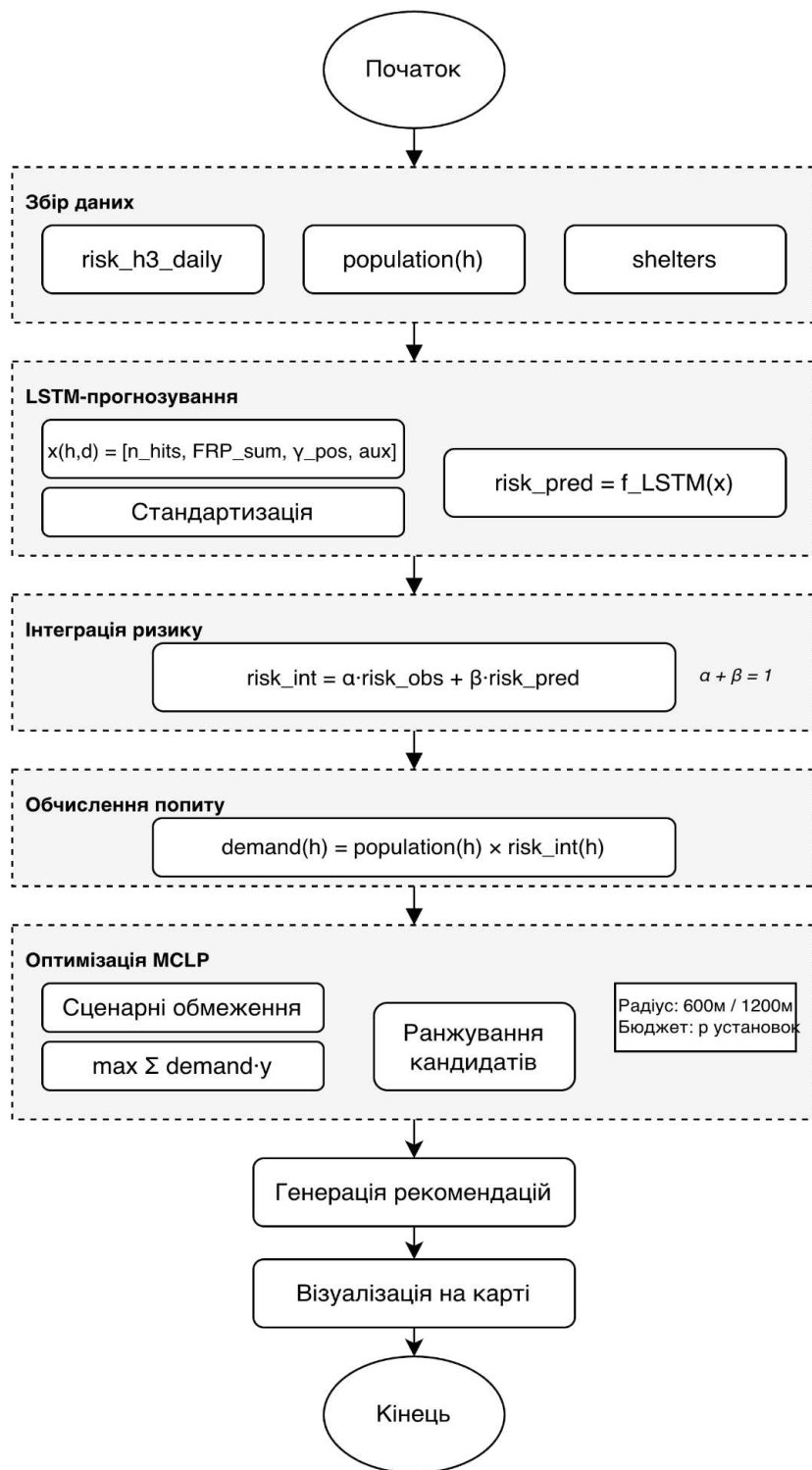


Рис. 2.3. Діаграма процесу прийняття рішень

Крок 3. Інтеграція прогнозів. Обчислюється інтегральний ризик:

$$risk_{int}(h, d) = \alpha \cdot risk_{obs}(h, d) + \beta \cdot risk_{pred}(h, d), \text{ де } \alpha + \beta = 1, \text{ ваги}$$

налаштовуються емпірично.

Крок 4. Обчислення попиту:

$$demand(h) = population(h) \times risk_{int}(h, d),$$

де *population* – чисельність населення шестикутника за даними Держстату.

Крок 5. Оптимізація MCLP. Застосовується алгоритм Maximum Covering Location Problem для максимізації покритого попиту: $\max \sum demand(h) \cdot y(h)$ за обмежень радіусу доступності та бюджету.

Крок 6. Генерація рекомендацій. Формуються конкретні координати для розміщення нових укриттів з оцінкою приросту покриття.

2.3.1. Метод інтеграції LSTM-прогнозування у конвеєр екологічного моніторингу

Події, які аналізуються (удари, пожежі, аномалії γ -фону), формують нерівномірні часові ряди з короткими імпульсами та сезонними хвилями. Такі ряди характеризуються різкими локальними змінами інтенсивності, що чергуються з тривалими ділянками відносного спокою, а це суттєво ускладнює їх математичний опис. Статистичні моделі, що покладаються на стаціонарність — зокрема ARIMA та її варіації — погано відтворюють нелінійні зв'язки між компонентами ризику, оскільки припускають сталість статистичних властивостей ряду в часі, що суперечить природі екологічних і безпекових подій. Крім того, такі підходи мають обмежену здатність враховувати взаємовплив різнорідних потоків даних, що надходять з асинхронних джерел. Натомість архітектура довгокороткострокової пам'яті LSTM забезпечує стійке «утримання контексту» на проміжках від днів до тижнів завдяки спеціалізованим механізмам вентилів, які вибірково зберігають або відкидають інформацію. Вона коректніше поводить себе з пропусками та різною довжиною послідовностей, що особливо важливо при роботі з гетерогенними потоками даних реального часу, і краще узгоджується з

мікропакетним обробленням, дозволяючи поступово оновлювати прогнози в міру надходження нових спостережень.

Розглянемо основні кроки алгоритму LSTM-інтеграції у конвеєр екологічного моніторингу.

Крок 1. Формування вхідного вектора. Для кожного дня d та шестикутника h формується вектор:

$$x(h, d) = \left[n_{hits(h, d)}, FRP_{sum}(h, d), \gamma_{pos}(h, d), aux(d) \right]. \quad (2.5)$$

де n_{hits} – кількість OSINT-подій, FRP_{sum} – сумарна пожежна потужність (MW), γ_{pos} – позитивна частина z-оцінки γ -фону, aux – допоміжні ознаки.

Крок 2. Стандартизація ознак. Кожна компонента стандартизується в межах шестикутника: $x_{norm} = \frac{(x - \mu)}{\sigma}$ для стабільності навчання.

Крок 3. LSTM-кодування. Застосовуються рівняння LSTM-клітинки:

$$\begin{aligned} i_t &= \sigma(W_i \tilde{x}_t + U_i h_{t-1} + b_i), \text{ – вхідні ворота;} \\ f_t &= \sigma(W_f \tilde{x}_t + U_f h_{t-1} + b_f), \text{ – ворота «забування»;} \\ \tilde{c}_t &= \tanh(W_c \tilde{x}_t + U_c h_{t-1} + b_c), \text{ – кандидатний стан комірки;} \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \text{ – оновлення стану комірки;} \\ h_t &= o_t \odot \tanh(c_t) \text{ – прихований стан;} \\ o_t &= \sigma(W_o \tilde{x}_t + U_o h_{t-1} + b_o), \text{ – вихідні ворота.} \end{aligned}$$

Крок 4. Прогнозування. Вихідний регресор генерує прогноз: $risk_{pred}(h, d + \tau) = W_{out} \cdot h_t + b_{out}$ для горизонту $\tau = 7$ діб (див. табл. 2.7).

Навчання виконується за втратами, що стримують вплив рідкісних викидів.

Крок 5. Вагова інтеграція. Поєднання з поточним ризиком: $risk_{int} = \alpha \cdot risk_{obs} + \beta \cdot risk_{pred}$, де $\alpha + \beta = 1$.

Крок 6. Адаптивне налаштування ваг. Ваги α , β коригуються на основі точності попередніх прогнозів для кожного регіону.

Таблиця 2.7

Параметри LSTM-моделі прогнозування

Параметр	Значення
Горизонт прогнозу τ	7 діб
Розмір вікна історії W	14 днів
Розмір прихованого стану	64 нейронів
Функція втрат	MAE / Huber Loss
Оптимізатор	Adam (lr=0,001)
Вагова схема	$\alpha + \beta = 1$, налаштовуються емпірично

На відміну від реактивних систем моніторингу, які лише відображають поточний стан середовища і реагують виключно на вже зафіксовані події, запропонований метод забезпечує:

- *проактивне попередження*: генерацію тривог на основі прогнозованого, а не лише спостережуваного ризику, що дозволяє завчасно інформувати відповідальних осіб і скорочувати час реакції на потенційно небезпечні ситуації до їх фактичного розвитку;
- *адаптивність*: автоматичне коригування вагових коефіцієнтів α , β залежно від точності прогнозів у конкретному регіоні, завдяки чому модель самостійно налаштовується на локальні особливості динаміки загроз і не потребує ручного перекалібрування при зміні умов спостереження;
- *інтеграцію з H3-DSMV*: прогнози зберігаються як окремий семантичний шар у просторовій структурі системи й залишаються доступними через той самий уніфікований API, що й решта компонентів, — це забезпечує взаємодію між модулями без додаткових адаптерів чи перетворень форматів.

Запропонований метод забезпечує перехід від реактивного до проактивного режиму моніторингу через інтеграцію LSTM-прогнозування у просторово-часовий конвеєр (рис. 2.4).

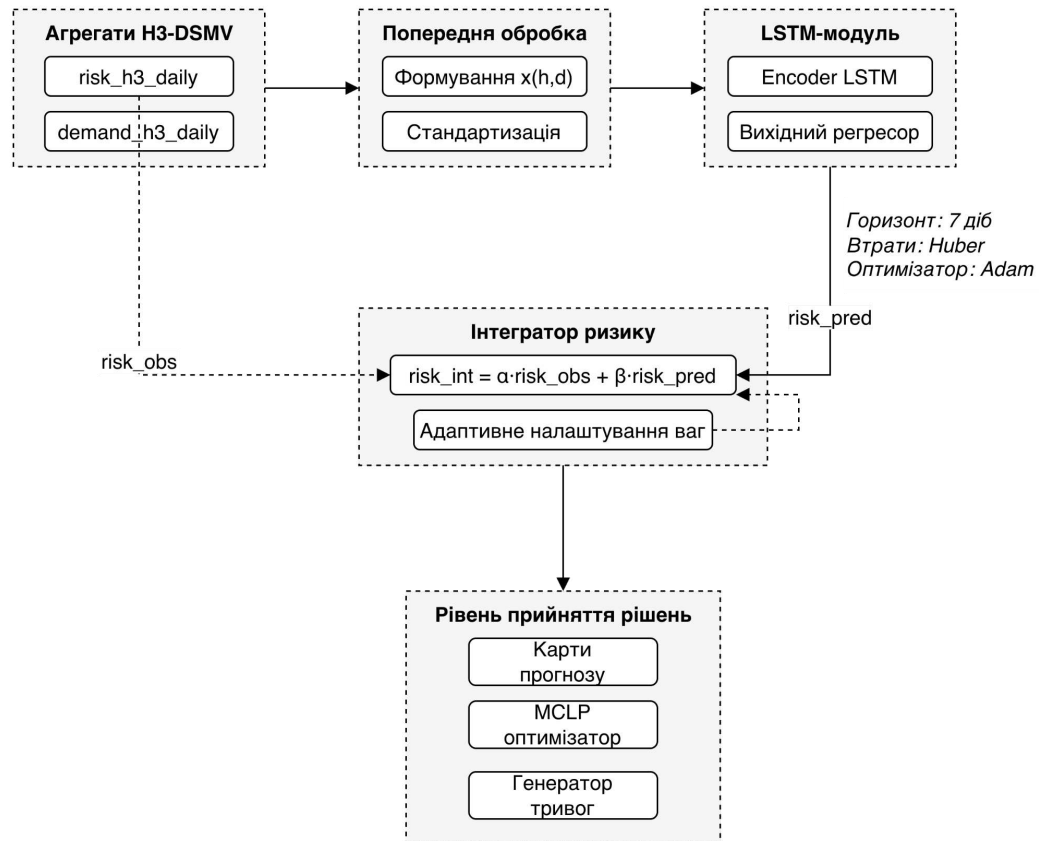


Рис. 2.4. Інтеграція LSTM-прогнозування у конвеєр моніторингу

2.3.2. Інтеграція прогнозів у ймовірнісну геопросторову модель ризику

Щойно прогноз стає доступним, система поєднує його з поточним оціненим ризиком, щоб сформувати інтегральний показник для прийняття рішень. Використовується вагова схема, у якій нова спостережувана інформація та прогнозований компонент мають керовані внески:

$$r_{t,h}^{(int)} = w_{cur} \cdot r_{t,h} + w_{fc} \cdot \hat{r}_{t+1,h} \quad (2.7)$$

де $w_{cur} + w_{fc} = 1$.

Ваги налаштовуються емпірично та можуть коригуватися зворотним зв'язком на основі точності попередніх прогнозів для кожного регіону. Оскільки

рішення мають просторовий контекст, інтегральний ризик інтерпретується як ймовірнісна інтенсивність для шестикутника h та поєднується з попитом в цьому шестикутнику, що формує пріоритезовану карту дій: шестикутники з великим значенням $r^{(int)}$ *хрор* отримують вищий пріоритет (рис. 2.5).

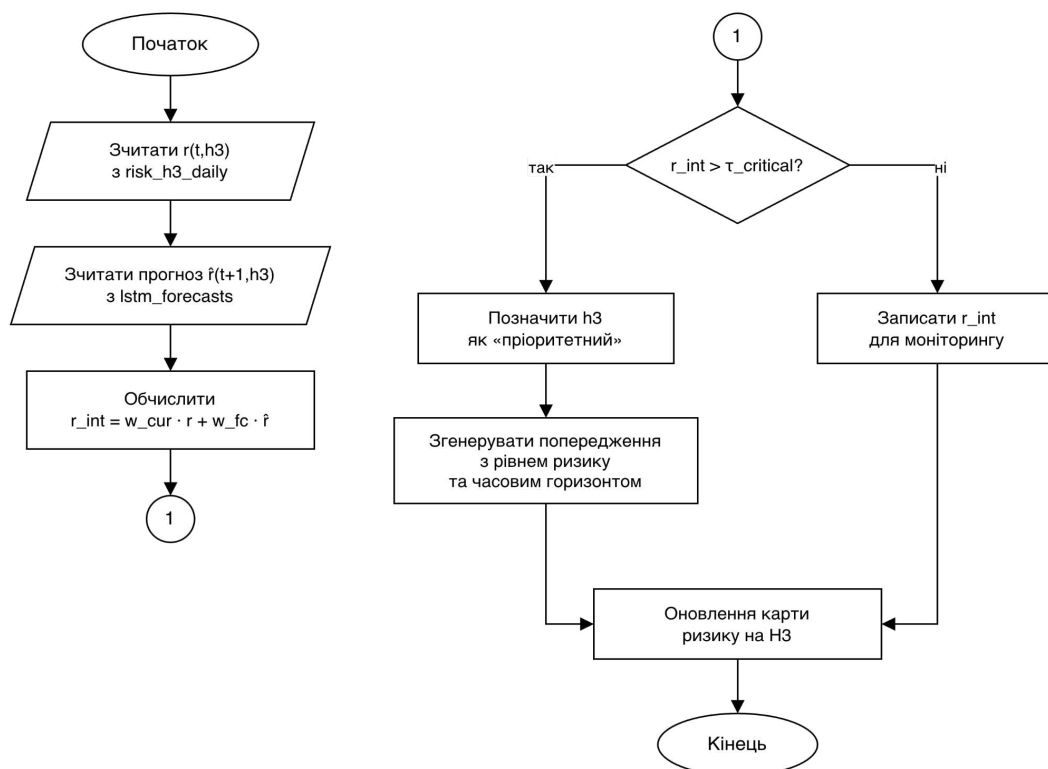


Рис. 2.5. Блок-схема інтеграції прогнозів у ймовірнісну модель ризику

У такій постановці, якщо прогноз невизначений, ваги зсувають баланс на користь поточного стану; якщо прогноз стабільно точний для певної місцевості, w_{fc} збільшується, що підсилює проактивність.

2.4. Процедури доменної агрегації ризику для гетерогенних потокових даних

Процедури доменної агрегації забезпечують уніфіковане оброблення різнорідних потоків даних з єдиною просторово-часовою основою на базі шестикутної сітки НЗ (рис. 2.6).

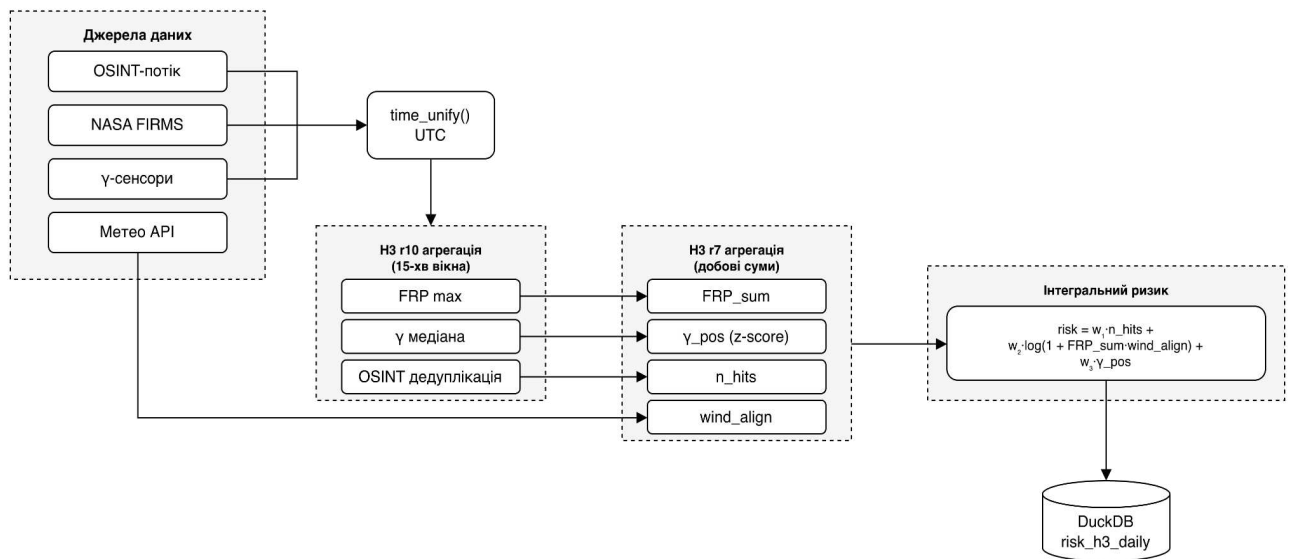


Рис. 2.6. Схема доменної агрегації ризику для гетерогенних потокових даних

Розглянемо основні кроки запропонованого алгоритму доменної агрегації ризику для гетерогенних потокових даних.

Крок 1. FRP-агрегація на H3 r10. Для кожного 15-хвилинного вікна у кожному шестикутнику H3 r10 обчислюється максимум FRP: $FRP_{window} = \max(FRP_i)$ для всіх детекцій у вікні.

Крок 2. Дедуплікація супутникових детекцій. Якщо в один шестикутник та часовий інтервал потрапляє кілька детекцій від різних супутникових проходів (MODIS, VIIRS), зберігається лише запис з максимальним FRP.

Крок 3. Добова агрегація на H3 r7. Виконується підсумовування унікальних 15-хвилинних максимумів за добу: $FRP_{sum} = \sum FRP_{window}(h,t)$ для всіх вікон доби.

Крок 4. Метеокорекція. Обчислюється коефіцієнт $wind_align(h,t)$ для врахування напрямку та швидкості вітру: $wind_align = \cos(\theta_{wind} - \theta_{neighbor}) \cdot (v_{wind} / v_{max})$, де θ – азимут, v – швидкості вітру.

Крок 5. Нормування радіаційних показників. Обчислюється z-індекс γ -випромінювання: $z(h,d) = (x(h,d) - \mu_{14}) / \sigma_{14}$, де μ_{14} , σ_{14} – середнє та стандартне

відхилення за 14 днів. Використовується лише позитивна частина: $\gamma_{pos} = \max(0, z)$.

Крок 6. Єдина часова база UTC. Усі часові мітки конвертуються до UTC функцією *time_unify(ts, tz_hint)* з валідацією діапазону (записи до 20.02.2022 відкидаються).

Крок 7. НЗ-узгодження. Усі показники індексуються на єдиній сітці НЗ з ієрархічним переходом $r10 \rightarrow r7$ для забезпечення міжрегіональної порівнюваності.

2.5. Вагові коефіцієнти системи

2.5.1. Визначення вагових коефіцієнтів системи

Вагові коефіцієнти системи можна поділити на три групи за методикою їх визначення та калібрування.

Група 1. Ваги формули геокодування (w_1, w_2, w_3, w_4 у формулі 2.1). Початкові значення ваг ($w_1=0,4, w_2=0,2, w_3=0,2, w_4=0,2$) визначались на основі емпіричного пріоритету етапів каскаду: лексичне зіставлення є найнадійнішим (найменша ймовірність хибного спрацювання) та отримує найбільшу вагу. Калібрування виконувалось на анотованому корпусі з 1500 речень шляхом максимізації точності геокодування @5 км на перевіірочній вибірці.

Група 2. Ваги формули складеної оцінки достовірності (формула 2.3). Коефіцієнти *reliability(src)*, *P(event)*, *geocode_confidence* та *evidence_match* поєднуються як добуток, де кожна компонента має діапазон [0;1].

Профілі надійності джерел формувались емпірично: для офіційних телеграм-каналів обласних адміністрацій це 0,95; для великих новинних каналів 0,80; для дрібних регіональних каналів – 0,65. Ці значення калібрувались шляхом зіставлення з підтвердженими подіями у ретроспективному наборі за період серпень 2022 - травень 2025 років.

Група 3. Ваги формули інтегрального ризику (w_{hits} , w_{FRP} , w_{rad} у формулі 2.4 та α , β у формулі 2.7). Для калібрування ваг інтегрального ризику використовувалось зіставлення з двома еталонними індикаторами.

Офіційні повітряні тривоги (дані Міністерства внутрішніх справ) – як зовнішній індикатор наявності реальної загрози. Досягнута кореляція $r = 0,78$ ($p < 0,001$) підтверджує адекватність вагової конфігурації.

Підтверджені OSINT-події з ретроспективною верифікацією – для перевірки того, що формула не пропускає значущі інциденти та не завищує ризик у спокійні періоди.

Ваги α та β (поточний та прогнозований ризик) калібруються адаптивно для кожного регіону на основі точності попередніх прогнозів: якщо LSTM-модель стабільно точна для певної місцевості, β збільшується (посилюється проактивність); при зниженні точності прогнозів β зменшується, і система більше покладається на поточні спостереження.

2.5.2. Оптимізація та процес підбору вагових коефіцієнтів

Підбір вагових коефіцієнтів здійснювався у три етапи, що поєднують емпіричну ініціалізацію, оптимізацію та валідацію на незалежних даних.

Етап 1. Емпірична ініціалізація. Початкові значення ваг встановлювались на основі доменних знань про природу джерел даних. Для формули ризику (2.4) ваги ініціалізувались пропорційно до надійності джерела: OSINT-повідомлення з підтвердженням ($w_{hits} = 0,4$) отримали найбільшу вагу як найбільш інформативні для ідентифікації типу та локації події; супутникові FRP-детекції ($w_{FRP} = 0,35$) – дещо меншу через часову затримку; радіаційна компонента ($w_{rad} = 0,25$) – найменшу через вищий рівень фонового шуму.

Етап 2. Алгоритмічна оптимізація. Для уточнення початкових ваг застосовувався метод Grid Search з перебором сітки значень при обмеженні нормалізації $\sum w = 1$. Цільовою функцією оптимізації було обрано максимізацію

кореляції Пірсона між інтегральним ризиком та бінарним вектором офіційних повітряних тривог на навчальному наборі (60% днів спостережень, обраних стратифіковано за типами сценаріїв solar/osint_heavy/frp_heavy). Додатково враховувалось обмеження: частка хибних спрацювань $\leq 5\%$. Grid search виконувався на сітці з кроком 0,05 для кожної ваги (загалом 66 допустимих комбінацій при $\sum w = 1$), що є обчислювально прийнятним завдяки швидкості агрегації у DuckDB (повний прогін займав менше 2 хвилин). Оптимальна конфігурація ($w_{hits}=0,40$, $w_{FRP}=0,35$, $w_{rad}=0,25$) виявилась близькою до емпіричної ініціалізації, що підтверджує обґрунтованість доменних знань. Для ваг формули геокодування (2.1) застосовувався аналогічний підхід із цільовою функцією - максимізацією точності @5 км на анованому корпусі. Для ваг α , β у формулі інтегрованого ризику (2.7) оптимізація виконувалась окремо для кожного кластера шестикутників (за рівнем урбанізації) шляхом мінімізації MAE прогнозу відносно фактично спостережуваного ризику на наступний день.

Етап 3. Валідація на незалежних даних. Оптимізовані ваги перевірялись на утримуваному наборі (40% днів спостережень), який не використовувався під час оптимізації. Кореляція з тривогами на валідаційному наборі склала $r = 0,76$ (порівняно з $r = 0,78$ на навчальному), що свідчить про відсутність перенавчання та стійкість оптимізованих параметрів. Усі вагові коефіцієнти зберігаються у версіонованих YAML-файлах конфігурації, що дозволяє відстежувати зміни через систему контролю версій та відтворювати будь-яку попередню конфігурацію.

2.5.3. Аналіз чутливості методів до зміни вагових коефіцієнтів

Для оцінки стійкості системи до варіації вагових коефіцієнтів було проведено дослідження чутливості формули інтегрального ризику (2.4).

Дослідження було виконано методом послідовного збурення (one-at-a-time, OAT) та методом Морріса для виявлення нелінійних ефектів.

Базова конфігурація ваг ($w_{hits} = 0,4$, $w_{FRP}=0,35$, $w_{rad} = 0,25$) змінювалась у діапазоні $\pm 20\%$ із кроком 5% для кожного коефіцієнта окремо, при цьому решта

ваг перенормовувалась для збереження умови $\sum w=1$. Для кожної конфігурації обчислювались: (1) кореляція з офіційними тривогами, (2) МАЕ добового ризику відносно базової конфігурації, (3) частка хибних спрацювань. Таблиця 2.8. узагальнює результати аналізу чутливості.

Таблиця 2.8

Вплив варіації вагових коефіцієнтів на метрики інтегрального ризику

(δ — відносне збурення)

Параметр	$\delta = -20\%$	$\delta = -10\%$	$\delta = 0\%$ (базовий)	$\delta = +10\%$	$\delta = +20\%$
w_{hits} : кореляція з тривогами	0,72	0,75	0,78	0,77	0,74
w_{hits} : МАЕ ризику	0,14	0,11	0,08	0,10	0,13
w_{FRP} : кореляція з тривогами	0,74	0,76	0,78	0,77	0,75
w_{FRP} : МАЕ ризику	0,13	0,10	0,08	0,11	0,14
w_{rad} : кореляція з тривогами	0,77	0,78	0,78	0,76	0,73
w_{rad} : МАЕ ризику	0,09	0,08	0,08	0,12	0,16

Виконаний аналіз дозволив виявити наступні закономірності.

1. При збуренні будь-якої ваги на $\pm 10\%$ кореляція з офіційними тривогами залишається у діапазоні 0,75–0,78, а МАЕ зростає не більш ніж на 0,03 відносно базового значення 0,08. Це свідчить про те, що система не потребує ідеально точного підбору ваг – допуск $\pm 10\%$ є прийнятним для практичного використання.

2. Найвища чутливість до w_{rad} . Збільшення ваги радіаційної компоненти на 20% знижує кореляцію до 0,73 та подвоює МАЕ (з 0,08 до 0,16), що пояснюється шумом від природних γ -аномалій (геомагнітних збурень), який посилюється при завищеній вазі. Це підтверджує важливість механізму фільтрації `spatial_uniformity_index`.

3. Стійкість до w_{hits} . Вага OSINT-компоненти є найменш чутливою, оскільки правило кворуму та складена оцінка достовірності (формула 2.3) вже забезпечують внутрішню фільтрацію шуму.

4. Адитивна стійкість. Мультифакторний аналіз методом Морріса (50 траєкторій) підтвердив, що взаємодії між вагами є слабкими (індекс взаємодії $\sigma/\mu^* < 0,3$ для всіх пар), що виправдовує використання адитивної моделі ризику.

Отже, визначення ваг з точністю $\pm 10\%$ від оптимальних значень є достатнім для забезпечення якості системи на рівні кореляції $\geq 0,75$ з офіційними тривогами. Це знижує вимоги до калібрувального набору даних і спрощує розгортання системи.

2.6. Архітектурний шаблон H3-Dual Scale Materialized Views

Побудова надійного програмного забезпечення для обробки гетерогенних поточкових даних вимагає архітектурного каркасу, здатного забезпечити стабільність під навантаженням, прозорість еволюції стану та відтворюваність аналітичних результатів. Для розв'язання цих завдань у даній роботі було розроблено подійно-орієнтовану архітектуру, яка розглядає подію як базову одиницю істини. Кожна зміна стану фіксується атомарно, часово впорядковано та з чітким походженням, що дозволяє відстежувати повну історію перетворень та зменшує конфлікти при паралельному доступі. Така організація узгоджується з рекомендаціями фахової спільноти щодо побудови складних транзакційно-аналітичних рішень [70-72].

Центральну роль в архітектурі відіграє шаблон журналювання подій, який зберігає кожну зміну як незмінний запис у часово впорядкованому журналі. Актуальний стан системи не фіксується безпосередньо, а відтворюється шляхом послідовного відтворення подій або через спеціальні матеріалізовані проєкції – агреговані подання, що обчислюються наперед для прискорення запитів. Це усуває потребу в блокуванні транзакцій при читанні та суттєво підвищує швидкодію аналітичних операцій. Важливою властивістю є детермінованість: оскільки стан

повністю визначається послідовністю подій, у будь-який момент можливо відтворити систему на довільну мить минулого – так звану «подорож у часі», що критично для ретроспективного аудиту та розслідування інцидентів.

Для оброблення потоків подій у режимі, наближеному до реального часу, застосовується Карра-підхід з єдиним журналом як для мікропакетного, так і для повторного оброблення. На відміну від Lambda-архітектури, яка реалізує паралельні шари пакетної та потокової обробки з дублюванням бізнес-логіки, Карра узгоджує логіку обробки через використання єдиного логу подій [73]. Це знижує вартість розробки, спрощує тестування та усуває ризик розбіжностей між швидким та пакетним шарами. Незмінна природа журналу спрощує реалізацію функцій, які можуть повторно обробляти ті самі дані, що важливо при відновленні після збоїв. Фізичне розділення шляхів запису та читання дає можливість незалежно масштабувати навантаження, а підтримка мікропакетів забезпечує еластичність при нерівномірному надходженні подій без ускладнень, властивих брокерам повідомлень. Просторову уніфікацію реалізовано через ієрархічну шестикутну сітку НЗ, яка забезпечує майже рівновіддалених сусідів та стабільні операції агрегування. Шестикутна геометрія природно апроксимує кругові зони покриття та знижує спотворення, характерні для квадратних решіток [74], [99]. Ієрархічна структура дозволяє просто переходити між рівнями роздільності: агрегати на вищому рівні обчислюються шляхом зведення дрібніших комірок, що спрощує побудову багатомасштабних аналітичних систем. Використання сталої сітки спрощує індексацію у сховищі: добові та тижневі подання будуються з однаковими ключами без складних операцій просторового з'єднання, що критично для задач, де близькість і покриття мають пріоритет над точністю геометрії окремих об'єктів. Для забезпечення високопродуктивного виконання локальних аналітичних OLAP-запитів було обрано аналітичне ядро DuckDB [75]. Векторизоване виконання запитів, стиснення колонок та підтримка сучасних форматів зберігання забезпечують високу швидкодію на одному вузлі без необхідності розгортання розподіленого кластера (рис. 2.7).

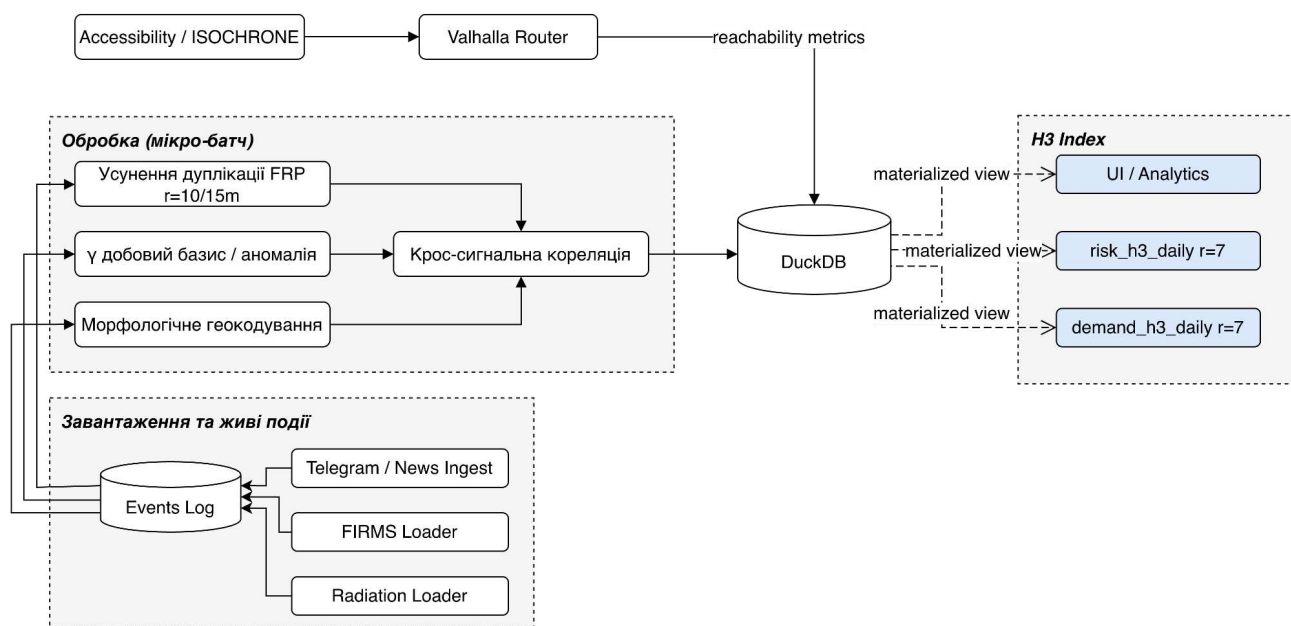


Рис. 2.7. Логічна схема взаємозв'язків елементів програмного забезпечення
(UML діаграма компонентів)

Для мережевих обчислень відстані, часу досяжності та побудови ізохрон застосовується автономний рушій Valhalla, що працює на графах OpenStreetMap та підтримує сценарії без підключення до мережі [104]. Ізохронні розрахунки виконуються на попередньо підготовлених картографічних тайлах та надають часові метрики доступності, які безпосередньо інтегруються в моделі покриття та візуалізуються на тих самих НЗ-тайлах. Необхідність попередньої підготовки компенсується надійністю та швидкістю оцінки доступності у критичних ситуаціях, що особливо значуще для автономної роботи [76].

Узгодження описаних компонентів формує послідовну архітектуру: незмінний журнал подій забезпечує точне походження даних; CQRS ізолює аналітичні запити від операцій запису та прискорює формування проєкцій; НЗ-набір ключів усуває проблему неоднорідності геометрій; DuckDB надає швидкі агрегати без додаткових серверів; Valhalla дозволяє оцінювати доступність у режимі автономної роботи. Гібридний підхід, що поєднує мікропакетне оброблення для основного конвеєра, НЗ-індексацію для просторових операцій,

DuckDB для локальної аналітики та Valhalla для оцінки доступності, забезпечує оптимальний компроміс між функціональністю та складністю для систем моніторингу екологічних загроз. У підсумку зменшується архітектурний профіль ризику: відмова будь-якого з модулів не призводить до втрати історії, оскільки події збережені у незмінному журналі, а відновлення аналітичних вітрин виконується повторним відтворенням. Вимоги до інфраструктури мінімізовані, а самодостатність вузла зростає, що зменшує залежність від зовнішніх сервісів під час кризових обставин.

У даній дисертаційній роботі розроблено архітектурний шаблон H3-DSMV (Dual Scale Materialized Views) для аналізу гетерогенних потоків даних, що забезпечує фіксацію подій на дрібнішій сітці з автоматичною побудовою матеріалізованих подань на вищій сітці.

Запропонований архітектурний шаблон забезпечує фіксацію подій на дрібномасштабній шестикутній сітці з автоматичним формуванням матеріалізованих подань на більш грубій сітці. В основі такого підходу лежить журнал подій на рівні H3 r10, що виступає єдиним джерелом істини та забезпечує високу просторову точність близько 15 м² на шестикутник, а також матеріалізовані подання рівня H3 r7, у яких зберігаються добові агрегати з нижчою деталізацією приблизно 5 км² на шестикутник. Архітектура запропонованого шаблону передбачає автоматичну синхронізацію, за якої подання оновлюються при надходженні кожного мікропакета даних без ручного втручання, і не має зовнішніх залежностей, оскільки весь цикл зберігання та оброблення реалізовано в межах одного процесу (рис. 2.8).

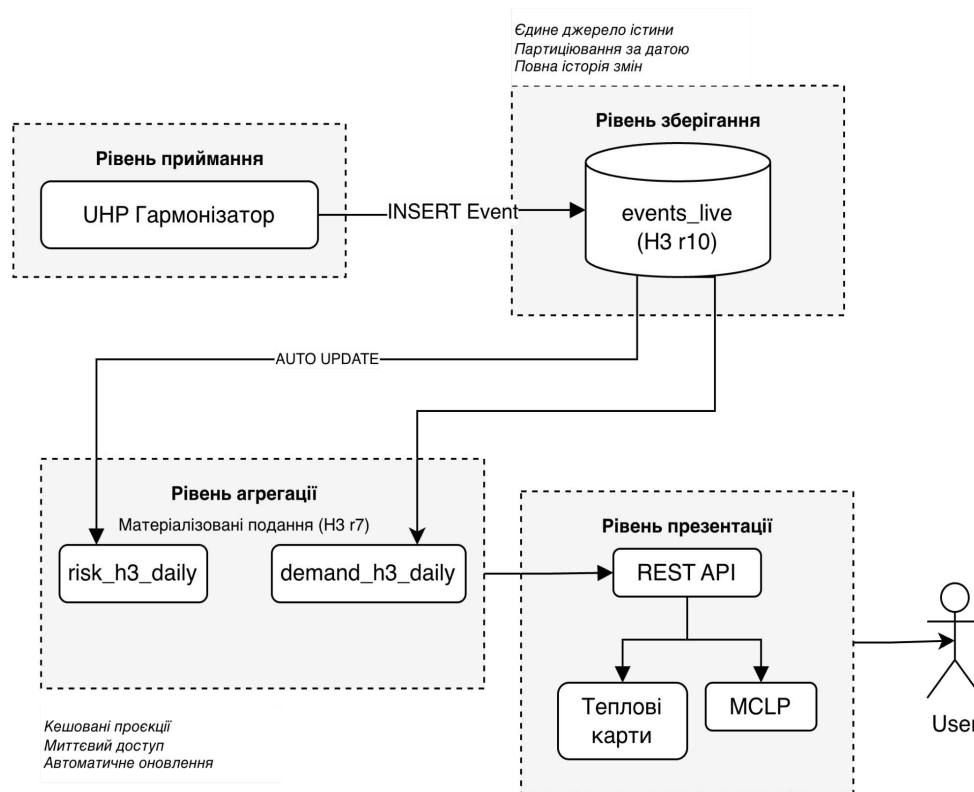


Рис. 2.8. Архітектурний шаблон H3-DSMV

Розглянемо основні кроки використання розробленого архітектурного шаблону.

Крок 1. Приймання події. Подія надходить від гармонізатора UHP у форматі Event з H3-індексом рівня 10.

Крок 2. Запис у журнал. Подія вставляється у таблицю events_live з партиціюванням за датою.

Крок 3. Тригерне оновлення. При вставці автоматично запускається оновлення матеріалізованих подань *risk_h3_daily* та *demand_h3_daily* на рівні H3 r7.

Крок 4. Агрегація $r10 \rightarrow r7$. Виконується ієрархічне згортання: *h3_cell_to_parent(h3_r10, 7)* з підсумовуванням метрик.

Крок 5. Доступ до результатів. Клієнти отримують миттєвий доступ до актуальних агрегатів через SQL-запити.

Крок 6. *Переобчислення історії*. При потребі журнал r10 дозволяє миттєво переобчислити будь-який історичний період.

Шаблон H3-DSMV ґрунтується на низці концептуальних засад, що забезпечують коректність, узгодженість оброблення геопросторових потокових даних. Ключовою з них є принцип єдиного джерела істини (Single Source of Truth), відповідно до якого журнал подій на рівні просторової роздільності H3 r10 виступає авторитетним та незмінним джерелом даних для всіх похідних обчислень. Матеріалізовані подання в цій архітектурі розглядаються як кешовані проєкції журналу подій, що формуються автоматично для прискорення аналітичних запитів, а не як незалежні сховища, що зменшує ризик логічної неузгодженості даних.

Важливою основою шаблону є принцип ієрархічної узгодженості, який забезпечується використанням шестикутної просторової індексації H3. Ієрархічна структура H3 гарантує ізоморфне відображення між різними рівнями роздільності, зокрема кожен шестикутник рівня r7 утворюється як агрегат семи шестикутників рівня r10. Це дозволяє виконувати просторові агрегації без втрати коректності та спотворення геометричних властивостей даних, що є критично важливим для аналітики потоків у просторі та часі.

Ще одним фундаментальним принципом є мінімальна інфраструктура, який передбачає відмову від використання зовнішніх брокерів повідомлень, розподілених кешів або складних сервісів синхронізації. Такий підхід зменшує поверхню потенційних відмов, спрощує розгортання та супровід системи, а також підвищує відтворюваність результатів, оскільки весь цикл оброблення даних реалізується в межах одного процесу та узгодженого обчислювального середовища. Завдяки своїм властивостям шаблон H3-DSMV може бути успішно застосований у широкому спектрі предметних галузей, де необхідна геопросторова агрегація потокових даних у близькому до реального часу режимі. У транспортній аналітиці запропонований архітектурний шаблон може використовуватися для моніторингу трафіку та оптимізації маршрутів

громадського транспорту; в урбаністиці – для аналізу пішохідних потоків та планування міської інфраструктури. У телекомунікаціях запропонований підхід є корисним для оптимізації покриття мереж та аналізу навантаження, в охороні здоров'я – для епідеміологічного моніторингу та просторового планування медичних закладів. В енергетиці запропонований шаблон може підтримувати задачі моніторингу споживання та оптимізації розподільних мереж, а в екологічних дослідженнях використовуватися для моніторингу забруднення довкілля та аналізу просторових закономірностей біорізноманіття.

2.7. Висновки до розділу 2

У цьому розділі систематизовано архітектурні та алгоритмічні засоби опрацювання гетерогенних потоків даних для задач ситуаційної обізнаності. Концептуальна основа спирається на подійно-орієнтовану архітектуру з розділенням читання та запису, журналюванням змін, що забезпечує масштабованість, відтворюваність та формальний аудит усіх трансформацій. Таке поєднання дозволяє відокремити контур приймання та оновлень від контуру аналітичних запитів, скорочуючи блокування та роблячи можливими регенерацію вітрин через відтворення подій та контроль якості на кожному кроці.

Просторово-часова уніфікація здійснюється на сітці H3, яка надає ієрархічні шестикутні ключі для агрегування ознак і стабільних операцій сусідства; легковагова аналітика реалізується у вбудованій СУБД DuckDB, а мережеві обчислення доступності підтримуються автономним рушієм Valhalla, що забезпечує автономність і експлуатацію в режимі автономної роботи у стресових сценаріях. Потоки OSINT та сенсорні дані інтегруються через модулі розпізнавання подій та геокодування. Багатомовні трансформери застосовуються для класифікації з багатьма мітками текстів з урахуванням дисбалансу класів та слабо контрольованих схем маркування; додатково використовується NER та аргументація для виділення сутностей та зв'язування їх з подіями. Геокодування реалізовано морфологічно-чутливим каскадом із лематизацією, нечітким

фонетичним зіставленням та семантичним ранжуванням, що дозволяє перетворювати неструктуровані повідомлення у стандартизовані події з координатами та часовими мітками, придатні до об'єднання з фізичними сенсорами. Фільтраційно-оцінювальний шар об'єднує надійність джерела, ймовірність події з NLP та сенсорне підтвердження у складений бал, що підпорядковує випуск подій правилам кворуму та усунення дублювання. Для вагових параметрів моделі розроблено трирівневу методику калібрування. Ваги геокодування визначено емпіричною пріоритизацією етапів зіставлення та відкалібровано на анотованому корпусі з 1500 речень за критерієм точності на порозі 5 км. Ваги інтегрального ризику та коефіцієнти поточного та прогнозного ризику оптимізовано процесом, що складається із трьох етапів: емпірична ініціалізація на основі доменного знання, сіткова оптимізація на навчальній вибірці (60%) з максимізацією кореляції Пірсона з офіційними повітряними тривогами та обмеженням частки хибних позитивів $\leq 5\%$, і валідація на відкладеній вибірці (40%), яка підтвердила відсутність перенавчання ($r = 0,76$ проти $r = 0,78$ на навчальній множині). Аналіз чутливості методами ОАТ та Морріса показав помірну стійкість системи до варіацій ваг: при збуреннях $\pm 10\%$ кореляція залишається в діапазоні 0,75–0,78, MAE зростає не більш ніж на 0,03 відносно базового значення 0,08, а слабкі взаємодії між параметрами ($\sigma/\mu^* < 0,3$) підтверджують коректність адитивної моделі. Добові показники ризику та попиту агрегуються на рівні НЗ $r=7$ й забезпечують узгоджений вхід для просторового оптимізаційного аналізу: оцінювання покриття наявною інфраструктурою та розв'язання задачі максимального покриття (MCLP) для вибору місць інтервенцій за обмеженої кількості ресурсів. У дисертаційній роботі використано поєднання подійної архітектури, багатомовних NLP-моделей, морфологічно адаптованого геокодування та гібридного підтвердження з фізичними сенсорами, що у комплексі формує прозору та відтворювану оцінку ризиків у квазіреальному часі.

Переваги запропонованих рішень охоплюють кілька рівнів. По-перше, забезпечується обчислювальна економність та автономність: відсутність брокерів

та розподілених кластерів зменшує експлуатаційні витрати, а локальні обчислення в DuckDB спрощують відтворюваність та аудит. По-друге, підвищується якість подій завдяки каскадному фільтру: комбінація текстових та сенсорних свідчень знижує частку хибних спрацювань та стабілізує добові агрегати. По-третє, уніфікація на НЗ та наявність прозорих матеріалізованих вітрин скорочують шлях від джерела до карти ризику і, зрештою, до рішень щодо розміщення інфраструктури захисту. У підсумку отримуються придатні до використання ГІС-шари ризику та попиту та обґрунтовані рекомендації для реагувальних заходів. Запропонована комбінація LSTM-прогнозування з ймовірнісною просторовою інтеграцією перетворює систему з реактивного моніторингу на інструмент підтримки превентивних рішень. Запропонований підхід забезпечує наочну, формально визначену основу для подальших управлінських дій – від раннього попередження до планування нових укриттів.

Таким чином, запропоновані рішення забезпечують інтеграцію різномірних потоків, узгоджений просторово-часовий аналіз та можливість переходу від карти ризику до оптимізаційних рішень щодо інфраструктури захисту, зберігаючи прозорість розрахунків і керованість порогів у межах сценаріїв автономної роботи.

Запропонований архітектурний шаблон НЗ-DSMV призначений для обробки геопросторових потокових даних та поєднує принцип єдиного джерела істини, ієрархічну узгодженість просторових агрегацій та мінімальну інфраструктурну складність. Фіксація подій на дрібнішій сітці НЗ з автоматичним формуванням матеріалізованих подань на вищому рівні роздільності забезпечує коректну та швидку аналітику без дублювання даних та втрати точності. Основними перевагами розробленого архітектурного шаблону є висока просторово-часова точність, детермінована агрегація, автоматична синхронізація подань, зменшення накладних витрат на інфраструктуру та підвищена надійність і масштабованість системи в задачах реального часу.

РОЗДІЛ 3. АРХІТЕКТУРА ТА РЕАЛІЗАЦІЯ МОДУЛЬНОЇ ПРОГРАМНОЇ СИСТЕМИ ОБРОБЛЕННЯ ПОТОКОВИХ ДАНИХ НА ОСНОВІ МІКРОПАКЕТНОГО КОНВЕЄРА

3.1. Узагальнена модульна архітектура мікропакетного конвеєра з НЗ-агрегацією та матеріалізованими поданнями для оброблення гетерогенних поточкових даних

Методи, розглянуті у попередньому розділі, формують теоретичну основу, проте їхнє практичне застосування вимагає відповідної інженерної реалізації. Гетерогенність джерел інформації супроводжується суттєвими відмінностями у форматах даних та часових шкалах, що ускладнюється додатковими обмеженнями щодо швидкодії та відмовостійкості. Це зумовлює необхідність побудови архітектури, здатної трансформувати теоретичні концепції у функціональний програмний конвеєр. Розроблена система охоплює повний цикл оброблення даних: від мікропакетного збору інформації з різнорідних джерел, через процедури уніфікації та агрегування, до формування матеріалізованих подань, які забезпечують оперативне оцінювання ризиків та попиту на ресурси.

Архітектурні рішення обґрунтовуються не лише функціональними можливостями окремих компонентів, а й характером поточкових даних, що обробляються у реальному часі.

Успішність реалізації системи значною мірою визначається обґрунтованим вибором технологічних компонентів, кожен з яких має задовольняти специфічним вимогам до швидкодії, гнучкості та відкритості. Вбудована аналітична СУБД DuckDB була обрана через її оптимізацію для аналітичних обчислень у колонковому форматі, підтримку прямого читання та запису у формати Parquet, CSV та JSON, а також можливість використання у звичайних застосунках без необхідності розгортання окремих серверних компонентів [109]. Така гнучкість дозволяє зберігати як журнальні таблиці з повною історією подій, так і матеріалізовані подання з агрегованими метриками, виконуючи SQL-запити з

мінімальною затримкою безпосередньо у процесі роботи програмного забезпечення. Розроблене програмне забезпечення надає можливість паралельної обробки мікропакетів з різних джерел без блокування основного потоку виконання під час очікування на мережеві виклики, що дозволяє досягти оброблення потоків даних у реальному часі без застосування зовнішніх брокерів повідомлень, що спрощує розгортання системи та знижує операційні витрати на підтримку додаткової інфраструктури.

Для аналізу текстових даних OSINT було обрано трансформери XLM-RoBERTa та mDeBERTa, що демонструють високі результати на крос-мовних завданнях [81]. Здатність цих моделей успішно працювати з кількома мовами без необхідності окремого навчання для кожної мови робить їх оптимальним вибором для обробки багатомовних потоків кризової інформації. Морфологічний аналіз забезпечується бібліотеками Stanza та Py morphology2, які надають точні інструменти для лематизації та аналізу граматичних форм текстів східнослов'янської групи мов [103]. Для реалізації нечіткої відповідності географічних назв використовується бібліотека RapidFuzz, що підтримує метрики подібності Jaro-Winkler та Levenshtein з оптимізованою реалізацією для високої швидкодії [11].

Використання в програмному забезпеченні OSMnx дозволяє завантаження та аналіз дорожніх мереж з бази OpenStreetMap через API [112], ліцензія MIT забезпечує придатність для наукових проєктів без юридичних обмежень. У комбінації з сервісом маршрутизації Valhalla [107] це створює повноцінну систему для отримання оптимальних маршрутів та аналізу доступності об'єктів, що є критичним для геопросторової оптимізації розміщення укриттів цивільного захисту. Візуалізація результатів та користувацький інтерфейс реалізовані за допомогою фреймворку Streamlit, який у поєднанні з бібліотекою PyDeck дозволяє створювати інтерактивні інструменти для відображення НЗ-шестикутників та результатів агрегування у тривимірному представленні.

Така можливість візуалізації є важливою для оперативної комунікації результатів аналізу з нетехнічними фахівцями, що приймають рішення щодо розподілу ресурсів та планування заходів реагування.

Загальну архітектуру розробленого програмного забезпечення для моніторингу екологічних загроз представлено на рис. 3.1. Система побудована за багаторівневим принципом, де кожен рівень відповідає за окрему фазу оброблення даних від збирання до подання результатів кінцевому користувачу.

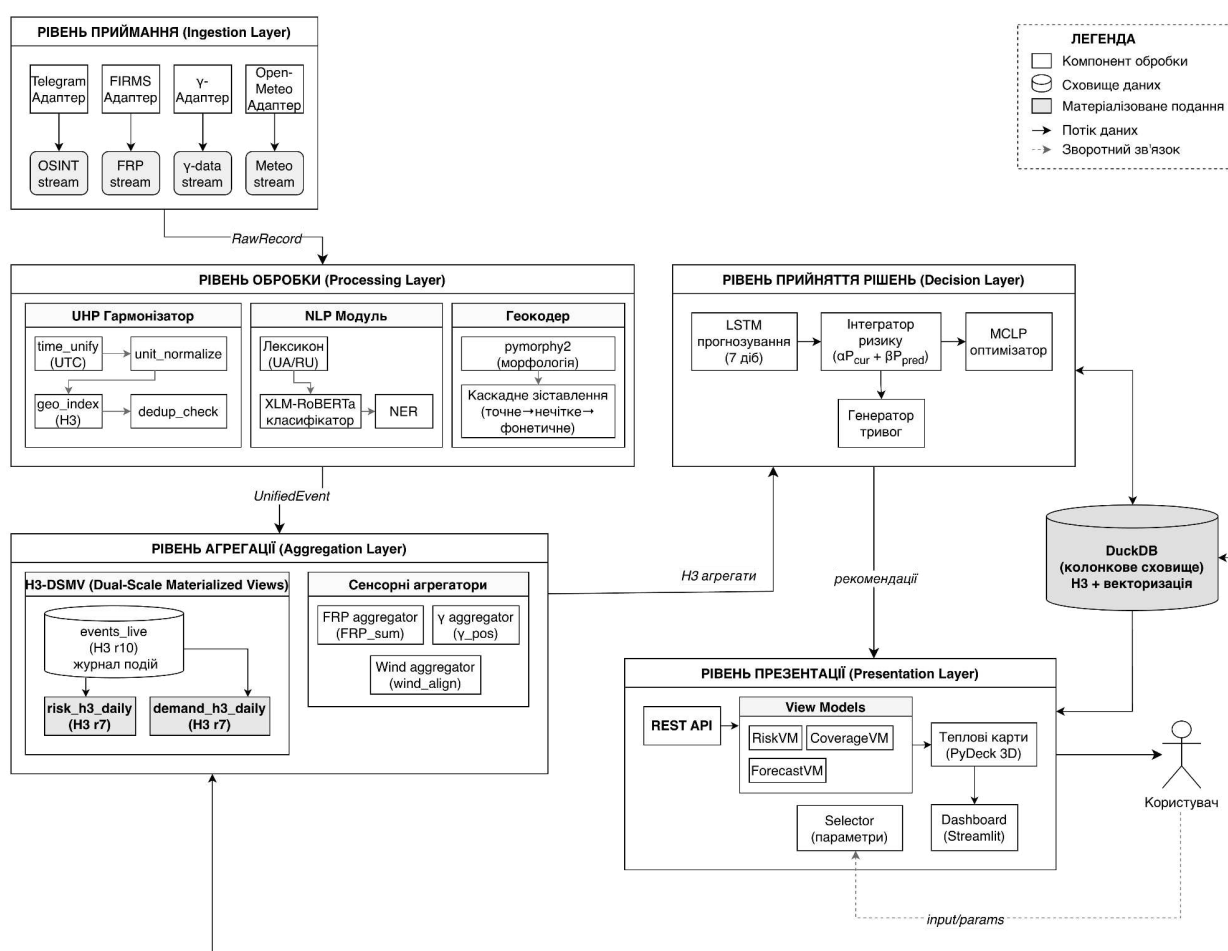


Рис. 3.1. Загальна архітектура розробленого програмного забезпечення для моніторингу екологічних загроз

Рівень приймання (Ingestion Layer) забезпечує збирання даних з гетерогенних зовнішніх джерел через уніфікований адаптерний шар. Кожне джерело інкапсульоване у відповідному адаптері, що реалізує інтерфейс

SourceAdapter з методом poll() → List[RawRecord]: Telegram-адаптер отримує OSINT-повідомлення з визначених каналів у форматі HTML/JSON; FIRMS-адаптер завантажує CSV-файли термічних детекцій із супутників MODIS та VIIRS; γ-адаптер збирає дані радіаційного моніторингу з мереж SaveEcoBot та EURDEP у JSON-форматі; Open-Meteo-адаптер отримує метеорологічні показники через REST API. Асинхронна реалізація адаптерів на основі asyncіо дозволяє паралельно опитувати всі джерела без взаємного блокування.

Рівень обробки (Processing Layer) складається з трьох модулів, що працюють конвеєрно. UHP-гармонізатор (Unified Harmonization Pipeline) виконує детерміновані перетворення: приведення часових міток до UTC (time_unify), нормалізацію одиниць вимірювання (unit_normalize), просторове індексування на сітці НЗ (geo_index) та перевірку дублікатів (dedup_check). Середній час затримки гармонізації не перевищує 0,5 с.

Після гармонізації потік розгалужується: числові сенсорні дані (FRP, γ-випромінювання, метеопоказники) спрямовуються безпосередньо до рівня агрегації, тоді як текстові OSINT-повідомлення проходять через NLP-модуль (лексична фільтрація кризових термінів на базі двомовного лексикону UA/RU, класифікація подій трансформером XLM-RoBERTa, розпізнавання іменованих сутностей) та геокодер (морфологічна нормалізація rymorphy2, каскадне зіставлення з реєстром географічних назв: точне - нечітке - фонетичне).

Рівень агрегації (Aggregation Layer) реалізує архітектурний шаблон НЗ-DSMV, описаний у підрозділі 2.5. Журнал подій events_live на рівні НЗ r10 є єдиним джерелом істини; при надходженні кожного мікропакету автоматично оновлюються матеріалізовані подання risk_h3_daily та demand_h3_daily на рівні НЗ r7. Паралельно сенсорні агрегатори обчислюють добові показники: FRP_sum (сумарна потужність термічних аномалій), γ_pos (нормалізований z-індекс γ-випромінювання) та wind_align (метеокорекція за напрямом вітру) відповідно до процедур доменної агрегації, описаних у підрозділі 2.4.

Рівень прийняття рішень (Decision Layer) інтегрує агрегати з попереднього рівня для формування оцінок ризику та рекомендацій. LSTM-модуль формує короткострокові прогнози (1–7 діб) для кожного гексагону. Інтегратор ризику поєднує поточну оцінку P_{cur} та прогнозну P_{pred} за формулою $\alpha P_{cur} + \beta P_{pred}$, де коефіцієнти адаптуються відповідно до точності попередніх прогнозів для кожного регіону. MCLP-оптимізатор розв'язує задачу максимального покриття для визначення оптимальних місць розташування нових укриттів. Генератор тривог формує сповіщення при перевищенні порогових значень ризику; кореляція генерованих тривог з офіційними повітряними тривогами становить $r = 0,78$ ($p < 0,001$). Рівень презентації (Presentation Layer) забезпечує доступ до результатів через REST API, набір моделей подання (RiskVM, CoverageVM, ForecastVM) та інтерактивну візуалізацію: тривимірні теплові карти НЗ-гексів (PyDeck) та аналітичну панель (Streamlit). Користувач може змінювати параметри аналізу (часовий діапазон, пороги, вагові коефіцієнти) через інтерактивний інтерфейс.

Наскрізним компонентом, що обслуговує рівні агрегації, прийняття рішень та презентації, є вбудована аналітична СУБД DuckDB. Колонкове зберігання, векторизоване виконання запитів та підтримка форматів Parquet забезпечують час відгуку на аналітичні запити менше 3 мс (p95) без необхідності розгортання окремого серверного процесу. Вся система функціонує на одному обчислювальному вузлі, що зменшує інфраструктурну складність та забезпечує автономність роботи у стресових сценаріях.

3.1.1. Архітектура програмного конвеєра для оброблення мікропакетних потоків даних

Традиційні системи оброблення даних зазвичай орієнтуються на великі періодичні партії інформації, що відповідає усталеним практикам пакетного оброблення. Проте для завдань моніторингу у реальному часі, де затримка між виникненням події та її аналізом має вимірюватися секундами, такий підхід виявляється непридатним. Досвід роботи з різнорідними джерелами – від

відкритих даних соціальних мереж до супутникових систем виявлення пожеж, від мережі радіаційного моніторингу до метеорологічних сервісів демонструє доцільність використання мікропакетів як основної одиниці оброблення. Мікропакети являють собою невеликі порції даних, що надходять з високою частотою та обробляються незалежно одна від одної, формуючи безперервний потік інформації. Кожне джерело даних характеризується власною специфікою форматів та затримок передачі. Телеграм-канали надають інформацію у вигляді HTML та JSON з мінімальною затримкою, система FIRMS оперує CSV-файлами з тригодинною затримкою, мережі радіаційного моніторингу SaveEcoBot та EURDEP передають дані у власних форматах з різними часовими поясами, а метеорологічні API повертають JSON-відповіді з годинною дискретністю.

Для уніфікації взаємодії з цими різнорідними джерелами було розроблено адаптерний шар, побудований навколо інтерфейсу SourceAdapter. Кожен адаптер втілює метод *poll()*, який стандартизує процес отримання даних, перетворюючи специфічні відповіді джерел на уніфіковані «сирі» записи типу RawRecord. Ці записи містять мінімальний набір обов'язкових полів – часову мітку *timestamp*, географічні координати *lat* та *lon*, вимірюване значення *value* з відповідною одиницею виміру *unit*, ідентифікатор джерела та додаткові метадані у форматі JSON для збереження специфічної інформації.

Після збору даних розпочинається етап гомогенізації, критичний для можливості подальшого зіставлення інформації з різних джерел. Неоднорідність часових форматів – ISO 8601 у телеграм-дописах, Unix-мітки у системі NASA FIRMS, локальний час на станціях радіаційного моніторингу – робить безпосереднє порівняння часових міток неможливим. Функція *time_unify()* виконує приведення всіх часових міток до всесвітнього координованого часу UTC та здійснює фільтрацію за робочим часовим вікном, відкидаючи записи, датовані раніше 20 лютого 2022 року, що дозволяє сфокусуватися на періоді активних подій. Нормалізація одиниць виміру усуває ризик помилок, спричинених змішуванням різних вимірювальних систем: значення Fire Radiative Power

приводяться до мегават, швидкість вітру – до метрів за секунду, показники гамма-випромінювання – до мікрозівертів на годину. Окрім цього, географічні координати у системі WGS-84 перетворюються у геопросторові ідентифікатори системи НЗ, що створює основу для просторової агрегації на різних рівнях деталізації [108]. Мікропакетний конвеєр реалізує послідовність детермінованих перетворень, кожне з яких забезпечує специфічну функцію узгодження даних. Етап отримання даних побудовано на асинхронних викликах з використанням бібліотеки Python *asyncio*, що дозволяє паралельно опитувати множинні джерела без блокування основного потоку оброблення. Така архітектура забезпечує раціональне використання обчислювальних ресурсів при роботі з джерелами, що характеризуються різною швидкістю відгуку та частотою оновлення даних.

Нормалізація часових міток виконується з урахуванням складності обробки часових поясів та переходів на літній час. Використання бібліотек *pendulum* та *datetime* забезпечує коректну інтерпретацію часових даних від географічно розподілених джерел, що є критичним для систем, які інтегрують інформацію з понад 50 станцій моніторингу у різних часових поясах. Перетворення одиниць виміру реалізовано з використанням *numpy* для забезпечення числової стабільності та коректної обробки граничних значень, включно з механізмами виявлення та фільтрації фізично неможливих показників, що могли з'явитися внаслідок збоїв сенсорів або помилок передачі даних.

Геопросторове узгодження виконує конвертацію географічних координат у індекси НЗ на двох рівнях роздільності: рівень десять ($\sim 1 \text{ км}^2$) використовується для точкових подій, що вимагають високої просторової точності, тоді як рівень сім ($\sim 50 \text{ км}^2$) застосовується для добових агрегатів, де просторова деталізація може бути знижена на користь швидкодії обчислень. Бібліотека *h3-py* забезпечує швидко просторову індексацію з можливістю подальшої агрегації за геометрично узгодженими комірками. Ієрархічна структура індексів НЗ створює передумови для багаторівневої просторової аналітики без необхідності повторного перерахунку координат при зміні масштабу аналізу.

Формування події полягає у створенні стандартизованого об'єкта Event, що містить поля *metric*, *timestamp_utc*, *lat*, *lon*, *value*, *unit*, *source* та *extra_json* для додаткових атрибутів. Ці події зберігаються у вбудованій СУБД DuckDB, архітектура якої оптимізована для аналітичних операцій завдяки колонковому зберіганню, що забезпечує високу швидкість виконання агрегаційних запитів при збереженні компактності представлення даних.

Маршрутизація подій здійснюється на основі типу метрики: текстові повідомлення направляються до NLP-підмодуля для семантичного аналізу, тоді як числові показники FRP, γ -випромінювання та параметрів вітру передаються безпосередньо до сенсорних агрегаторів. Координація паралельної обробки різнотипних даних реалізована з використанням Python coroutines, що дозволяє раціонально керувати множинними потоками без взаємного блокування.

Реалізація конвеєра вимагає ретельного врахування специфічних характеристик джерел даних. Затримка у системі Fire Information for Resource Management System може досягати трьох годин [79], що зумовлює необхідність використання вікна дедуплікації тривалістю п'ятнадцять хвилин при об'єднанні сигналів з різних орбітальних проходів супутників. Асинхронний цикл забезпечує одночасний збір та оброблення даних з усіх джерел, підтримуючи високу пропускну здатність системи без залежності від зовнішніх брокерів повідомлень, що спрощує розгортання та обслуговування інфраструктури.

3.1.2. Реалізація програмних модулів

Модульна архітектура системи базується на принципі розмежування відповідальності компонентів, що є фундаментальним для забезпечення обслуговуваності та можливості масштабування. Кожен компонент функціонує у власному обмеженому контексті, що мінімізує зв'язність між модулями та полегшує подальше розширення функціональності системи. Адаптери джерел інкапсулюють знання про специфічні API та формати даних, тоді як агрегатори та аналітичні модулі оперують виключно стандартизованими об'єктами Event. Такий

підхід гарантує, що інтеграція нового джерела даних, наприклад, мережі сенсорів забруднення повітря – вимагатиме лише розробки відповідного адаптера без необхідності модифікації існуючих компонентів системи.

Модуль *osint_tg_news* обробляє HTML та JSON повідомлення з телеграм-каналів та RSS-стрічок, виконуючи їхнє перетворення у об'єкти *RawRecord* з полями *timestamp*, *text*, *source* та *metadata*. Фільтрація за ключовими словами відбувається безпосередньо на рівні адаптера, що знижує навантаження на подальші етапи обробки через раннє відсіювання нерелевантних повідомлень, не пов'язаних з кризовими подіями.

Модуль *firms_frp* інтегрує дані з CSV-файлів системи FIRMS, що містять поля *acq_date*, *acq_time*, *lat*, *lon* та *frp* для кожного виявленого теплового аномалії. Об'єднання даних з супутників MODIS та VIIRS ускладнюється можливістю перекриття зон покриття та різницею у часі спостережень, що може досягати трьох годин. Це зумовлює необхідність спеціальної логіки дедуплікації, яка забезпечує вибір найсвіжіших спостережень при наявності множинних детекцій однієї події, запобігаючи завищенню енергетичних показників пожеж.

Модуль *radiation* опрацьовує API SaveEcoBot і EURDEP, формуючи записи з часовою міткою, значенням гамма-випромінювання та ідентифікатором станції моніторингу. Перетворення локального часу станцій у UTC є критичним завданням для узгодження даних з понад п'ятдесяти різних джерел у різних часових поясах. Система автоматично виявляє та коректує переходи на літній час, що є необхідним для забезпечення коректного часового вирівнювання даних при подальшій аналітиці часових рядів.

Модуль *weather_wind* забезпечує погодинні дані про параметри вітру через обробку JSON-відповідей метеорологічних API. Нормалізація включає не лише приведення швидкості вітру до метрів за секунду та кутового напрямку до градусів від північного напрямку, а й перетворення між різними системами координат, що використовуються різними метеорологічними службами.

Data Event Harmonizer є центральним компонентом конвеєра, що приймає список об'єктів RawRecord і виконує комплексну уніфікацію часових міток, одиниць виміру та координат, формуючи список стандартизованих об'єктів Event. Цей критичний компонент забезпечує узгодженість даних через застосування детермінованих правил перетворення, при цьому здійснюючи повне журналювання всіх операцій для можливості аудиту та відтворення результатів.

Модуль NLP з геокодуванням обробляє об'єкти Event з текстовим вмістом, застосовуючи послідовність аналітичних процедур. Трансформери забезпечують семантичне розуміння природної мови для класифікації типу події, морфологічний аналіз виявляє географічні назви у тексті з урахуванням відмінювання слів, а географічний довідник визначає координати згаданих локацій. Вихідні записи ImpactsRecord включають тип події, геолокацію та додаткові характеристики, при цьому для кожного виявленого атрибута зберігається рівень впевненості, що дозволяє оцінити надійність отриманої інформації.

Модуль *Sensor Transformations* виконує агрегацію числових показників через обчислення п'ятнадцятихвилинних сум FRP для пожежних детекцій, годинних медіан γ -випромінювання для усунення випадкових змін фону та добових профілів вітру для аналізу напрямків поширення забруднювачів. Застосування окремих правил усунення дублювання та згладжування для кожного типу сенсора враховує специфічні характеристики шуму та варіативності відповідних вимірювань, забезпечуючи оптимальне співвідношення між точністю та стабільністю результатів.

Важливою складовою архітектури є політика, що забезпечує коректність обробки при можливому повторному надходженні даних. Кожен запис RawRecord має унікальний ідентифікатор для текстових подій це криптографічний хеш змісту повідомлення, для сенсорних вимірювань – комбінація дати, часу та географічних координат. Коли один і той самий запис надходить повторно внаслідок повторного запиту до API або мережевих збоїв, система розпізнає дублікат та відкидає його

без додаткового оброблення. Такий підхід забезпечує узгодженість даних без необхідності використання зовнішніх брокерів повідомлень, спрощуючи архітектуру та знижуючи операційні витрати. Система чітко визначених контрактів між модулями створює основу для модульної архітектури, що спрощує тестування окремих компонентів через можливість їхньої ізольованої перевірки. Незалежна еволюція модулів при збереженні сумісності інтерфейсів дозволяє оновлювати реалізацію окремих компонентів без впливу на функціонування системи у цілому. Така архітектура формує надійну основу для масштабування системи через додавання нових джерел даних та аналітичних компонентів відповідно до еволюції вимог моніторингу екологічних загроз.

3.1.3. Реалізація агрегатора НЗ та матеріалізованого подання

Всі нормалізовані події зберігаються у журнальній таблиці, яка функціонує як єдине джерело істини для всієї системи. Проте безпосереднє використання «сирих» подій для аналітичних запитів виявляється повільним: операції над мільйонами рядків призводять до неприйнятних затримок відповідей, що несумісно з вимогами моніторингу у реальному часі. Відповіддю на цей виклик стало впровадження НЗ-агрегатора, який виконує підсумовування подій за просторовими та часовими критеріями. Шестикутна система НЗ забезпечує рівномірне покриття поверхні Землі без спотворень, характерних для прямокутних сіток, та підтримує гнучке масштабування між рівнями деталізації. Система використовує два рівні індексації: r10 (~1 км²) для коротких часових вікон, де необхідна висока просторова точність, та r7 (~50 км²) для добових шарів.

Агрегування здійснюється у два послідовні етапи, кожен з яких оптимізований для специфічних вимог відповідного рівня деталізації. На першому етапі відфільтровані події згортаються у п'ятнадцятихвилинні вікна на рівні r10 (рис. 3.2). Для показника FRP це означає, що з усіх спостережень у межах одного шестикутника r10 протягом 15 хвилин зберігається максимальне значення теплової потужності, що відповідає піковій інтенсивності пожежі у цьому

просторово-часовому вікні. Для γ -радіації обчислюється медіана значень за годинний інтервал, що дозволяє усунути випадкові викиди та зосередитися на стійких тенденціях радіаційного фону. Параметри вітру усереднюються як за напрямком, так і за модулем швидкості, формуючи репрезентативний вектор для відповідного часового проміжку.

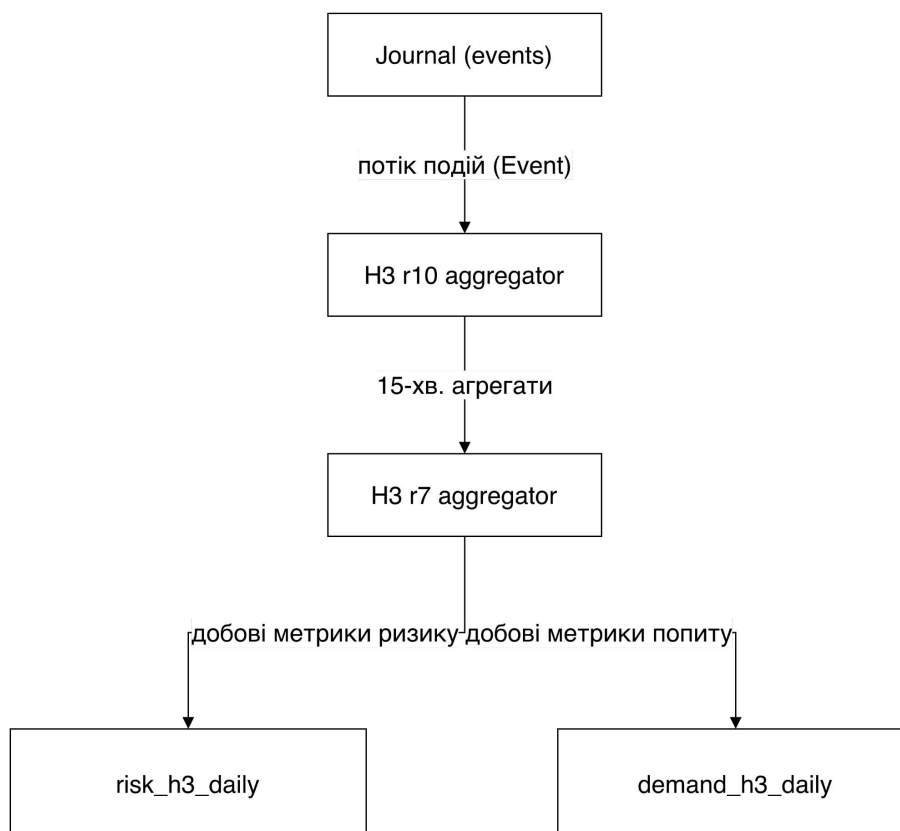


Рис. 3.2. Згортання потоків нормалізованих подій у H3-агрегаторі (DFD – діаграма потоків даних).

На другому етапі ці тимчасові блоки узагальнюються на рівні r7 за добовими інтервалами. Результуюча структура даних містить набір інтегральних показників для кожного шестикутника r7 та кожної доби: *n_hits* відображає кількість зареєстрованих ударів у потоках OSINT, *frp_sum* агрегує сумарну теплову потужність пожеж після усунення дублювання супутникових спостережень, *γ_{pos}* представляє усереднений позитивний z-індекс

гамма-випромінювання, що характеризує відхилення від нормального фону, а *wind_align* кодує ваговий коефіцієнт для моделювання поширення забруднювачів у напрямку домінуючого вітру. Ці метрики зберігаються у матеріалізованих поданнях *risk_h3_daily* та *demand_h3_daily*, де перший шар використовується для оцінки сумарного ризику екологічних загроз, тоді як другий призначений для оцінки попиту на ресурси як добутку ризику на чисельність населення відповідного шестикутника. Така архітектура дозволяє виконувати складні просторово-часові запити за лічені секунди. Коли виникає необхідність змінити параметри ризикової формули або вагові коефіцієнти, наприклад, посилити вплив фактора вітру при моделюванні поширення радіоактивних забруднювачів достатньо переобчислити матеріалізоване подання, не використовуючи історичний журнал подій. Агрегування у геопросторі НЗ забезпечує природну сумісність із візуальними засобами, які можуть безпосередньо відображати шестикутні плитки різних рівнів деталізації без додаткових перетворень координат.

3.1.4. Реалізація архітектурного шаблону НЗ-DSMV на основі двошарового матеріалізованого підходу

Під час проєктування системи виявилось, що традиційні схеми оброблення поточкових даних не забезпечують необхідного поєднання швидкодії та гнучкості для геопросторових подій з динамічними вимогами до аналітики. Довгі ланцюги ETL-процесів та численні проміжні сервіси призводять до кумулятивного збільшення затримки між моментом виникнення події та отриманням підсумкової метрики, що робить систему непридатною для оперативного моніторингу.

У розробленому архітектурному шаблоні НЗ-DSMV дані спочатку потрапляють у журнал на дрібнішому рівні шестикутної сітки НЗ (r10), де забезпечується висока просторова точність для окремих подій, після чого негайно агрегуються у подання на вищому рівні (r7) для створення оглядових шарів з меншою деталізацією. Термін «матеріалізовані» відображає той факт, що агрегати створюються та оновлюються у вбудованій СУБД автоматично при надходженні

нових даних, зберігаючи повну історію змін та забезпечуючи постійний доступ до актуальних шарів без необхідності повторного обчислення.

У шаблоні H3-DSMV всі операції відбуваються у межах одного процесу та єдиного сховища даних, що радикально спрощує архітектуру порівняно з розподіленими системами. Журнал на рівні r10 функціонує як єдине джерело істини, приймаючи події від модуля приймання після їхнього перетворення на уніфікований формат Event. Агрегатор рівня r7 безперервно підсумовує ці події у часових вікнах, формуючи добові показники ризику та попиту для кожного шестикутника r7. Паралельно формуються матеріалізовані подання *risk_h3_daily* та *demand_h3_daily*, які залишаються доступними для миттєвих запитів від клієнтського інтерфейсу без затримок на обчислення. Така архітектура усуває потребу в окремих брокерах повідомлень, розподілених кешах чи сторонніх потокових платформах – усі компоненти взаємодіють безпосередньо у межах вбудованої СУБД через стандартизовані SQL-запити.

Запропонований шаблон H3-DSMV має низку практичних переваг, що виходять за межі простого спрощення архітектури. Низька затримка оброблення даних досягається завдяки тому, що агрегування здійснюється безпосередньо в СУБД без передачі даних через мережеві з'єднання, що дозволяє отримувати результати протягом секунд після появи події у журналі.

Складність інфраструктури зведена до мінімуму відсутністю окремих брокерів повідомлень, черг та багатовузлових кластерів, що знижує вимоги до обслуговування та зменшує поверхню потенційних відмов. Геопросторова узгодженість забезпечується використанням H3 на обох рівнях агрегації, що створює ізоморфну структуру даних для інтеграції з алгоритмами оптимізації та візуалізацією без додаткових перетворень координат. Гнучкість аналізу досягається завдяки матеріалізованим поданням, які дозволяють користувачам змінювати ваги у формулі ризику або вибирати інші часові інтервали без необхідності перебудови всього каскаду обчислень. Практична оцінка впровадження шаблону H3-DSMV продемонструвала його переваги у реальних

умовах експлуатації. Замість погодинних або тригодинних циклів оновлення добові подання тепер оновлюються після кожного мікропакета з частотою кілька разів на хвилину, що зменшує затримку надходження інформації та робить карти ризику придатними для оперативного прийняття рішень.

Перенесення всього контуру від приймання даних до візуалізації результатів у єдину систему DuckDB спростило процедуру розгортання та знизило витрати на обслуговування інфраструктури. Використання двох рівнів НЗ – r10 як деталізованого журналу та r7 для добових шарів забезпечує оптимальний баланс між точністю та швидкодією. Можливість миттєвого відтворення історичних даних безпосередньо з журналу r10 без необхідності повного ETL-перезапуску підвищує надійність системи для проведення досліджень та аудиту рішень.

Винесення вагових коефіцієнтів у YAML-конфігурацію з їхнім застосуванням під час виконання запитів дозволяє експериментувати з різними сценаріями оцінювання ризику без простоїв системи для рекомпіляції чи перезапуску сервісів.

3.2. Програмна реалізація модуля оброблення мовних компонент

Функціональна цілісність системи критично залежить від здатності своєчасно та точно інтерпретувати текстові повідомлення з відкритих джерел. Дані OSINT, що надходять з новинних порталів та інших публічних джерел, характеризуються високим рівнем неструктурованості, багатомовністю та присутністю неформальних мовних конструкцій, що ускладнює їхнє автоматизоване оброблення. Завданням мовного модуля є «витягування» сигналів про обстріли, вибухи та інші кризові інциденти з цього інформаційного потоку, їхня прив'язка до географічних координат та оцінювання достовірності отриманої інформації. Поєднання лексичних правил для попередньої фільтрації, глибоких нейромереж для семантичного аналізу та алгоритмів постоброблення для усунення помилок дозволяє досягти високої точності класифікації при збереженні прийнятної швидкодії для потокової системи.

3.2.1. Реалізація лексикону кризових термінів

Використання глибоких моделей машинного навчання для оброблення всього потоку вхідних повідомлень без попередньої фільтрації виявляється надто ресурсомістким для системи реального часу, враховуючи, що значна частина повідомлень не стосується кризових подій.

У даній дисертаційній роботі було розроблено лексикон кризових термінів, що охоплює лексику української та ворожої мови для різних типів інцидентів. Для генерації всіх можливих граматичних форм кожного терміну – відмінків, чисел, родів, часових форм використовується морфологічний аналізатор *rumorphy2*. Наприклад, для базової форми «обстріл» автоматично генеруються слововформи «обстрілу», «обстрілом», «обстрілам», «обстрілами», тоді як для ворожого слова «взрыв» створюються відповідні форми «взрыва», «взрывом», «взрывов» та інші. Сам словник організовано у форматі *YAML* для зручності редагування та може динамічно доповнюватися під час експлуатації системи на основі аналізу помилок класифікації. Формування списку термінів базується на емпіричному аналізі реальних новинних повідомлень про кризові події, що дозволило виявити не лише прямі позначення типів інцидентів («вибух», «ракетний удар», «пожежа»), а й характерні для кризової комунікації синоніми та евфемізми («прильот», «влучання», «попадання»). Для кожного терміна у словнику вказується категорія події – наприклад, *strike* для ударів, *fire* для пожеж, *casualty* для повідомлень про постраждалих, а також ідентифікатор мови, що дозволяє коректно застосовувати морфологічні правила. Така попередня фільтрація за ключовими словами значно скорочує обсяг текстів, що передаються до обчислювально затратного етапу класифікації нейромережею, при цьому ризик втрати важливих подій мінімізується завдяки всебічному покриттю можливих слововформ. Розширення словника триває паралельно з навчанням класифікатора. Виявлення нових термінів відбувається на основі аналізу помилок: якщо модель часто пропускає повідомлення через відсутність ключового слова, воно додається до словника. Подібна інкрементальна процедура відповідає підходу активного навчання.

Таблиця 3.1

Фрагмент лексикону кризових термінів

Категорія	Основне слово	Приклади морфологічних форм	Синоніми
Удар (strike)	вибух	вибуху, вибухом, вибухи	прильот, приліт, взрив
Артилерія (shelling)	обстріл	обстрілу, обстрілом, обстріли	бомбардування
Пожежа (fire)	пожежа	пожежі, пожежою, пожежі	займання
Сирена (alarm)	сирена	сирени, сиреною, сирен	тривога
Постраждалі (casualty)	постраждалий	постраждалих, постраждалим	поранений

3.2.2. Розробка та навчання багатокласового класифікатора з фокальною втратою

Після фільтрації повідомлень за лексиконом кризових термінів відібрані тексти потрапляють до етапу точної категоризації типу події. Задача формулюється як багатокласова класифікація з наступними категоріями: *strike* (обстріли або вибухи), *alert* (повітряні тривоги), *fire* (пожежі), *damage* (руйнування без постраждалих), *casualty* (загиблі або поранені) та *other* (інші події).

Для вирішення цієї задачі використано трансформерну архітектуру XLM-RoBERTa, яка забезпечує багатомовну обробку текстів без необхідності використання окремих мовних моделей.

Якість моделі критично залежить від якості навчальних даних, що зумовило необхідність ретельного формування корпусу. Для даного дослідження було зібрано понад 50 000 маркованих повідомлень з періоду від серпня 2022 до травня 2025 року, охоплюючи весь спектр кризових подій з різною інтенсивністю.

Для зменшення негативного впливу класового дисбалансу було застосовано стратегію додаткового збирання рідкісних класів, зокрема, повідомлень про пожежі та постраждалих, які природно зустрічаються рідше за повідомлення про удари. До корпусу включено повідомлення як українською так і ворожою мовою, що підвищує стійкість моделі до варіативності мовних формулювань у кризовій комунікації. Навчання моделі реалізовано на фреймворку PyTorch з

використанням оптимізатора AdamW, що забезпечує стабільне оновлення ваг з адаптивною швидкістю навчання. Стандартна функція крос-ентропійної втрати була замінена на фокальну втрату, яка дозволяє приділяти більше уваги важким прикладам шляхом зменшення впливу легко класифікованих зразків [83].

Фокальна втрата модифікує стандартну крос-ентропію через додавання фокусуєчого параметра, що посилює градієнти для помилково класифікованих прикладів:

$$FL(pt) = -\alpha_t(1 - p_t)^\gamma \log(p_t), \quad (3.1)$$

де p_t – передбачена ймовірність правильного класу, α_t – ваговий множник для балансування класів, γ – параметр фокусування (у нашій реалізації $\gamma=2.0$), що контролює швидкість зменшення ваги для впевнено класифікованих прикладів. Параметр α налаштовується окремо для кожного класу на основі його частоти у навчальному наборі, компенсуючи природний дисбаланс розподілу подій.

Структуру навчального процесу демонструє узагальнений приклад коду лістингу 3.1, який ілюструє збереження контрольних точок моделі, ведення журналу метрик навчання, валідацію на окремій вибірці.

Лістинг 3.1. Фрагмент коду, що демонструє навчальний процес

```
import torch
from transformers import AutoTokenizer,
AutoModelForSequenceClassification
from torch.utils.data import DataLoader
from torch.nn import functional as F
MODEL_NAME = "xlm-roberta-base"
NUM_CLASSES = 6
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
model = AutoModelForSequenceClassification.from_pretrained(MODEL_NAME,
num_labels=NUM_CLASSES)
optimizer = torch.optim.AdamW(model.parameters(), lr=2e-5)
def focal_loss(logits, targets, gamma=2.0, alpha=None):
    """Виконує фокус-втрату для класів targets."""
    probs = F.softmax(logits, dim=-1)
    pt = probs.gather(1, targets.unsqueeze(1)).squeeze()
    log_pt = torch.log(pt + 1e-12)
    # якщо alpha задано, множимо на коефіцієнт класу
    if alpha is not None:
```

```

        at = alpha.gather(0, targets)
        loss = -at * ((1 - pt) ** gamma) * log_pt
    else:
        loss = -((1 - pt) ** gamma) * log_pt
    return loss.mean()
for epoch in range(NUM_EPOCHS):
    for batch in DataLoader(train_dataset, batch_size=BATCH_SIZE,
shuffle=True):
        inputs = tokenizer(batch['text'], padding=True,
truncation=True, return_tensors='pt')
        labels = torch.tensor(batch['label'])
        outputs = model(**inputs)
        loss = focal_loss(outputs.logits, labels, gamma=2.0,
alpha=ALPHA_TENSOR)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

```

Таблиця 3.2 відображає порівняння показників точності, повноти та F1-міри для кожного класу класифікатора до і після оптимізації, демонструючи загальне покращення якості класифікації після налаштування моделі.

Таблиця 3.2

Порівняння метрик класифікатора до та після оптимізації

Клас	Точність (до)	Повнота (до)	F1 (до)	Точність (після)	Повнота (після)	F1 (після)
strike	0,82	0,79	0,80	0,86	0,84	0,85
alert	0,88	0,85	0,86	0,91	0,88	0,89
fire	0,68	0,64	0,66	0,76	0,73	0,74
damage	0,73	0,69	0,71	0,80	0,76	0,78
casualty	0,62	0,58	0,60	0,71	0,68	0,69
other	0,85	0,82	0,83	0,88	0,85	0,86
Середнє	0,76	0,73	0,74	0,82	0,79	0,80

Після завершення навчання модель зберігається у форматі TorchScript для швидкого завантаження у виробничому середовищі без залежності від Python-інтерпретатора. На етапі валідації виконується узгодження порогів

класифікації для кожного класу окремо: замість фіксованого значення 0,5 застосовується оптимальний поріг, що максимізує F1-міру для відповідного класу на перевіірочній вибірці. Це дозволяє адаптувати чутливість системи до специфічних вимог різних типів подій, наприклад, для класу *casualty* критичнішою є повнота виявлення навіть ціною деякого зростання хибних спрацьовувань, тоді як для класу *other* пріоритетом є точність. Високі значення F1-міри (0,91) та загальної точності (0,92) на перевіірочному наборі свідчать про адекватність обраної архітектури (табл. 3.3). Аналіз метрик окремих класів показує, що категорії *strike* та *alert* досягають F1-міри понад 0,94 завдяки великій кількості навчальних прикладів та чітким лінгвістичним маркерам, тоді як клас *casualty* демонструє нижчий показник 0,87 через обмежену кількість прикладів та більшу варіативність формулювань повідомлень про постраждалих.

Таблиця 3.3

Гіперпараметри та результати класифікації

Параметр	Значення	Коментар
Кількість епох	3	достатньо для збіжності, без перенавчання
Розмір пакету	64	баланс між швидкодією та пам'яттю
Швидкість навчання (lr)	2e-5	оптимізатор AdamW
γ (focal loss)	2,0	посилює «важкі» приклади [83]
α (focal loss)	за класами	компенсує дисбаланс
Macro F1	0,91	середній показник для 6 класів
Точність	0,92	точність на перевіірочному наборі

3.2.3. Реалізація черги обчислювальних запитів та кешування результатів

Виведення результатів класифікації у системі реального часу вимагає ретельного балансування між пропускнуою здатністю та затримкою. Для оптимізації цього балансу було розроблено систему черги виведення, що накопичує повідомлення у мікропакети та обробляє їх пакетно на GPU.

Адаптер *osint_tg_news* після нормалізації та фільтрації додає повідомлення у *asyncio.Queue*, звідки окремий процесор читає їх пакетами – від 32 до 128 повідомлень залежно від поточного навантаження. Розмір пакета динамічно

адаптується: під час періодів високої активності пакети формуються швидше для максимального використання паралельних можливостей GPU, тоді як у спокійні періоди система переходить на обробку меншими порціями для зменшення затримки окремих запитів. Модель класифікатора завантажується один раз при ініціалізації системи та залишається у пам'яті протягом всього часу роботи, що усуває накладні витрати на повторне завантаження ваг. Використання *mixed-precision inference* з типом даних *float16* дозволяє прискорити обчислення на GPU без суттєвої втрати точності. Для уникнення дублювання обчислень застосовується LRU-кеш: коли однаковий текст з'являється повторно – наприклад, внаслідок репостів одного й того ж повідомлення у різних каналах – результат класифікації негайно повертається з кешу без виклику моделі. Розмір кешу, типово 5000 записів, добирається експериментально як компроміс між економією пам'яті та частотою влучень; при переповненні найстаріші елементи автоматично видаляються згідно з політикою LRU. Архітектура системи ілюструє послідовність оброблення від надходження «сирих» повідомлень через фільтрацію за лексиконом до класифікації з використанням кешу. Кешування інтегрується після первинної фільтрації: лише при пропуску в кеші модель виконує повне виведення, що дозволяє досягти не лише прискорення обробки, а й суттєвого зменшення навантаження на обчислювальні ресурси GPU.

На рис. 3.3. зображена схема запропонованого потокового оброблення текстових повідомлень соціальних мереж та новинних ресурсів.

Для моніторингу продуктивності системи ведеться статистика затримок на різних етапах обробки: час токенізації тексту, час виведення моделі, час постоброблення результатів. Ця інформація використовується для виявлення «вузьких» місць та оптимізації конфігурації системи. Такий мікропакетний підхід з динамічним кешуванням дозволив досягти середньої затримки менше 200 мілісекунд для обробки одного повідомлення при пікових навантаженнях до 50 повідомлень на секунду, скоротивши середній час виведення до менш ніж однієї секунди від моменту надходження повідомлення до отримання класифікації.

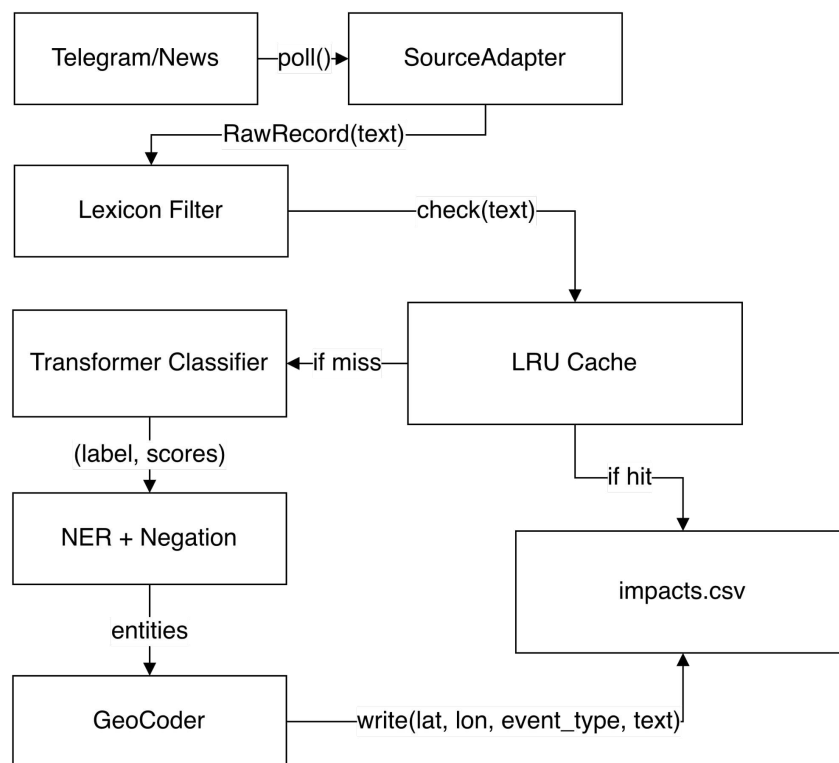


Рис. 3.3. Схема запропонованого потокового оброблення текстових повідомлень соціальних мереж та новинних ресурсів (DFD діаграма потоків даних).

3.2.4. Реалізація модулів розпізнавання іменованих сутностей та виявлення заперечень у текстах природною мовою

Виявлення іменованих сутностей (Named Entity Recognition, NER) та коректне оброблення заперечувальних конструкцій є необхідними умовами для формування достовірного архіву подій impacts.csv.

Для виявлення іменованих сутностей було використано бібліотеку Stanza, яка забезпечує підтримку мов східнослов'янської групи з точним морфологічним аналізом [103]. На виході Stanza надає список токенів з їхніми лемами, частинами мови та БІО-тегами для ідентифікованих сутностей, що дозволяє структурувати неформальні тексти кризових повідомлень.

Фокус системи зосереджено на типах сутностей, що безпосередньо впливають на точність геокодування: LOC (географічні локації), FAC (будівлі та споруди) та HAZARD (небезпечні об'єкти чи зони). Наприклад, речення "Вибухи

чули у Києві біля вокзалу" буде автоматично розмічене як LOC: Києві, FAC: вокзалу, що дозволяє системі ідентифікувати як загальну локацію події (місто), так та конкретне місце у його межах (вокзал). NER-модель ідентифікує не лише прості однослівні назви населених пунктів, а й складні багатослівні конструкції, адміністративні одиниці (області, райони), природні географічні об'єкти (річки, гори) та інфраструктурні споруди (мости, заводи, торговельні центри).

Важливою особливістю мов східнослов'янської групи є наявність розвиненої системи відмінків, що призводить до множинних словоформ одного топоніма. Після виконання NER система трансформує виявлені назви до початкової форми (леми) за допомогою морфологічного аналізу.

Наприклад, "Києві", "Києвом", "у Києві" усі приводяться до базової форми "Київ" перед передачею до модуля геокодування. Ця нормалізація є критичною для забезпечення успішного пошуку координат у географічному довіднику, де записи зберігаються у називному відмінку.

Оброблення заперечень становить окремий виклик для автоматичної системи класифікації. Повідомлення про кризові події часто містять заперечувальні конструкції різних типів: пряме заперечення («ситуація без втрат», «жертв немає», «обстрілу не було»), числові вирази («загиблих 0», «0 постраждалих») або негативні характеристики стану («безпечний район», «все спокійно»). Некоректна інтерпретація таких фраз може призвести до формування хибних записів про постраждалих або інциденти, що не відбулися.

Для вирішення цієї проблеми було розроблено правило-базований модуль оброблення заперечень. Алгоритм виконує послідовний обхід токенизованого тексту, виявляючи ключові маркери заперечення ("не", "без", "немає", "ні") та аналізуючи їхній синтаксичний контекст. Коли частка "не" розташована безпосередньо перед кризовим терміном у конструкції типу "не було обстрілу", відповідний клас події позначається як заперечений, і запис не потрапляє до архіву подій (табл. 3.4). Для числових виразів система аналізує конфігурацію:

присутність нуля перед або після слова, що позначає постраждалих ("загиблих 0", "0 постраждалих"), інтерпретується як явне заперечення наявності жертв.

Таблиця 3.4

Основні правила ідентифікації заперечень

Тип конструкції	Приклад	Інтерпретація
Заперечення перед іменником	«не було обстрілу», «немає пожеж»	Скидає клас strike або fire
Заперечення з числом	«0 загиблих», «без постраждалих»	Клас casualty позначається як відсутній
Негативний прикметник	«безпечний район», «все спокійно»	Не створює події, відкидає повідомлення
Подвійне заперечення	«не було не було»	Ігнорується, як шум

Правила оброблення заперечень, незважаючи на їхню відносну простоту порівняно з можливими нейромережними альтернативами, значно зменшують кількість хибних позитивів у системі. Впровадження цього модуля зменшило частоту помилкової реєстрації неіснуючих подій приблизно на чверть.

3.2.4. Інтеграція мовного модуля з геокодуювальною підсистемою

Завершальним етапом оброблення у мовному модулі є передача результатів класифікації та розпізнавання сутностей до геокодуювальної підсистеми, детально описаної у підрозділі 3.3. Після виконання класифікації типу події та ідентифікації географічних назв через NER система формує структуру ImpactsRecord, яка інкапсулює всю зібрану інформацію про подію: часову мітку події, присвоєну категорію (strike, alert, fire тощо), поле координат (на цьому етапі встановлене у None), текст оригінального повідомлення для можливості ручної верифікації, список розпізнаних топонімів з їхніми лемами та контекстну інформацію про кожен топонім, зокрема його тип (місто, область, споруда) та позицію у тексті.

Сформований запис ImpactsRecord передається до асинхронної черги геокодування, де він очікує на обробку модулем визначення координат. Геокодуювальна підсистема, отримавши запис, виконує послідовність операцій для кожного виявленого топоніма: нормалізацію написання, пошук у географічному

довіднику, усунення неоднозначності при множинних кандидатах та присвоєння рівня впевненості результату. У відповідь геокодувальна система повертає кортеж (*lat*, *lon*, *conf*, *admin_path*), де *lat* та *lon* – географічні координати у системі WGS-84, *conf* – числовий показник впевненості у діапазоні від 0 до 1, *admin_path* – повний адміністративний шлях типу "Україна/Київська область/Київ" для контекстної верифікації. Отримані координати записуються у публічний архів *impacts.csv*, що формує відкритий набір даних про кризові події для використання дослідниками та аналітиками. Паралельно у таблиці *events* встановлюється прапорець *hit_reported*, що вказує на завершення оброблення події та її експорт до архіву, запобігаючи повторним записам при можливому перезапуску або відновленні системи після збоїв. Усі поля у таблицях системи стандартизовано у нотації *snake_case* для забезпечення узгодженості іменування: *n_hits*, *frp_sum*, *y_pos*, *wind_align* тощо, що спрощує програмний доступ та знижує ймовірність помилок через неузгоджені конвенції найменування. Така модульна організація з чітким розмежуванням відповідальності дозволяє незалежно вдосконалювати мовний модуль та геокодувальну підсистему без впливу на функціонування суміжних компонентів. Наприклад, під час експериментів можна протестувати альтернативну модель NER з кращою точністю для рідкісних топонімів або розширити лексикон кризових термінів новими синонімами, не вносячи жодних змін у логіку геокодування. Аналогічно, покращення алгоритмів усунення неоднозначності у геокодері не вимагають перенавчання класифікатора подій. Така архітектурна незалежність забезпечує гнучкість розвитку системи та спрощує тестування окремих компонентів у ізоляції.

3.2.5. Розроблення уніфікованого конвеєра гармонізації гетерогенних потокових даних UHP

Хоча мовний модуль зосереджений на аналізі текстових повідомлень, успішне функціонування всієї системи критично залежить від того, як «сирі» дані з різнорідних джерел уніфікуються до спільного формату перед надходженням до

класифікатора та інших аналітичних компонентів. Цю функцію виконує уніфікований конвеєр гармонізації UHP – спеціально розроблена система, яка забезпечує модульну, швидку та детерміністичну обробку гетерогенних потоків даних. Розглянемо основні кроки запропонованого алгоритму гармонізації.

Крок 1. Приймання від адаптерів. Кожен адаптер джерела реалізує *poll()* → *List[RawRecord]*, повертаючи «сирі» записи з полями: *timestamp*, *lat*, *lon*, *value*, *unit*, *source*, *metadata*.

Крок 2. Уніфікація часу. Функція *time_unify(ts, tz_hint)* конвертує часову мітку до UTC з валідацією діапазону (записи до 20.02.2022 відкидаються як неактуальні).

Крок 3. Нормалізація одиниць. FRP – МВт, γ -радіація – $\mu\text{Sv/h}$, швидкість вітру – м/с. На цьому кроці усуваються можливі помилки від змішування систем вимірювання.

Крок 4. Геопросторова прив'язка. Координати WGS-84 конвертуються у індекси НЗ рівня 10 для точкових подій.

Крок 5. Усунення дублювання. Записи ідентифікуються за унікальними ключами: хеш тексту для OSINT, комбінація *timestamp* - *lat* - *lon* для сенсорів. Дублікати автоматично відкидаються.

Крок 6. Контроль якості. Генеруються структуровані записи про пропуски, помилки форматування та аномалії у значеннях.

Крок 7. Маршрутизація. Уніфіковані події Event маршрутизуються: текстові – мовний модуль, числові – сенсорні агрегатори.

Архітектура конвеєра базується на чіткому розділенні відповідальності між адаптацією даних від специфічних джерел та їхньою уніфікацією до загальносистемного формату (рис. 3.4).

Конвеєр організовано у вигляді двох функціональних шарів з різними зонами відповідальності. Перший шар складається з адаптерів джерел – спеціалізованих модулів, що інкапсулюють знання лише про своє конкретне джерело: специфіку його API, формат відповідей та часовий пояс.

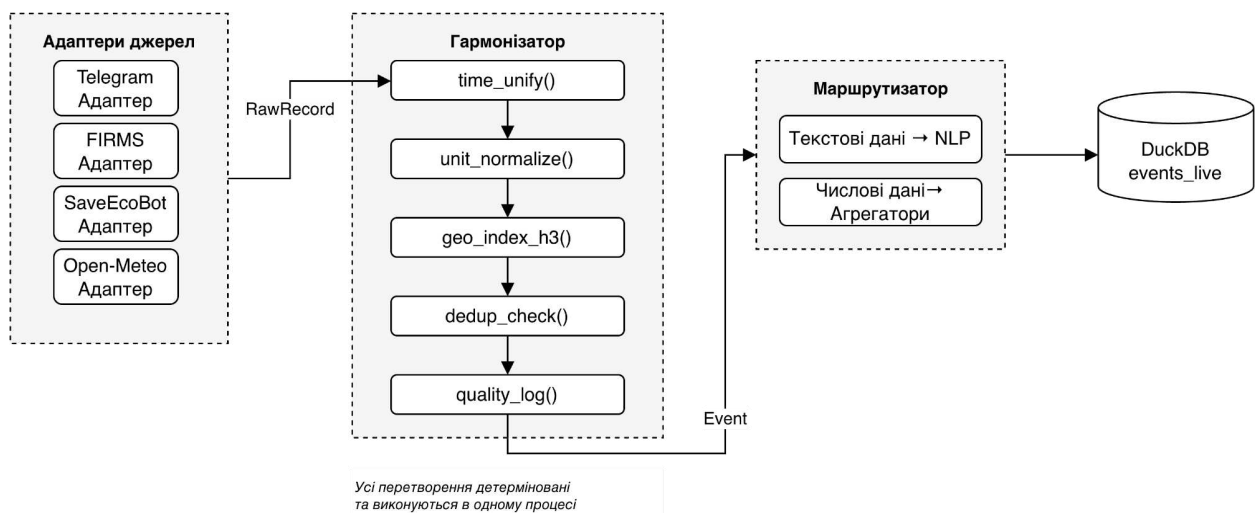


Рис. 3.4. Архітектура уніфікованого конвеєра гармонізації UHP

Кожен адаптер реалізує простий уніфікований контракт $poll() \rightarrow List[RawRecord]$, що повертає список «сирих» записів з мінімальним набором стандартизованих полів: *timestamp* (часова мітка у довільному форматі джерела), *lat* та *lon* (географічні координати), *value* (вимірюване значення), *unit* (одиниця виміру у нотації джерела), *source* (ідентифікатор джерела) та *metadata* (додаткові атрибути у форматі JSON). Така архітектурна ізоляція дозволяє додавати нові джерела даних, наприклад, датчики якості повітря або сейсмічні станції – шляхом створення нового адаптера без необхідності змін у центральній логіці системи. Другий шар архітектури представлено гармонізатором – центральним компонентом, який приймає списки об'єктів *RawRecord* від множинних адаптерів та перетворює їх на стандартизовані об'єкти *Event* з уніфікованими часовими мітками, одиницями виміру та геопросторовими координатами. Процес гармонізації включає чотири критичні операції. По-перше, уніфікація часу виконується функцією $time_unify(ts, tz_hint)$, яка приймає часову мітку у довільному форматі (ISO 8601, Unix timestamp, локальний формат) разом з підказкою про часовий пояс та переводить її до всесвітнього координованого часу (UTC), одночасно виконуючи валідацію діапазону – записи, датовані раніше

лютого 2022 року, відкидаються як неактуальні для аналізу поточної кризи. По-друге, нормалізація одиниць виміру приводить всі значення до узгоджених базових одиниць: теплову потужність пожеж (FRP) – до мегават, γ -радіацію – до мікрозівертів на годину, швидкість вітру – до метрів за секунду, що повністю усуває можливість помилок від змішування різних систем вимірювання. По-третє, геопросторова прив'язка конвертує координати з системи WGS-84 у індекси шестикутної системи НЗ на рівні роздільності десять для точкових подій, забезпечуючи сумісність з подальшими етапами просторової агрегації. По-четверте, усунення дублювання та контроль якості ідентифікують записи за унікальними ключами – криптографічним хешем тексту для OSINT-повідомлень або комбінацією *timestamp-lat-lon* для сенсорних вимірювань, що дозволяє автоматично відкидати дублікати, а пропуски у послідовностях даних сигналізуються системі контролю якості для подальшого розслідування. Результатом процесу гармонізації є однорідний перелік об'єктів Event, які маршрутизуються до відповідних підсистем оброблення: текстові події направляються до мовного модуля для класифікації та NER, тоді як числові показники FRP, γ -випромінювання та параметрів вітру передаються безпосередньо до сенсорних агрегаторів для обчислення просторово-часових метрик. На відміну від стандартного ETL, де логіка перетворення даних була розсіяна по множинних компонентах, уся логіка уніфікації тепер сконцентрована в одному визначеному місці, що спрощує тестування окремих перетворень, відлагодження проблем з форматами даних та повторне використання логіки гармонізації в інших проєктах.

3.3. Програмна реалізація геопросторової підсистеми з використанням геокодування зі збереженням коректності для різних словоформ

Точна локалізація подій є фундаментальною вимогою для побудови просторових та ризикових моделей, що забезпечують основу для прийняття обґрунтованих рішень. Однак східнослов'янські топоніми представляють комплекс складних випадків для автоматизованого геокодування: багата система

відмінкових форм, існування історичних та альтернативних назв, множинні варіанти написання у кирилиці та латиниці, численні скорочення та аббревіатури. Стандартні сервіси геокодування, оптимізовані переважно для англomовних назв, часто повертають неточні або неповні координати при роботі зі східнослов'янськими топонімами. Для усунення цих обмежень було розроблено спеціалізовану геопросторову підсистему, що враховує морфологічні особливості та використовує каскадний механізм пошуку з поступовим зниженням строгості критеріїв відповідності.

3.3.1. Реалізація нормалізації топонімів з урахуванням морфологічної варіативності

Початковим етапом геокодування є перетворення варіативних написань топонімів до базової канонічної форми, яка може бути зіставлена з записами у географічному довіднику. Топоніми підлягають відмінюванню за численними граматичними категоріями: «Києві» перетворюється на «Київ» у називному відмінку, «області» нормалізується до однини «область», «Харківського» приводиться до базової форми «Харківський». Система генерує лему для кожного слова у топонімі, одночасно виділяючи семантичні підказки про адміністративний рівень об'єкта: маркери *city* для міст, *region* для областей, *district* для районів, що з'являються у скороченнях типу «м.», «обл.», «р-н». Окрім морфологічного аналізу, система використовує таблиці історичних та альтернативних назв для відображення застарілих або перейменованих топонімів на їхні сучасні еквіваленти. Наприклад, «Житомирщина» розпізнається як неформальна назва Житомирської області, «Дніпропетровськ» відображається на сучасну назву «Дніпро» тощо (див. приклади табл. 3.5). Ці таблиці відповідностей зберігаються у окремих Parquet-файлах, що дозволяє їх періодично оновлювати відповідно до офіційних реєстрів без необхідності модифікації програмного коду системи. Для кожної нормалізованої назви зберігається метайнформація про адміністративний

рівень географічного об'єкта, що надалі використовується для ранжування кандидатів при наявності множинних відповідностей.

Таблиця 3.5

Приклади нормалізації топонімів

Вхідний рядок	Лема	Підказка	Коментар
«Києві», «м. Київ»	Київ	city	Визначення міста з відмінка та скорочення
«Харківського району»	Харківський район	district	Морфологічний аналіз відмінка; виділення адміністративного рівня
«Дніпропетровськ»	Дніпро	city	Історична назва, актуалізована після декомунізації
«Львівської обл.»	Львівська область	region	Розшифровка скорочення «обл.»; лема
«m. Odesa», «Odesa»	Одеса	city	Транслітерація з латиниці; лема кирилицею

Система також розпізнає числові та алфавітні коди географічних об'єктів, наприклад, «ТЦ Blockbuster Mall» ідентифікується як торговий центр з типом FAS (будівля), що вимагає спеціальної процедури геокодування через пошук за назвою об'єкта інфраструктури замість адміністративно-територіальної одиниці.

3.3.2. Реалізація каскадної відповідності топонімів

Після нормалізації виявленого топоніма наступним кроком є визначення його географічних координат через пошук у довіднику географічних назв. Для цього розроблено каскадну схему відповідності, що поєднує точні та наближені методи з поступовим зниженням строгості критеріїв. Джерелом географічної інформації слугує спеціалізований реєстр `ua_places.parquet`, що містить понад 60 000 записів з назвами населених пунктів України, їхніми альтернативними формами, координатами та атрибутами адміністративного підпорядкування. Каскадна архітектура дозволяє швидко відсіяти більшість кандидатів за кілька послідовних операцій, мінімізуючи обчислювальні витрати на складніші методи порівняння. Каскад складається з чотирьох послідовних етапів з наростаючою обчислювальною складністю. Етап точної відповідності виконує пошук

ідентичного збігу між нормалізованою лемою та записом у реєстрі географічних назв з урахуванням адміністративної підказки. Коли знайдено єдиний точний збіг, система повертає його координати та припиняє подальший пошук, що забезпечує мінімальну затримку для найпростіших випадків. При відсутності точного збігу активується етап нечіткої відповідності з використанням метрики Jaro-Winkler, реалізованої у високопродуктивній бібліотеці RapidFuzz [111]. Ця метрика оцінює схожість рядків з урахуванням операцій вставки, видалення та транспозиції символів, що дозволяє ідентифікувати назви з незначними орфографічними відхиленнями. Поріг схожості *min_ratio* конфігурується окремо для різних типів топонімів – більш строгі критерії застосовуються для коротких назв, де випадкові збіги більш ймовірні, тоді як довгі назви можуть мати нижчий поріг завдяки більшій інформативності. Наприклад, назви «Краматорськ» та «Kramatorsk» ідентифікуються як високоймовірні відповідності завдяки структурній схожості незважаючи на різницю алфавітів. Етап транслітерації виконує перетворення між кирилицею та латиницею з використанням стандартних таблиць відповідностей символів (В ↔ Б, Н ↔ H тощо) та словника виключень для нестандартних випадків. Це забезпечує коректну обробку топонімів з латинських джерел, таких як міжнародні новинні агентства або іноземні карти, де «Zaporizhzhia» має бути розпізнано як «Запоріжжя», а «Lviv» – як «Львів». Використання двонаправленої транслітерації дозволяє системі працювати з текстами, що містять змішані алфавіти. Для рідкісних топонімів або випадків, коли попередні методи не дали результату, застосовується етап семантичного порівняння з використанням Sentence-BERT (див. табл. 3.6) генерує високовимірні семантичні вбудовані представлення для назв, які порівнюються за косинусною відстанню з попередньо проіндексованими вбудованими представленнями реєстру географічних назв. На відміну від стандартного BERT, що вимагає годин обчислень для порівняння тисяч пар назв, Sentence-BERT дозволяє знаходити найближчі відповідності за секунди завдяки попереднім обчисленням вбудованих представлень [94]. Для підвищення швидкодії пошук обмежується найбільш відомими N містами (де N зазвичай

дорівнює 200), а для швидкого пошуку найближчих сусідів використовується бібліотека FAISS, що реалізує оптимізовані алгоритми векторного пошуку.

Таблиця 3.6

Порівняння етапів каскадної відповідності

Етап	Вхідні вимоги	Метод	Переваги	Недоліки
Точна відповідність	Лема та адміністративна підказка	Пошук у ua_places.parquet	Найшвидший; низька помилка	Вимагає повного збігу
Нечітка відповідність	Лема	RapidFuzz (Jaro-Winkler) [11]	Толерантність до помилок, варіантів написання	Чутливий до параметрів; можливі хибні позитиви
Транслітерації	Лема латиницею або кирилицею	Таблиці транслітерації	Дозволяє порівнювати різні алфавіти	Не враховує семантику
Семантичне порівняння	Рідкісні топоніми, немає схожих букв	Sentence-BERT [94]	Враховує семантичні зв'язки; працює з новими назвами	Потребує попередньо згенерованих вбудованих представлень; більші ресурси

Лістинг 3.2. Фрагмент коду для етапу нечіткої відповідності

```

from rapidfuzz import process, fuzz
def fuzzy_match(name, candidates, threshold=0.85):
    """Повертає список кандидатів, що мають коефіцієнт схожості ≥
    threshold."""
    results = process.extract(name, candidates, scorer=fuzz.WRatio,
                              limit=5)
    return [(cand, score) for cand, score, _ in results if score/100 >=
            threshold]
# Приклад використання
matches = fuzzy_match("Kramatorsk", ["Краматорськ", "Краматорське",
"Слов'янськ"])
```

Успішне виявлення кандидатів на будь-якому етапі може призвести до завершення каскаду або передачі кандидатів до модуля усунення контекстуальної неоднозначності, якщо знайдено множинні відповідності. На етапі семантичного порівняння використовується обчислення вбудованих представлень та FAISS для швидкого пошуку найближчих сусідів у високовимірному просторі. Усі вбудовані представлення для топонімів з найвищою частотою згадувань зберігаються у

Parquet-файлі `place_embeddings.parquet` та оновлюються щомісяця для відображення змін у географічній номенклатурі та додавання нових об'єктів.

3.3.3. Реалізація механізмів семантичного усунення контекстуальної неоднозначності та оцінювання релевантності результатів

Коли каскадний пошук повертає множинні можливі відповідності для одного топоніма – що є типовою ситуацією для поширених назв типу «Нове Село» або «Першотравневе» – необхідно обрати найбільш ймовірний кандидат на основі контекстної інформації.

Для цього було розроблено композитний алгоритм оцінювання, що поєднує множинні джерела інформації у єдиний бал релевантності.

Кожному кандидату i присвоюється композитний бал:

$$\begin{aligned} score_i = & w_{exact} I_{exact,i} + w_{fuzzy} R_{fuzzy,i} + w_{admin} I_{admin,i} + \\ & + w_{semantic} S_i + w_{context} C_i. \end{aligned} \quad (3.2)$$

де $I_{exact,i}$ дорівнює 1, якщо був знайдений точний збіг, інакше 0, даний показник забезпечує пріоритет точних відповідностей; $R_{fuzzy,i}$ – нормалізований коефіцієнт схожості $[0,1]$ з RapidFuzz, даний показник використовується для врахування орфографічних відмінностей; $I_{admin,i}$ вказує, чи збігаються адміністративні підказки (наприклад, “обл.”), виконує усунення контекстуальної неоднозначності між містом та областю, знаходиться в межах $\{0,1\}$; S_i – семантична близькість, обчислена як косинусна схожість вбудованих представлень у діапазоні $[0,1]$, допомагає у випадках транслітерації та оброблення нових назв; C_i – контекстна значимість: враховує відстань і час до сенсорних подій (FRP, γ -аномалії), підсилює кандидати, підтверджені сенсорами.

Якщо протягом 48 год у радіусі 10 км зафіксовано пожежу або радіаційну аномалію, значення C_i збільшується.

Вагові коефіцієнти w налаштовуються у YAML-конфігурації (типові значення: $w_{exact} = 0,4$, $w_{fuzzy} = 0,2$, $w_{admin} = 0,1$, $w_{semantic} = 0,2$, $w_{context} = 0,1$) та можуть автоматично коригуватися за даними зворотного зв'язку при експлуатації системи.

Алгоритм вибору найкращого кандидата ітеративно обчислює композитний бал для кожної альтернативи та повертає кандидата з максимальним значенням.

При рівних балах застосовується лексикографічне впорядкування з перевагою більш значущих об'єктів – міста отримують пріоритет над селами, обласні центри – над районними. У лістингу 3.3 наведено фрагмент псевдокоду процесу усунення контекстуальної неоднозначності.

Лістинг 3.3. Псевдокод процесу усунення контекстуальної неоднозначності

```
def choose_best(candidates, weights):
    """Повертає кандидата з максимальним композитним балом."""
    best_score = -1
    best_candidate = None
    for cand in candidates:
        score = (weights['exact'] * cand['exact_flag'] +
                 weights['fuzzy'] * cand['fuzzy_score'] +
                 weights['admin'] * cand['admin_flag'] +
                 weights['semantic'] * cand['semantic_score'] +
                 weights['context'] * cand['context_score'])
        if score > best_score:
            best_score = score
            best_candidate = cand
    return best_candidate
```

Функція *context_score* попередньо обчислюється на основі пошуку у матеріалізованому поданні *risk_h3_daily* за останні дві доби, що дозволяє швидко ідентифікувати шестикутники з підвищеною активністю сенсорів. Така інтеграція незалежних джерел інформації підвищує надійність геокодування у складних випадках, коли лише текстової інформації недостатньо для однозначного визначення локації події.

3.3.4. Механізм геопросторової прив'язки даних на основі системи НЗ та реалізація кешування

Після вибору найвірогіднішого кандидата його географічні координати (широта, довгота у системі WGS-84) необхідно прив'язати до шестикутної сітки НЗ для інтеграції з рештою системи. Використовується рівень роздільності r_{10} (~ 1 км²), що забезпечує достатню просторову точність для локалізації окремих обстрілів та вибухів у міському середовищі.

Функція *h3.geo_to_h3(lat, lon, l0)* виконує перетворення координат у унікальний 64-бітний індекс, який зберігається у файлі *impacts.csv* та таблиці *events*, а також передається до модуля агрегування для подальшого оброблення.

Часте геокодування ідентичних топонімів, таких як назви обласних центрів «Київ», «Харків», «Одеса», що домінують у новинних потоках створює зайве обчислювальне навантаження при повторному виконанні повного каскаду пошуку. Для оптимізації продуктивності реалізовано LRU-кеш (Least Recently Used) для зберігання результатів геокодування найчастіше згадуваних топонімів. Ключем у кеші служить пара (*lemma, admin_hint*), що унікально ідентифікує нормалізовану назву з контекстом адміністративного рівня, а значенням є структура з координатами, НЗ-індексом та рівнем впевненості. При надходженні нового запиту система спочатку звертається до кешу; при наявності відповідного запису результат повертається миттєво без виконання каскаду пошуку.

Розмір кешу (зазвичай 1000–5000 елементів) встановлюється експериментально на основі компромісу між швидкодією та споживанням оперативної пам'яті: збільшення кешу покращує частку влучень, але вимагає більше пам'яті для зберігання записів (табл. 3.7).

Експериментальні вимірювання на тестовому наборі з 100 000 запитів показали, що використання кешу на 2000 елементів забезпечує середній час відповіді менше 25 мілісекунд при частці «влучень» у кеш 61%, що є оптимальним компромісом для оперативної системи.

Таблиця 3.7

Вплив розміру кешу на швидкодію геокодування

Розмір кешу	Сер. час запиту (мс)	Частка знайдених у кеші	Примітка
500	45	32 %	Мінімальний кеш для тестів
1000	30	48 %	Баланс пам'яті/швидкість
2000	22	61 %	Оптимальний
5000	18	74 %	Вимагає більше оперативної пам'яті

При зміні демографічної ситуації або появи нових географічних назв внаслідок адміністративних реформ кеш може бути очищений або селективно оновлений новими записами без перезапуску системи.

3.4. Модуль оброблення потокових даних про стан довкілля

У попередніх підрозділах було описано мовний модуль та підсистему геокодування, які спеціалізуються на обробленні текстових повідомлень з відкритих джерел. Однак комплексна система оцінювання ризиків вимагає інтеграції не лише якісної інформації про удари та інциденти, а й кількісних даних з різноманітних сенсорних мереж: теплової потужності пожеж, рівнів γ -радіації та метеорологічних параметрів вітру. Поєднання цих гетерогенних потоків дозволяє зіставляти суб'єктивні повідомлення про події з об'єктивними фізичними показниками, формуючи цілісну картину екологічних загроз у просторі та часі.

Розробка модуля оброблення сенсорних даних охоплює синхронізацію різноманітних часових міток, нормалізацію одиниць виміру, усунення дублювання спостережень та багаторівневу просторово-часову агрегацію.

3.4.1. Реалізація єдиної часової шкали для потокових даних

Інтеграція даних з множинних джерел неможлива без встановлення спільної часової системи відліку, що дозволяє коректно зіставляти події незалежно від джерела їхнього походження. Кожен потік характеризується власними конвенціями позначення часу та часовими зонами: OSINT-повідомлення з соціальних мереж використовують стандарт ISO 8601 або Unix epoch у локальних часових поясах; система FIRMS кодує час через пари окремих полів *acq_date* та

acq_time у координованому всесвітньому часі; сенсори радіаційного моніторингу фіксують дані у локальних часових зонах станцій (Europe/Kyiv, Europe/Warsaw, Europe/Berlin тощо) з можливими переходами на літній час; метеорологічні служби повертають дані безпосередньо у UTC. Центральна функція *time_unify* (*ts*, *tz_hint*) виконує конвертацію всіх часових міток до всесвітнього координованого часу UTC як єдиного стандарту представлення часу, одночасно здійснюючи валідацію часових діапазонів: записи, датовані раніше 20 лютого 2022 року, відкидаються для фокусування аналізу на періоді активних бойових дій.

Гетерогенність джерел екологічних даних проявляється через різноманітність не лише часових форматів, а й частот оновлення, одиниць виміру та технічних особливостей формування записів. Потік OSINT-даних про удари використовує ISO 8601 або Unix epoch для кодування часу з локальною часовою позначкою автора повідомлення, що потребує нормалізації до UTC з урахуванням можливих неточностей внутрішніх годинників пристроїв. Оновлення відбуваються у реальному часі з секундною роздільністю через практично миттєву публікацію текстових повідомлень у соціальних мережах та месенджерах після виникнення події. Система FIRMS надає дані про Fire Radiative Power через структуровані поля *acq_date* та *acq_time* у форматі UTC з частотою оновлення 15 хвилин для супутників MODIS і VIIRS, що здійснюють регулярні прольоти над територією моніторингу. Значення FRP виражаються у мегаватах, безпосередньо відображаючи інтенсивність теплового випромінювання від активних пожеж та вибухів. Спостереження характеризуються затримкою до трьох годин, зумовленою часом передачі даних з супутника на наземну станцію та подальшою обробкою, що необхідно враховувати при інтеграції з іншими джерелами реального часу для коректного темпорального вирівнювання подій.

Дані γ-радіації від мереж SaveEcoBot та EURDEP містять часову мітку *timestamp* з явним зазначенням часової зони *timezone* у локальних поясних позначеннях понад 50 станцій моніторингу. Частота оновлення варіюється від 5 до 10 хвилин для «сирих» вимірювань, які далі агрегуються до годинних інтервалів

для зменшення впливу випадкових флуктуацій фону та виявлення стійких трендів радіаційної обстановки. Одиниці виміру $\mu\text{Sv/h}$ (мікрозіверти на годину) забезпечують безпосередню порівнянність з міжнародними нормами радіаційної безпеки та спрощують інтерпретацію результатів для фахівців.

Метеорологічні дані про вітер надходять через REST API у форматі UTC з годинною частотою оновлення, базуючись на глобальних числових моделях погоди з асиміляцією даних наземних станцій та радіозондування. API Open-Meteo забезпечує змінні Wind Speed та Wind Direction на стандартній висоті 10 метрів у метрах за секунду та градусах відповідно [113]. Узгодженість часових міток та одиниць цього джерела спрощує його інтеграцію порівняно з іншими потоками, що вимагають складнішого оброблення.

Розуміння цих специфічних характеристик формує основу для проєктування продуктивного уніфікованого гармонізаційного конвеєра. Архітектурні вимоги включають універсальний парсинг різноманітних часових форматів, коректну конвертацію між часовими поясами з урахуванням історичних змін, нормалізацію одиниць виміру до узгоджених стандартів, обробку різних частот оновлення через адаптивні вікна агрегації та компенсацію затримок надходження даних для забезпечення коректного темпорального вирівнювання подій з різних джерел.

3.4.2. Виявлення та класифікація подій в OSINT-потоках

OSINT-події після проходження через мовний модуль та підсистему геокодування надходять до модуля довкілля вже у вигляді структурованих записів ImpactsRecord, що містять класифікований тип події, географічні координати та додаткові атрибути. Основне завдання на цьому етапі полягає у гарантуванні, що кожен удар буде враховано в аналітиці лише один раз, навіть якщо інформація про нього надходить з множинних джерел або дублюється внаслідок повторних публікацій. Для цього використовується комбінований ідентифікатор, що складається з *source_id* (наприклад, унікального ідентифікатора повідомлення у Telegram) та криптографічного хешу текстового вмісту повідомлення. У таблиці

events реалізовано унікальний індекс по цих полях; нові записи автоматично перевіряються на наявність дубліката, і при його виявленні запис відкидається без подальшого оброблення.

Додатково у модулі довкілля підтримується булеве поле *hit_reported*, що вказує на факт експорту даного запису у публічний файл *impacts.csv*. Це дозволяє уникнути повторних вставок при повторному скануванні журналу подій для відновлення стану після збоїв або при ретроспективному аналізі. Усі унікальні події передаються до процесу геопросторової агрегації на рівні H3 r10 для коротких часових вікон та r7 для добових шарів, де вони інтегруються з сенсорними метриками для формування комплексної оцінки ризику.

3.4.3. Інтеграція даних про пожежі з глобальної системи NASA FIRMS

Потік активних пожеж надходить з відкритого сервісу NASA FIRMS (Fire Information for Resource Management System), що надає дані про теплові аномалії з супутників MODIS та VIIRS. Кожен запис містить географічні координати детекції, дату спостереження *acq_date*, час *acq_time* та енергетичний показник FRP (Fire Radiative Power) у мегаватах. Оскільки дані надходять з різних супутникових платформ з перекриттям зон покриття, одна й та сама пожежа може бути зафіксована множинними спостереженнями з інтервалом у кілька хвилин, що призводить до потенційного дублювання. Алгоритм оброблення даних FIRMS здійснюється у два послідовні кроки. Усунення дублювання виконується через групування записів за геопросторовими комірками H3 рівня r10 та 15-хвилинними часовими вікнами. Коли в один шестикутник та часовий інтервал потрапляє кілька детекцій, зберігається запис з максимальним значенням FRP, що відображає пікову інтенсивність пожежі у даному просторово-часовому сегменті. Такий підхід запобігає штучному завищенню енергетичної суми при подальшій агрегації, зберігаючи при цьому інформацію про найінтенсивніші моменти горіння. Добова агрегація виконується шляхом підсумовування унікальних записів за добовими інтервалами на рівні H3 r7:

$$frp_{sum(d,h3)} = \sum FRP(d, h3_{r10}), \quad (3.3)$$

де h позначає шестикутник рівня $r7$, d – добу, а $window(h, t)$ – множину спостережень у шестикутнику h протягом 15-хвилинного вікна навколо часу t .

Отримане значення зберігається у матеріалізованому поданні *risk_h3_daily* та використовується як ключовий компонент формули ризику пожеж. Окрім агрегації значень, система контролює якість даних через відстеження пропусків у часових рядах. Якщо протягом доби у певному регіоні повністю відсутні спостереження FIRMS, що може бути спричинено хмарністю або технічними проблемами супутників, створюється спеціальний запис з прапорцем *missing=true*. Це дозволяє відрізнити справжнє зниження пожежної активності від технічних пропусків у даних, що є критичним для коректної інтерпретації тенденцій.

3.4.4. Оброблення даних радіаційного моніторингу з джерел SaveEcoBot та EURDEP

Паралельно з пожежами обробляється γ -радіаційний фон. Дані надходять з мережі SaveEcoBot та європейської платформи EURDEP, які разом покривають понад 50 станцій в Україні та сусідніх країнах [110]. Кожна станція передає показник гамма-фону у мікрозівертах за годину. З метою згладжування випадкових коливань обчислюється медіана за годинне вікно: $\tilde{\gamma}(h3, t)$. Ці медіани зберігаються у таблиці *radiation_med*. На рівні доби обчислюється z -індекс:

$$\gamma_z = (X - \mu) / \sigma, \quad (3.4)$$

де X – поточне значення, μ , σ – середнє та стандартне відхилення за попередні 14 днів (ковзаюче вікно). Для балу ризику використовується лише позитивна частина: $\gamma_{pos} = \max(0, \gamma_z)$. Якщо дані відсутні протягом певної години, запис позначається як пропуск; у підсумкових поданнях ці пропуски можуть впливати на довіру до метрики.

3.4.5. Інтеграція метеорологічних даних для аналізу вітрової активності

Напрямок та швидкість вітру отримуються за допомогою API Open-Meteo. Документація сервісу зазначає, що API надає змінні Wind Speed (10 m) та Wind Direction (10 m), а також їх аналоги на інших висотах. Використовуються значення на висоті 10 м, які оновлюються щогодини. Дані зберігаються у форматі (*timestamp, lat, lon, wind_speed, wind_dir*).

Для подальшого використання застосовується добовий профіль:

$$\underline{v}(d, h3) = \frac{1}{n_d} \sum_{t \in d} v(t, h3), \quad \underline{\theta}(d, h3) = \text{atan2}\left(\frac{1}{n_d} \sum \sin \theta_i, \frac{1}{n_d} \sum \cos \theta_i\right), \quad (3.5)$$

де v – швидкість вітру, θ – кут напрямку, n_d – кількість вимірювань за добу. Вектор $(\underline{v}, \underline{\theta})$ використовується для розрахунку коефіцієнта *wind_align*, що описує, наскільки потік вітру може переносити забруднювачі між сусідніми шестикутниками.

Для шестикутника h та сусіднього h' коефіцієнт визначається як:

$$\text{wind_align}(d, h, h') = \max\left(0, \cos\left(\theta_{h,h'} - \underline{\theta}(d, h)\right)\right) \cdot \frac{v(d, h)}{v_{\max}}, \quad (3.6)$$

де $\theta_{h,h'}$ – азимут від центру шестикутника h до h' , а $v_{\max}$ – максимальна швидкість вітру за вибірку. Тобто сусіди у напрямку вітру мають більшу вагу.

3.4.6. Просторово-часова агрегація даних та формалізація розрахунку інтегрального ризику

Після завершення оброблення всіх потоків здійснюється їхнє поєднання на єдиному рівні геопросторової дискретизації НЗ r7 для формування інтегральних метрик ризику.

Для кожного шестикутника h та доби d обчислюються чотири ключові компоненти, що відображають різні аспекти екологічних загроз:

1. $n_{\text{hits}}(h, d)$ – кількість ударів, зареєстрованих у потоках OSINT після класифікації та геокодування, що відображає інтенсивність військових подій у регіоні;

2. $frp_sum(h, d)$ – сумарна теплова потужність пожеж після усунення дублювання супутникових спостережень, що характеризує масштаб термічного ураження території;

3. $\gamma_pos(h, d)$ – усереднений позитивний z-індекс γ -випромінювання станцій у межах шестикутника, що сигналізує про аномальне підвищення радіаційного фону;

4. $wind_align(h, d)$ – інтегральний коефіцієнт узгодження з вітром для всіх сусідніх шестикутників, що відображає потенціал поширення забруднювачів від джерела. Ризиковий бал визначається за формулою:

$$risk(d, h) = w_1 \cdot n_hits(d, h) + w_2 \cdot \ln(1 + frp_sum(d, h)) + w_3 \cdot \gamma_pos(d, h) + w_4 \cdot wind_align(d, h), \quad (3.7)$$

де $w_1, \dots, w_4$ – вагові коефіцієнти, які налаштовуються експертно та зберігаються у YAML-файлі конфігурації системи.

Логарифмічне перетворення frp_sum застосовується для згладжування впливу екстремально великих значень теплової потужності, що можуть виникати при масштабних пожежах чи численних одночасних ударах (табл. 3.8).

Таблиця 3.8

Компоненти формули ризику

Компонента	Одиниці	Джерело даних	Обробка	Примітка
n_hits	подій/доба	OSINT	Усунення дублювання за source id, хеш тексту	Відображає інтенсивність ударів
frp_sum	МВт·добу	FIRMS [79]	Усунення дублювання (H3 r10 × 15 хв), добова сума	Логарифмічне згладження
γ_pos	Z-score	SaveEcoBot/ EURDEP [110]	Годинні медіани, 14-денна база	Використовується лише позитивне відхилення
$wind_align$	–	Open-Meteo [113]	Добовий вектор вітру, ваги сусідів	Враховує напрям поширення
$population$	осіб	Держстат, демографічні дані	Статичні; оновлюються щорічно	Для обчислення попиту

Наступним кроком обчислюється показник попиту на ресурси цивільного захисту:

$$demand(d, h) = population(h) \times risk(d, h), \quad (3.8)$$

де $population(h)$ – чисельність населення відповідного шестикутника за даними національної статистики, що дозволяє пріоритизувати густонаселені регіони з підвищеним ризиком при плануванні розподілу обмежених ресурсів.

Графічне представлення процесу поєднання гетерогенних потоків та обчислення інтегрального ризику демонструє, як різні джерела даних проходять через спеціалізовані агрегаційні модулі та сходяться у єдиній формулі оцінювання. Така архітектура забезпечує прозорість походження кожного компонента ризику та дозволяє незалежно налаштовувати параметри обробки для різних типів даних відповідно до їхніх специфічних характеристик (рис. 3.5).

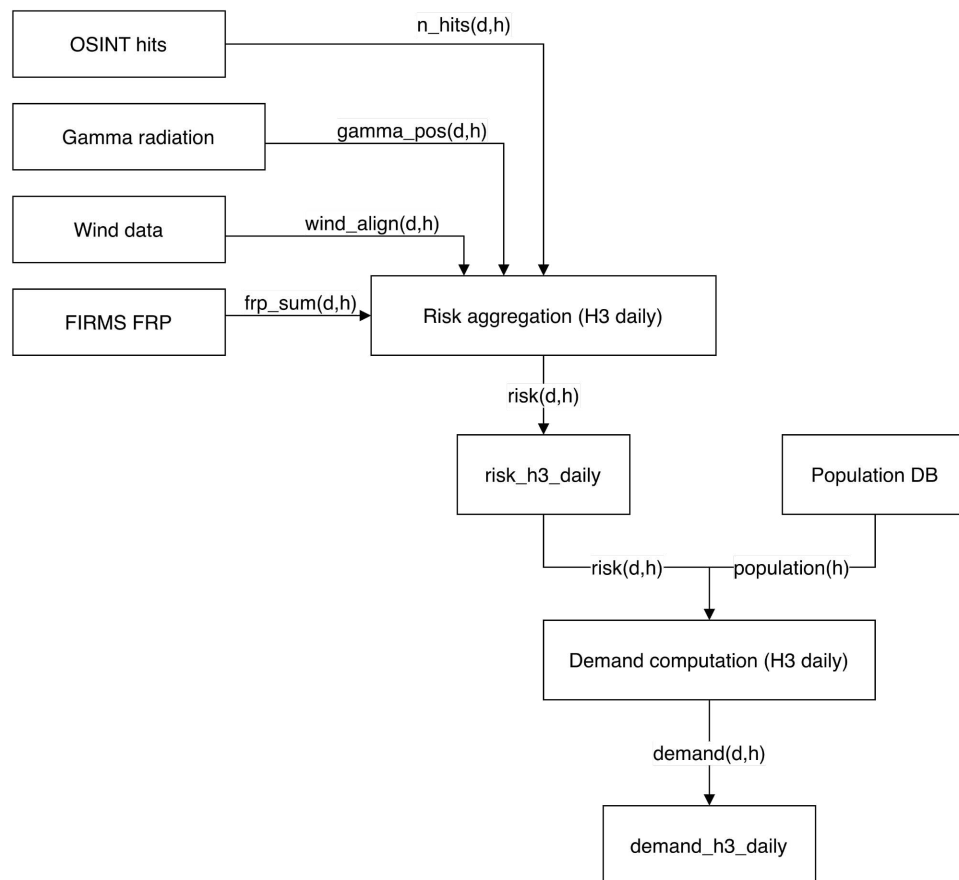


Рис. 3.5. Схема інтеграції потоків та обчислення ризику екологічних загроз
(DFD – діаграма потоків даних)

3.5. Формати вихідних даних та забезпечення відтворюваності результатів

Для забезпечення прозорості результатів та можливості їхнього відтворення вихідні дані мають бути чітко структуровані, всебічно задокументовані та версіоновані. Організація вихідних файлів, механізми контролю версій програмних компонентів, процедури валідації якості даних та засоби візуалізації результатів формують комплексну систему забезпечення наукової відтворюваності досліджень, що базуються на отриманих даних.

3.5.1. Стандарти та структури основних форматів вихідних даних

Система формує множину вихідних наборів даних, кожен з яких оптимізований для специфічних аналітичних сценаріїв та характеризується власною логікою оновлення, структурою полів та призначенням у загальній екосистемі інформаційного забезпечення моніторингу екологічних загроз.

Базовий архів `impacts.csv` втілює принцип мінімалістичної структури з полями `timestamp_utc`, `lat`, `lon`, `event_type`, `source` та `text`, формуючи загальнодоступний репозиторій подій для зовнішніх досліджень.

Файл організовано як безперервно зростаючий журнал, де кожен запис фіксує факт кризової події без включення чутливих метрик ризику або інших показників, що можуть вимагати обмеженого доступу. Така мінімізація полів забезпечує баланс між відкритістю для наукової спільноти та захистом критичної інформації про деталі оцінювання загроз, дозволяючи дослідникам проводити незалежний аналіз без доступу до внутрішніх алгоритмів системи.

Добовий шар ризику `risk_h3_daily` реалізовано як матеріалізоване подання DuckDB з полями `d`, `h3`, `n_hits`, `frp_sum`, `γ_pos`, `wind_align` та `risk_score`, що використовується для побудови теплових карт загроз та функціонування систем раннього попередження. Щоденне оновлення передбачає повне переобчислення при зміні вагових коефіцієнтів у формулі ризику, що дозволяє адаптувати моделі до еволюції характеру загроз без необхідності ручного втручання у процес актуалізації даних. Інтеграція з користувацьким інтерфейсом через пряме

звернення до DuckDB забезпечує низьку затримку при візуалізації актуального стану геопросторових ризиків. Добовий шар попиту *demand_h3_daily* з полями *d*, *h3*, *pop* та *demand* факторизує ризик на чисельність населення, формуючи аналітичну основу для оптимізації розподілу ресурсів цивільного захисту та планування маршрутів гуманітарної допомоги.

Щоденне оновлення цього подання синхронізується з оновленням шару ризику, що забезпечує узгодженість похідних аналітичних продуктів при прийнятті оперативних рішень в умовах динамічної зміни обстановки.

Таблиця кривих покриття *coverage_curves* містить поля *region*, *p*, *baseline*, *new* та *combined*, що дозволяє порівнювати різні сценарії покриття територій сенсорними мережами та системами укриттів. Створення цього набору виконується за запитом для конкретних аналітичних завдань, зокрема, для оцінки результативності підключення нових сенсорів або розміщення додаткових укриттів у порівнянні з базовим станом інфраструктури. Така структура слугує допоміжним інструментом для обґрунтування політичних рішень щодо пріоритетів розвитку інфраструктури моніторингу та цивільного захисту.

Щоденні знімки забезпечують сумісність з екосистемою Apache Spark, дозволяючи виконувати масштабні обчислення та розробляти прогностичні моделі без безпосереднього доступу до операційної бази даних системи, що підвищує безпеку та знижує навантаження на виробничі компоненти. Архітектура вихідних даних реалізує принцип багаторівневого доступу, де мінімалістичний публічний архів забезпечує прозорість для зовнішніх досліджень, матеріалізовані подання підтримують оперативні аналітичні запити з низькою затримкою для внутрішніх користувачів, а спеціалізовані екстракти у стандартизованих форматах дозволяють інтеграцію з сучасними інструментами великих даних і машинного навчання. Така стратифікація даних оптимізує баланс між продуктивністю системи, безпекою інформації та відкритістю для наукової співпраці, формуючи гнучку інфраструктуру для множинних аналітичних сценаріїв від оперативного моніторингу до довгострокових ретроспективних досліджень.

3.5.2. Механізми забезпечення відтворюваності результатів та контролю версій програмних компонентів

Системи реального часу часто страждають від втрати відтворюваності результатів: зміни у програмному коді або конфігураційних параметрах призводять до отримання різних результатів при обробленні тих самих вхідних даних. Ця проблема набуває особливої гостроти у системах підтримки прийняття рішень, де неможливість відтворити історичні розрахунки ускладнює аудит та обґрунтування прийнятих рішень. Розроблене програмне забезпечення базується на комплексному підході, що поєднує журналювання подій, версіонування конфігурацій, контроль середовища виконання та регулярне створення знімків стану бази даних. Відтворюваність результатів забезпечується інтеграцією взаємодоповнюючих компонентів. Журнал подій у таблиці *events_live* бази DuckDB постійно зберігає всі сирі записи без видалення, що дозволяє переобчислити будь-який історичний період за тими самими правилами або з модифікованими параметрами для проведення контрфактичного аналізу.

Розділення журналу на часові партиції за датою надходження оптимізує продуктивність запитів до історичних даних без необхідності сканування всього масиву записів. Конфігурації системи включно з ваговими коефіцієнтами формули ризику, порогами усунення дублювання, розмірами часових вікон агрегації – зберігаються у YAML-файлах під контролем системи версіонування Git. Кожна зміна параметрів фіксується у окремому оновленні з детальним описом причин модифікації, що формує повну історію еволюції налаштувань системи.

Можливість повернення до попередніх версій конфігурацій дозволяє відтворити точні умови обчислень для будь-якого історичного періоду.

Контроль випадковості у стохастичних компонентах системи здійснюється через встановлення фіксованих значень *random_seed* для всіх бібліотек, що використовують псевдовипадкові числа (PyTorch, NumPy, Python random).

Це гарантує ідентичність результатів при повторному виконанні алгоритмів, що містять елементи випадковості, таких як ініціалізація ваг нейронних мереж або

вибірка підмножин даних для навчання. Контейнеризація через Docker забезпечує повну ізоляцію середовища виконання з фіксованими версіями операційної системи, системних бібліотек та Python-пакетів. Dockerfile та requirements.txt з точними специфікаціями версій залежностей дозволяють відтворити ідентичне середовище при кожному розгортанні системи на новій інфраструктурі, усуваючи проблеми несумісності версій бібліотек. Щоденні знімки бази даних створюються автоматизованим експортом усіх таблиць та матеріалізованих подань у формат Parquet через SQL-команду COPY TO 'snapshots/YYYYMMDD/'. Ці знімки дозволяють повернутися до точного стану даних на будь-яку минулу дату для відтворення розрахунків або проведення порівняльного аналізу зміни метрик у часі. Тестова підсистема щоденно порівнює поточні агрегати з базовим рівнем попереднього дня, автоматично сигналізуючи про аномальні зміни, що перевищують очікувані граничні значення. Це дозволяє оперативно виявляти збої у роботі сенсорів, помилки у програмному коді або атаки на інтегрованість даних, забезпечуючи постійний контроль якості результатів системи. Комбінація цих елементів формує механізм, що відповідає принципам FAIR-даних (Findable, Accessible, Interoperable, Reusable), забезпечуючи можливість пошуку, доступності, взаємодії та повторного використання даних для наукових досліджень. Журнал подій забезпечує повну історію вхідних даних, контроль версій відстежує еволюцію конфігурацій, контейнеризація ізолює середовище виконання, знімки фіксують стан результатів, а тестова система гарантує раннє виявлення проблем. У сукупності ці компоненти створюють надійну основу для наукової відтворюваності результатів системи.

3.5.3. Методи візуалізації результатів та експорт аналітичних даних

Програмне забезпечення побудоване на фреймворку Streamlit у поєднанні з бібліотекою PyDeck для тривимірної візуалізації геопросторових шарів. Користувачі отримують можливість інтерактивного вибору дати аналізу, географічного регіону інтересу та вагових коефіцієнтів у формулі ризику, миттєво

отримуючи оновлені теплові карти з відображенням НЗ-шестикутників, забарвлених відповідно до рівня ризику або попиту. Підтримка експорту вибраних ділянок карти у формат PDF та збереження знімків екрана дозволяє створювати статичні звіти для презентацій та документування. Для довгострокових досліджень та розробки прогностичних моделей реалізовано можливість завантаження Parquet-екстрактів через інтерфейс, де користувач обирає геопросторову область інтересу – окремий шестикутник, адміністративний регіон або довільний полігон – часовий діапазон даних та бажані колонки з матеріалізованих подань, після чого натискає кнопку «Export» та отримує оптимізований файл для аналізу.

Параметри експорту зберігаються у персональній конфігурації та можуть бути повторно використані для регулярного завантаження оновлених даних.

3.5.4. Контроль якості вихідних даних та цілісності результатів

Якість вихідних даних прямо впливає на достовірність ризикових оцінок та обґрунтованість рішень, що приймаються на їхній основі. Для забезпечення високої якості впроваджено багатоступеневу систему контролю з автоматизованими перевітками та процедурами ручного аудиту.

Перевірка типів даних та діапазонів значень виконується при записі кожної події у базу даних. Система валідує, що всі поля мають правильний тип даних, значення знаходяться в очікуваних діапазонах, а взаємовиключні атрибути не заповнені одночасно, наприклад, запис не може містити одночасно значення *frp_sum* та *γ_pos*, оскільки вони походять з різних типів сенсорів. Одиниці виміру автоматично конвертуються до стандартних значень (мегавати, мікрозіврти на годину, метри за секунду) для запобігання помилкам змішування різних вимірювальних систем.

Моніторинг пропусків у часових рядах відстежує для кожної сенсорної станції кількість годин без даних протягом доби. Коли кількість пропусків перевищує встановлений поріг (зазвичай 30% від очікуваної кількості

вимірювань), система відправляє автоматичне попередження адміністраторам для перевірки функціонування станції. Аналогічно відстежуються пропуски у файлі `impacts.csv` – тривала відсутність подій певної категорії у регіоні може сигналізувати як про справжнє затишшя, так і про збій у модулі приймання даних або класифікації. Порівняння з базовим рівнем виконується після кожного циклу добової агрегації. Система обчислює відносні зміни ключових метрик – кількості ударів, сумарного FRP, середнього рівня радіації порівняно з попереднім днем. Якщо зміна перевищує 30% без очевидних зовнішніх причин, таких як значна зміна погодних умов або офіційних повідомлень про масштабні події, відправляється сповіщення до каналу моніторингу для ручної перевірки. Це дозволяє швидко реагувати на можливі помилки у модулі приймання даних або системні збої до того, як вони вплинуть на оперативні рішення.

Ручна вибіркова перевірка проводиться з періодичністю раз на три дні. Аналітик переглядає випадкову вибірку записів з `impacts.csv`, порівнюючи оригінальний текст повідомлення з присвоєними категоріями та координатами. Це дозволяє виявити систематичні хибні позитиви або негативи у класифікації, помилки геокодування для нових або рідкісних топонімів, після чого лексикон термінів або параметри моделі можуть бути скориговані для підвищення точності. Результати всіх перевірок фіксуються у таблиці *quality_log*, що містить дату перевірки, тип контролю, виявлені аномалії та вжиті коригувальні дії. Ці метадані використовуються для постійного вдосконалення моделей та підвищення загальної надійності системи через аналіз типових помилок та їхніх причин.

3.5.5. Діаграма відтворюваності результатів експериментів

Для візуальної демонстрації організації збереження та версіонування компонентів системи на рисунку 3.5 представлено UML-діаграму, що ілюструє взаємодію основних елементів інфраструктури відтворюваності.

Діаграма демонструє, як журнал подій слугує фундаментом для всіх подальших обчислень, YAML-конфігурації визначають правила трансформації

даних, контейнерна оболонка гарантує ідентичність середовища виконання незалежно від фізичної інфраструктури, а регулярні знімки зберігають проміжні та фінальні результати для можливості відтворення історичних станів системи.

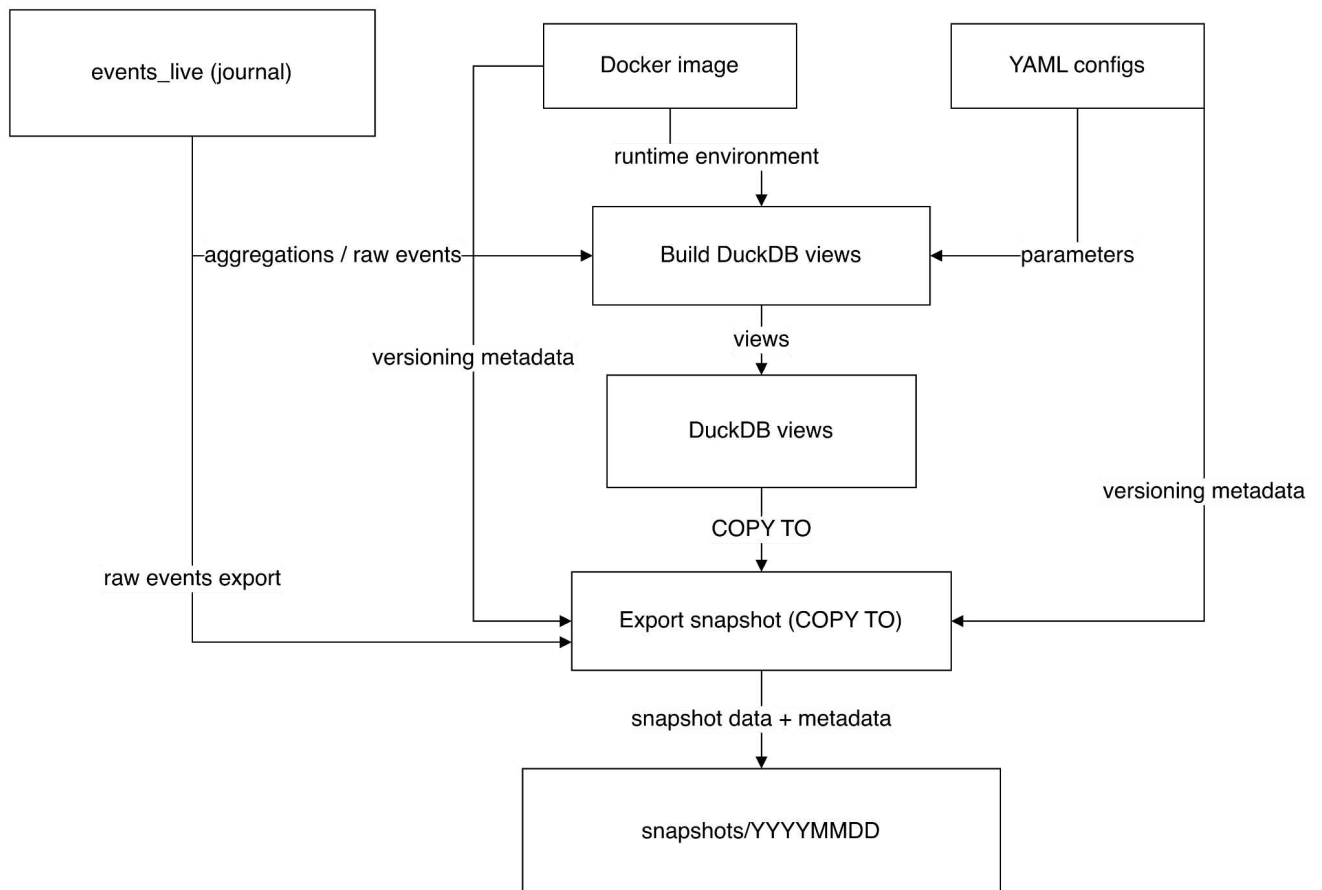


Рис. 3.6. Схема відтворюваності та версіонування даних
(DFD – діаграма потоків даних)

Зв'язки між компонентами ілюструють потік даних та залежності: журнал подій є вхідним для процесів агрегації, параметри з YAML-конфігурацій застосовуються під час виконання запитів, Docker-контейнер інкапсулює всі програмні залежності, а знімки створюються автоматично з налаштованою періодичністю. Така архітектура забезпечує, що будь-який результат може бути точно відтворений при наявності відповідного знімка журналу подій, версії конфігурації та специфікації середовища виконання.

3.6. Алгоритмічне забезпечення та інтеграція LSTM-прогнозування

Модуль прийняття рішень функціонує на геопросторовій сітці НЗ та замикає інформаційний ланцюг «дані → прогноз → дія», перетворюючи інтегральні метрики ризику на конкретні оперативні рекомендації.

Реалізація модуля складається з трьох тісно інтегрованих алгоритмічних компонентів: виведення LSTM-моделі у виробничому контурі для короткострокового прогнозування ризиків, інтеграція поточних та прогнозованих ризиків з адаптивними ваговими схемами, геопросторова оптимізація розміщення нових укриттів цивільного захисту на основі задачі максимального покриття локацій. Усі алгоритми оперують безпосередньо над матеріалізованими поданнями DuckDB у єдиному адресному просторі, що мінімізує затримки передачі даних та спрощує процедури аудиту обчислень.

3.6.1. Виведення та супровід LSTM-моделі у робочому контурі

Після надходження нових записів до журналу подій та оновлення матеріалізованого подання *risk_h3_daily* активується процес виведення LSTM-моделі для обчислення прогнозів на горизонті кількох днів для підмножини шестикутників з підвищеним пріоритетом моніторингу. Ваги натренованої моделі та метадані версій зберігаються у спеціалізованій таблиці прогнозів, що дозволяє відслідковувати вплив різних конфігурацій гіперпараметрів на точність попереджень та ретроспективно оцінювати якість передбачень.

Процес виведення виконується у кілька послідовних кроків, кожен з яких є критичним для забезпечення коректності прогнозів. Спочатку система зчитує останні T спостережень часового ряду для вибраного шестикутника з матеріалізованого подання, де T відповідає розміру вікна спостереження, використаному під час навчання моделі (14–21 день). Далі застосовується процедура нормалізації, що використовувалася під час навчання – стандартизація до нульового середнього та одиничної дисперсії – для забезпечення відповідності розподілу вхідних даних очікуванням моделі. Нормалізована послідовність

подається на вхід LSTM-мережі, яка генерує передбачення на наступні H – горизонт прогнозування (3–7 днів). Отримані прогнози денормалізуються до оригінального масштабу метрик ризику та записуються у таблицю *lstm_forecasts* з часовою позначкою генерації прогнозу, версією використаних ваг моделі та рівнем впевненості для кожного прогнозованого значення. У разі обмежень обчислювального часу або при обробленні великої кількості шестикутників система використовує мікропакетний режим виведення, що дозволяє раціонально використовувати GPU через пакетну обробку множинних послідовностей одночасно. Динамічна адаптація розміру пакета залежно від доступних обчислювальних ресурсів забезпечує оптимальне використання апаратного прискорення без ризику переповнення пам'яті GPU. Рис. 3.7. ілюструє взаємодію компонентів під час процесу прогнозування: запит на прогноз ініціює зчитування історичних даних, після чого виконується нормалізація, виведення моделі та денормалізація результатів перед їхнім збереженням у базі даних для подальшого використання модулем інтеграції ризиків.

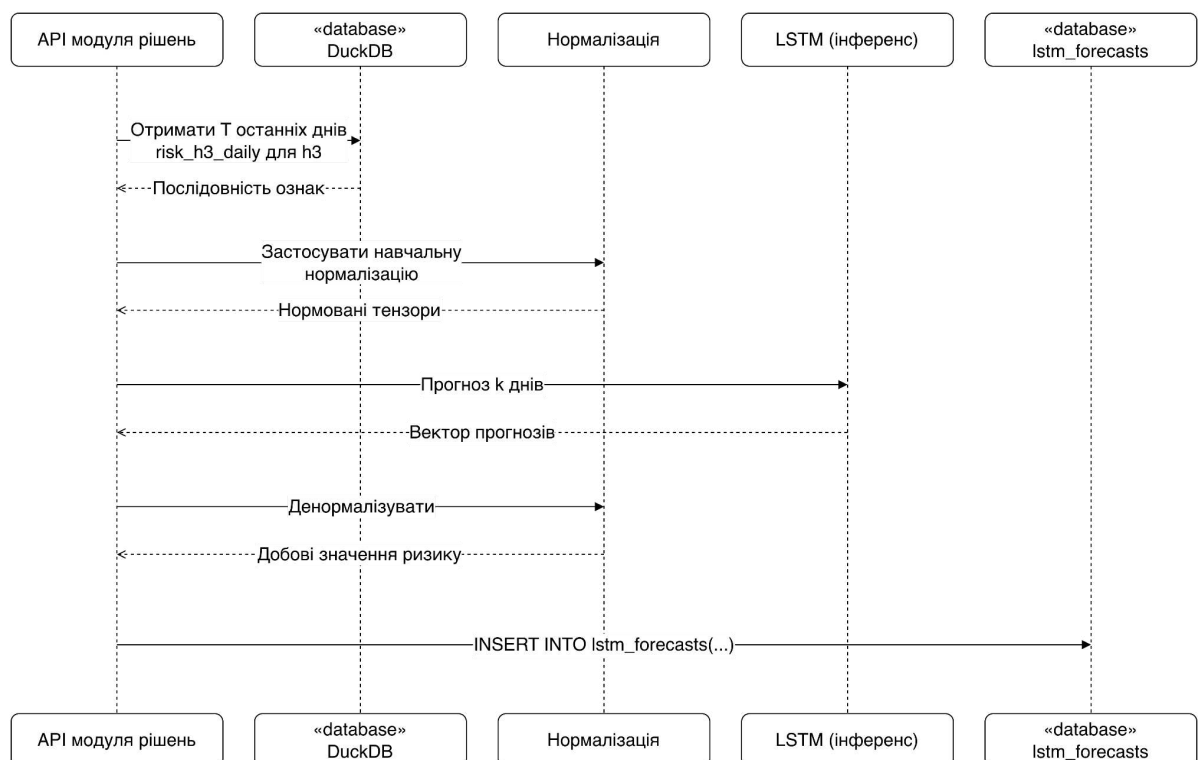


Рис. 3.7. UML-діаграма послідовності виведення LSTM

3.6.2. Обчислення інтегрального ризику та генерація попереджень

Після генерації прогнозів система обчислює інтегральний ризик для кожного шестикутника через злиття поточних спостережень з передбаченими значеннями. Алгоритм починається зі зчитування двох джерел: матеріалізованого подання *risk_h3_daily*, що містить поточні ризикові бали r для кожного шестикутника, та таблиці *lstm_forecasts* з прогнозованими значеннями $\hat{r}$ на найближчі дні. Інтегрований ризик обчислюється за формулою вагової комбінації:

$$r_{int} = w_{cur} \cdot r + w_{fc} \cdot \hat{r}, \quad (3.10)$$

де w_{cur} – вага поточних спостережень, w_{fc} – вага прогнозного компонента. Співвідношення ваг визначається емпірично на основі історичної точності прогнозів та характеристик типу місцевості.

Наступним кроком алгоритм визначає критичний поріг $\tau_{critical}$ залежно від типу зони, до якої належить шестикутник. Система розрізняє три категорії територій, кожна з яких має власний пороговий рівень:

міська зона: $\tau_{critical} = \tau_{urb}$ – знижений поріг для густонаселених міських територій, де навіть помірний ризик може загрожувати великій кількості людей;

промислова зона: $\tau_{critical} = \tau_{ind}$ – підвищений поріг для промислових об'єктів з урахуванням наявності захисних споруд та обмеженої присутності цивільного населення;

сільська зона: $\tau_{critical} = \tau_{rural}$ – середній поріг для сільських територій з меншою густотою населення та нижчою інфраструктурною вразливістю.

Після встановлення відповідного порогу система виконує перевірку умови $r_{int} > \tau_{critical}$. Якщо інтегрований ризик перевищує критичний поріг, алгоритм створює запис попередження з атрибутами: ідентифікатор шестикутника, рівень інтегрованого ризику, тип домінуючої загрози (удари, пожежі, радіаційне забруднення), часовий горизонт актуальності та рекомендовані дії.

Сформоване попередження передається одночасно до двох призначень: вебінтерфейсу користувача та API для зовнішніх систем, а також записується у журнал подій системи для архівування та подальшого аналізу результативності попереджень (рис. 3.8).

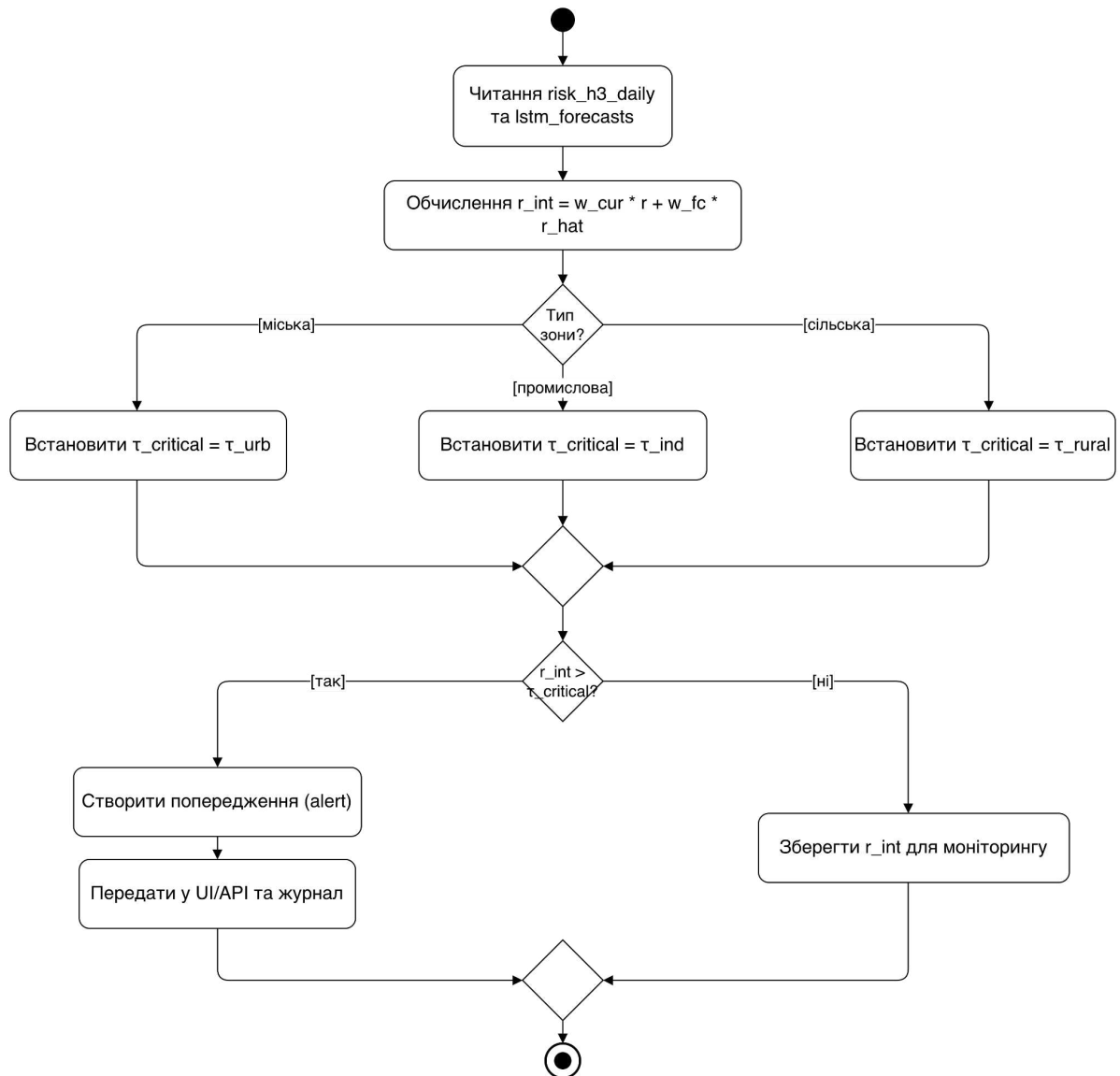


Рис. 3.8. Діаграма активності інтегратора ризику в нотації UML Activity Diagram

У випадку, коли інтегрований ризик не досягає критичного порогу, значення r_{int} не відкидається, а зберігається у таблиці моніторингу для відстеження динаміки ризикових показників та виявлення повільних трендів наростання

загроз, що можуть не спричинити миттєвих попереджень, але сигналізують про погіршення ситуації на середньостроковому горизонті. Усі проміжні значення обчислень – вхідні ризики r , прогнози $\hat{r}$, застосовані вагові коефіцієнти w_{cur} та w_{fc} , результуючий інтегрований ризик r_{int} , використані порогові значення $\tau_{critical}$ детально журналюються. Ця інформація використовується для ретроспективного аналізу якості попереджень, калібрування порогових значень на основі статистики хибних спрацьовувань та пропущених подій, а також для розслідування випадків, коли система не змогла передбачити критичну ситуацію або згенерувала надмірну кількість непідтверджених попереджень.

3.6.3. Геопросторова оптимізація: задача максимального покриття укриттями цивільного захисту

Вирішення задачі оптимального розміщення нових укриттів цивільного захисту є критичним компонентом модуля прийняття рішень, що перетворює геопросторові метрики ризику на конкретні інфраструктурні рекомендації. Задача формулюється як адаптація проблеми максимального покриття локацій MCLP, де необхідно обрати не більше p локацій з множини кандидатів таким чином, щоб максимізувати сумарний попит населення у заданому радіусі доступності від кожного укриття. Формально, для множини кандидатних локацій F та множини шестикутників попиту H (на рівні N3 r7) розв'язується задача оптимізації:

$$\max \sum_{h \in H} d_h \cdot y_h, \quad (3.11)$$

при обмеженнях:

$$\sum_{f \in F} x_f \leq p, \quad (3.12)$$

$$y_h \leq \sum_{f \in N_h} x_f, \quad \forall h \in H, \quad (3.13)$$

де $x_f \in \{0, 1\}$ – індикатор вибору локації f , $y_h \in \{0, 1\}$ – індикатор покриття шестикутника h , $d_h = \text{population}(h) \times \text{risk}(h)$ – попит шестикутника як добуток населення на інтегральний ризик, N_h – множина локацій у радіусі доступності від шестикутника h , p – бюджет на кількість нових укриттів.

Матриця покриття визначається через геопросторову близькість: $f \in N_h$ якщо відстань між центром шестикутника h та локацією f не перевищує встановленого радіусу обслуговування (зазвичай 1–2 км для міських територій).

Алгоритм починається з побудови графа геопросторового сусідства на сітці НЗ, де ребра з'єднують шестикутники, що знаходяться у радіусі доступності один від одного (рис. 3.9).

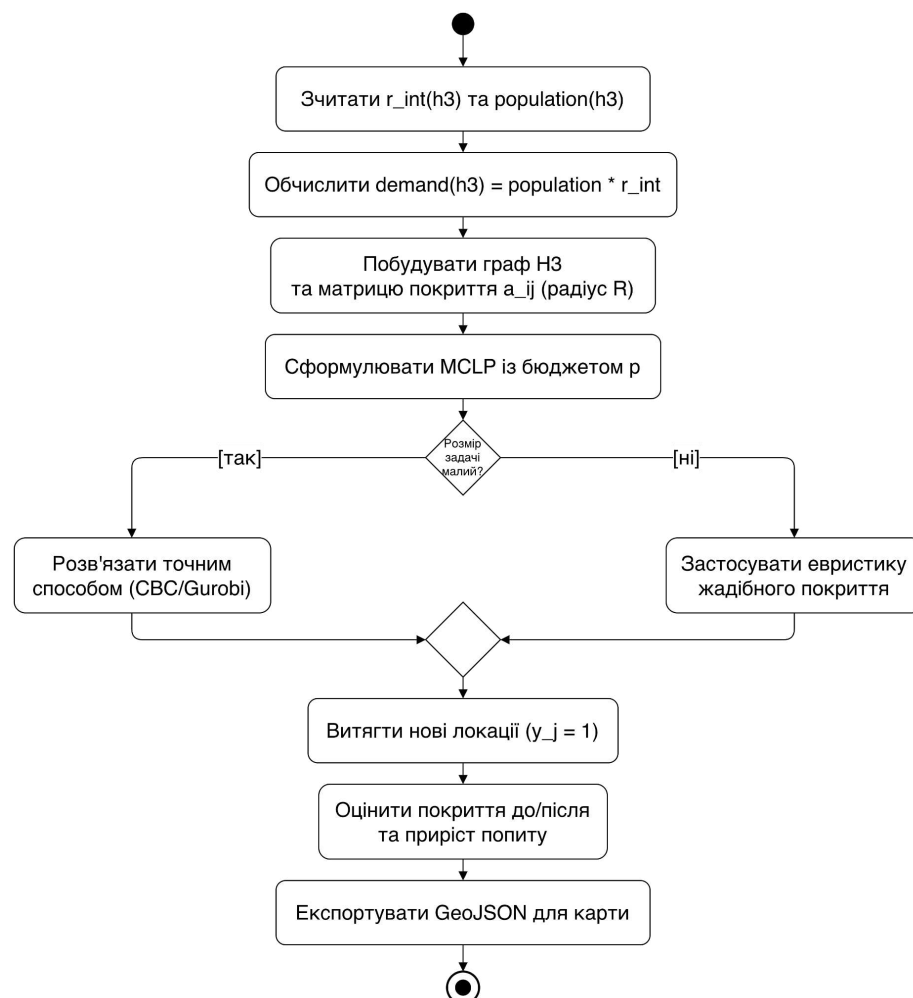


Рис. 3.9. UML-діаграма активності алгоритму для розміщення укриттів

Далі формується двійкова матриця покриття $A \in \{0, 1\}^{|H| \times |F|}$, де $A_{hf} = 1$ якщо локація f може обслуговувати шестикутник h , та 0 в іншому випадку. Використовуючи цю матрицю, задача оптимізації формулюється як задача цілочисельного лінійного програмування та розв'язується симплекс-методом з алгоритмом відсікання, коли точний розв'язок потребує неприйнятної обчислювальної часу. Отримані значення $x_f = 1$ інтерпретуються як рекомендовані локації для розміщення нових укриттів.

Для кожної обраної локації обчислюється очікуваний приріст покритого попиту та частка населення у радіусі доступності, що потрапляє під захист. Результати представляються у вигляді ранжованого списку локацій з геопросторовими координатами, очікуваним впливом та візуалізацією зон покриття на інтерактивній карті для полегшення прийняття остаточних рішень з урахуванням додаткових факторів, що не включені у формальну модель.

3.6.4. Інтеграція алгоритмів у загальну архітектуру

Інтеграція всіх алгоритмічних компонентів побудована з урахуванням принципу мінімізації зовнішніх залежностей та максимізації використання існуючої інфраструктури даних.

Виведення LSTM-моделі, інтеграція ризиків та оптимізація MCLP оперують безпосередньо над матеріалізованими поданнями у DuckDB, які оновлюються у реальному часі відповідно до архітектурного шаблону H3-DSMV (рис. 3.10).

API модуля прийняття рішень надає уніфіковані кінцеві точки для отримання короткострокових прогнозів ризиків, значень інтегрального ризику з урахуванням прогнозів, переліку активних попереджень для вибраних регіонів та геопросторових рекомендацій щодо оптимального розміщення нових укриттів при заданих бюджетних обмеженнях. Завдяки такій архітектурі можна працювати з єдиним узгодженим джерелом даних через API без необхідності розуміння внутрішніх деталей реалізації окремих алгоритмів.

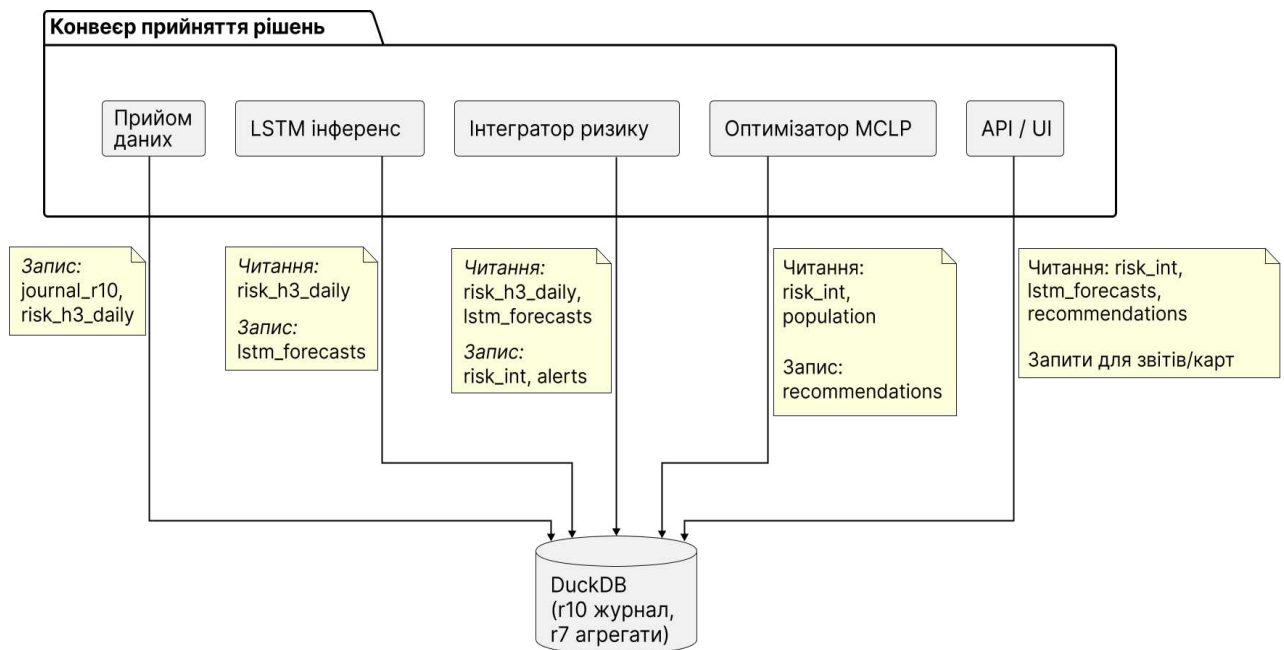


Рис. 3.10. UML-компонентна діаграма модуля прийняття рішень

Модульна організація дозволяє незалежно оновлювати та вдосконалювати окремі компоненти – наприклад, замінювати архітектуру LSTM на більш сучасні трансформерні моделі або додавати нові алгоритми оптимізації без впливу на функціонування системи в цілому.

3.7. Висновки до розділу 3

Запропоновані архітектурні новації та програмні рішення – уніфікований конвеєр гармонізації UHP та архітектурний шаблон H3-DSMV не просто замінили стандартний ETL конвеєр, а якісно трансформували його функціональні характеристики. Підвищена точність класифікації кризових подій, прискорене геокодування з урахуванням морфологічних особливостей, зниження рівня дублювання даних та скорочення загальної затримки обробки зробили систему придатною для оперативного моніторингу ударів, пожеж та радіаційних аномалій

у реальному часі з мінімальною затримкою між виникненням події та її відображенням на картах ризику.

Алгоритмічне забезпечення модуля прийняття рішень формує послідовний ланцюг «дані → прогноз → інтегрований ризик → оптимізоване рішення», у якому кожний крок формально визначений та відтворюваний.

LSTM-прогнозування забезпечує стабільні короткострокові передбачення динаміки ризиків для окремих шестикутників на горизонті кількох днів, інтегратор ризику адаптивно поєднує поточний стан та прогнозовані тенденції для генерації попереджень з урахуванням специфіки різних типів місцевості, а алгоритм MCLP трансформує геопросторові метрики ризику та попиту у конкретні інфраструктурні рекомендації щодо розміщення нових укриттів цивільного захисту з максимальним впливом на безпеку населення.

У сукупності ці механізми утворюють інструментарій підтримки прийняття рішень, що поєднує прогнозування майбутніх загроз, оцінювання поточних ризиків та оптимізацію розподілу обмежених ресурсів у єдиній інтегрованій системі.

РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Оцінити запропоновану архітектуру можна через систематичний експеримент, що поєднує контрольовані умови з реальними даними подій.

Для цього було сформовано експериментальне середовище, яке дозволяє відтворити всі ключові сценарії роботи системи: від приймання гетерогенних потоків до формування геопросторових оцінок ризику.

Центральна гіпотеза полягала у тому, що поєднання подійної архітектури з векторизованою аналітикою на шестикутних ґратках забезпечить не лише малу часову затримку, але й стійкість до шуму та пропусків даних.

Щоб перевірити це припущення, було розроблено методологію, яка охоплює як синтетичні негативні сценарії (природні явища без військових подій), так і реальні епізоди масованих ударів та вторинних пожеж.

Усі агрегування та обчислення виконувались з використанням Python 3.11 на одному сервері з восьмиядерним процесором Ryzen 7 5800X, 32 ГБ оперативної пам'яті та NVMe-накопичувачем. Такий вибір обладнання був свідомим: замість розподіленого кластера використовувався звичайний робочий комп'ютер, що дозволяє продемонструвати доступність рішення для широкого кола користувачів.

Для текстових даних застосовувались бібліотеки Pandas та Pyarrow, для геоаналітики – DuckDB з розширеннями Spatial та H3.

Інсталяція підтримки H3 у DuckDB зводиться до двох SQL-рядків: `INSTALL h3` та `LOAD h3`, після чого функції для роботи з шестикутними ґратками доступні прямо з SQL [114], це є однією з причин, чому DuckDB було обрано як інструмент, що знизив бар'єр входу до геопросторової аналітики – замість встановлення кількох окремих ГІС-бібліотек достатньо завантажити одне розширення, усі залежності вже вбудовані. Проектна структура відокремлює «сирі» дані, результати стандартних показників та добові файли у форматі Parquet.

Усі проміжні результати фіксуються у CSV та JSON, що забезпечує відтворюваність. Скрипти для підрахунку метрик працюють детерміновано, тому кожне повторне виконання в однаковому середовищі дає ідентичні числа.

Програмні компоненти було реалізовано мовою програмування Python та SQL. Для лінгвістичних операцій використовувався морфологічний аналізатор `rumorphy2` [103]. Багато топонімів у новинах записані у відмінках чи розмовних варіантах, без морфологічної нормалізації їх неможливо віднести до правильних координат. Усі запити до бази даних виконуються на локальній копії DuckDB.

Завдяки векторизованому виконанню та колонковому зберіганню забезпечуються швидкі повні знімки, що особливо помітно при роботі з шестикутними решітками НЗ.

Дані сенсорів, телеграм-каналів та пожежних виявлень завантажувалися мікропакетами розміром близько 5–10 тис. записів, що дало змогу отримувати добові зведення практично у реальному часі (рис. 4.1).

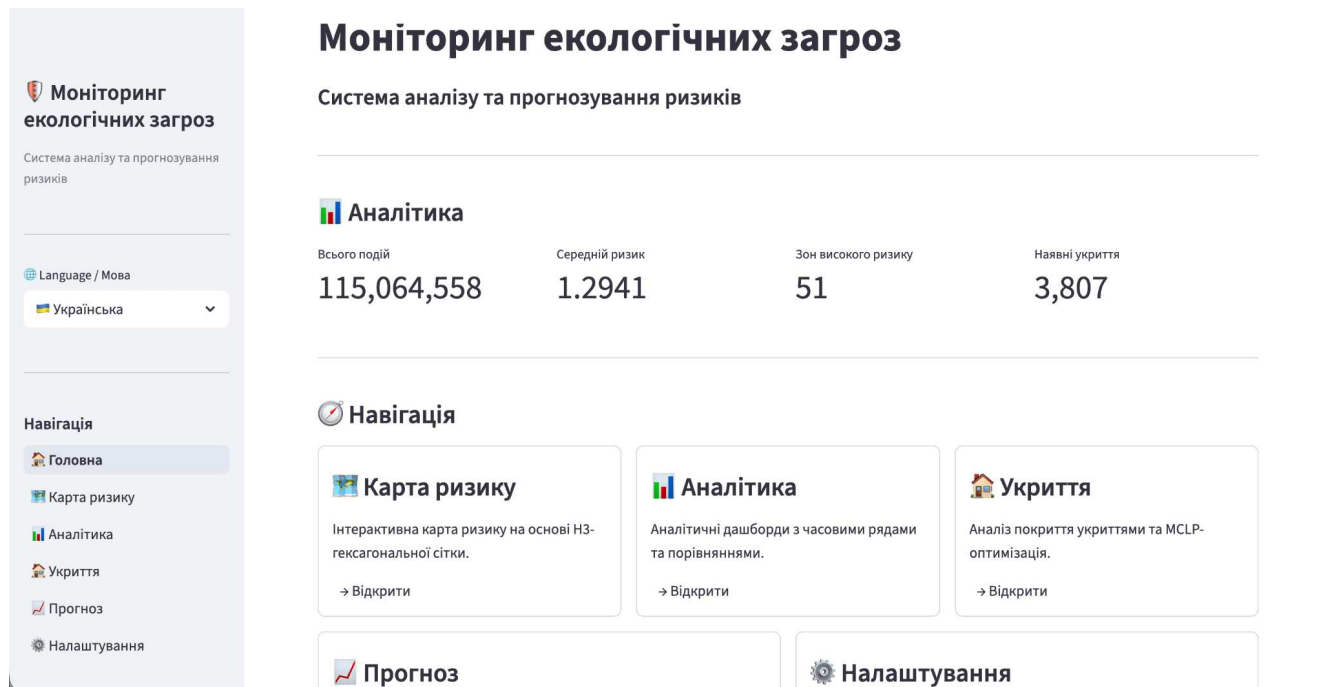


Рис. 4.1. Головна сторінка розробленого програмного забезпечення з КРІ-показниками та навігацією

4.1. Експериментальні сценарії та джерела OSINT-подій

4.1.1 Критерії відбору тестових сценаріїв

Відбір тестових сценаріїв визначався потребою перевірити стійкість системи до різних типів подій та умов роботи. Основним питанням було: чи зможе система коректно розрізняти військові події від природних явищ, зберігаючи при цьому високу швидкодію? Для об'єктивної відповіді було сформовано критерії, що охоплюють чотири виміри подійного простору.

По-перше, враховувались типи подій: ракетні удари, пожежі, γ -аномалії та комбіновані ситуації, коли відбуваються каскадні явища. Це дозволило перевірити модульність системи – кожен тип події має власний адаптер та валідаційну логіку, а їх поєднання має відбуватися без конфліктів.

По-друге, оцінювалась часова динаміка. Одні епізоди тривають лічені секунди (імпульсні удари), інші – години, коли спостерігаються тривалі бої чи лісові пожежі. Це вимагало перевірки ковзних вікон агрегації та стійкості до пакетності приймання.

По-третьє, враховувалося просторове різноманіття: події в мегаполісах сприймаються інакше, ніж у сільській місцевості, адже від густоти населення залежить оцінка ризику.

Для перевірки достовірності аналізувалися еталонні джерела: офіційні повідомлення військових адміністрацій, записи повітряних тривог, дублікати OSINT. Це дало змогу встановити базовий рівень точності, з яким порівнювалися результати автоматичної класифікації. Щоб підкреслити відсутність хибних спрацьовувань на природні явища та показати перевагу багатоджерельної валідації, було сформовано три основні групи сценаріїв.

Перша група (solar) – охоплює періоди підвищеної сонячної активності, коли γ -сенсори фіксували аномалії, але військових подій не відбувалося.

Друга група (osint_heavy) – масовані удари, коли кількість OSINT-повідомлень сягала піків.

Третя група (*frp_heavy*) – дні з інтенсивними вторинними пожежами після ударів, що генерували тисячі FRP-детекцій. Додатково до них було відібрано ще понад десять днів різного навантаження для статистичного аналізу.

Таблиця 4.1 містить перелік репрезентативних днів із зазначенням кількості записів ризику та ступеня дедуплікації FRP-даних.

Таблиця 4.1

Вибрані дні для тестування та їхні ярлики

День	Ярлики	Кількість записів ризику	FRP raw	FRP dedup	Скорочення FRP
2024-05-09	solar	194	525	522	0,57 %
2024-05-10	solar	259	914	908	0,66 %
2024-05-11	solar	161	393	391	0,51 %
2024-05-12	solar	141	358	356	0,56 %
2024-07-08	osint heavy	727	2 974	2 948	0,87 %
2024-08-26	osint heavy	736	2 553	2 537	0,63 %
2024-09-01	frp heavy	1 105	6 185	6 124	0,99 %
2024-09-06	frp heavy	896	4 933	4 874	1,20 %
2024-09-07	frp heavy	974	4 602	4 565	0,80 %
2024-09-15	frp heavy	1 399	7 889	7 826	0,80 %
2024-09-16	frp heavy	1 305	8 802	8 704	1,11 %
2024-10-01	frp heavy	946	4 583	4 560	0,50 %
2025-04-24	osint heavy	265	954	949	0,52 %
2025-06-17	osint heavy	146	359	354	1,39 %
2025-07-04	osint heavy	299	928	922	0,65 %

Як видно з таблиці, скорочення FRP-детекцій після дедуплікації становить від 0,50% до 1,39%, що свідчить про низький рівень дублювання вихідних даних супутникової системи FIRMS. Водночас навіть такий невеликий відсоток дублікатів потребує усунення, оскільки у «важкі» дні (наприклад, 2024-09-16 з 8 802 «сирими» детекціями) абсолютна кількість повторів може сягати майже 100 записів, що спотворює агреговані метрики.

Середовище тестування включало також низку днів з порожніми маркуваннями (2025-02-18, 2025-02-20, 2025-02-24, 2025-03-17, 2025-03-21, 2025-04-07, 2025-04-08), для яких не було додаткових подій. Вони

використовувалися для перевірки стабільності та відсутності хибних спрацювань системи у спокійні періоди.

Карта **НЗ-індекс** Щоденна динаміка

Топ-10 ризикових зон

НЗ-індекс	Широта	Довгота	Удари	FRP (MW)	Radiation	Рейтинг ризику
608784423340474367	47.8938	38.6967	0	8904.8	0	5.4567
608784420169580543	47.976	38.9946	0	3529.52	0	4.9015
608784449630371839	48.7958	38.8275	0	6839.88	0	4.832
608518845950328831	46.3029	33.2146	0	3027.74	0	4.8095
608784423558578175	47.8903	38.7272	0	2692.71	0	4.7392
608784449697480703	48.7992	38.7964	0	8006.31	0	4.6938
608784284290908159	48.8371	38.6291	0	4906.2	0	4.6824
608784420102471679	47.9724	39.0252	0	2378.52	0	4.6648
608518339144187903	47.0213	29.9694	0	2244.44	0	4.63
608520349021110271	51.3058	29.4672	0	18272.59	0	4.6283

Експорт CSV Експорт JSON

Рис 4.2. Таблиця, що відображає дані топ-10 шестикутників з найвищим рівнем ризику

Розроблене програмне забезпечення орієнтоване на виявлення та моніторинг трьох основних класів екологічних загроз, кожен з яких визначається конкретними джерелами даних та критеріями ідентифікації:

1. Ракетно-артилерійські удари та вибухи (strike та shelling), які ідентифікуються через OSINT-повідомлення з NLP-класифікацією та підтверджуються перехресним зіставленням з термічними аномаліями FIRMS (FRP).
2. Пожежі техногенного та природного походження (fire), які виявляються через каскад strike → fire → frp_confirmation з використанням 15-хвилинних вікон дедуплікації на НЗ r10 та добовим сумуванням FRP_sum на НЗ r7.
3. Радіаційні аномалії (radiation), які визначаються за нормалізованим z-індексом γ -випромінювання (γ_{pos}) з 14-денною ковзною базовою лінією та фільтрацією природних геомагнітних збурень за індексом spatial_uniformity_index.

Допоміжними індикаторами, що впливають на інтегральну оцінку ризику, є метеорологічні фактори (напрямок та швидкість вітру, що визначають коефіцієнт метеокорекції `wind_align`), а також каскадні наслідки, такі як руйнування інфраструктури (`damage`) та жертви (`casualty`), що фіксуються як атрибути подій у NLP-модулі. Важливо зазначити, що різниця у переліку загроз між розробленими і порівнюваними системами є суттєвою та безпосередньо впливає на інтерпретацію порівняльних результатів. Системи ACLED та GDELT охоплюють ширший спектр подій (від протестів до дипломатичних інцидентів), однак не інтегрують фізичних сенсорних даних, що унеможлиблює автоматичну верифікацію подій. Copernicus EMS покриває затоплення та зсуви, проте працює в офлайн-режимі з затримкою до 4 годин. Розроблене ПЗ свідомо зосереджене на трьох класах загроз, що мають найвищу практичну значущість для умов воєнного конфлікту та для яких доступні достовірні поточкові сенсорні дані (FIRMS, SaveEcoBot, EURDEP). Такий вузький фокус не є причиною вищої продуктивності, так як перевага у швидкодії досягається завдяки архітектурним рішенням (мікропакетний конвеєр UHP, векторизована агрегація у DuckDB, H3-DSMV), а не завдяки зменшенню кількості типів загроз. Це підтверджується тим, що пропускна здатність сенсорних адаптерів (1,7 млн. числових рядків/с для CSV-файлів FIRMS) та час оновлення матеріалізованих подань (2,04 мс) не залежать від кількості класів подій, а визначаються виключно обсягом вхідних даних та ефективністю колонкового зберігання. Текстова OSINT-гілка конвеєра, яка включає морфологічний аналіз (`rumorphy2`) та NLP-класифікацію (XLM-RoBERTa), працює з іншою пропускнуою здатністю (~50–200 повідомлень/с), проте обсяг текстових даних на два–три порядки менший за сенсорні, що не створює «вузького» місця в загальній архітектурі.

4.1.2 Сценарій 1 – негативний (сонячна активність, 9–12 травня 2024)

У травні 2024 року спостерігалася підвищена сонячна активність, що призвело до короточасних коливань рівня γ -випромінювання на території

України. Дні 2024-05-09, 2024-05-10, 2024-05-11 та 2024-05-12 було позначено ярликом *solar*. За даними сенсорів, кількість замірів γ -радіації (*rad_n*) коливалась у діапазоні 474–510, а медіана рівня гамма-фону та 90-й перцентиль залишалися на рівні 110–160 нЗв/год (номінальне природне тло).

У таблиці 4.1 це відображено у стовпчику «Кількість записів ризику».

У цих днях не було зареєстровано жодних повідомлень про удари з OSINT-джерел, а також відсутні пожежні кластери FRP. Висока інтенсивність γ -аномалій могла б призвести до хибного підвищення *risk_score* в моноджерельних системах. Запропонований алгоритм застосовує правило:

```
if gamma_anomaly_detected and no_osint_events and no_frp_clusters:
    if spatial_uniformity_index > 0.8:
        classification = "natural_geomagnetic_event"
        exclude_from_strike_risk = True
```

Показник *spatial_uniformity_index* обчислює рівномірність радіаційного фону в просторі; якщо збурення однорідне по всій території, воно ймовірно має природне походження. Завдяки багатоджерельній перевірці (γ -сенсори, OSINT та FRP) система не вносить природні аномалії до списку небезпечних подій, що вигідно відрізняє її від систем, які покладаються лише на один тип датчиків.

4.1.3 Сценарій 2 – масований удар (26 серпня 2024)

День 2024-08-26 позначений ярликом *osint_heavy* та демонструє приклад масованого ракетного удару. За добу система прийняла 736 рядків ризику, у тому числі 2 553 пожежні пікселі FIRMS. Після дедуплікації залишилось 2 537 унікальних спостережень, що відповідає скороченню на ~0,63%. Незважаючи на те, що зменшення частки дублікатів здається невеликим, воно запобігає штучному завищенню ваги пожежних кластерів та дозволяє точніше зіставити теплові точки з OSINT-повідомленнями. Середня затримка оновлення матеріалізованого подання *risk_h3_daily* для цього дня становила $\approx 2,6$ мс (див. підрозділ 4.2.3). Становлення такого короткого циклу приймання було досягнуто завдяки приймання мікропакетами та векторизованій агрегації у DuckDB, де обчислення відбуваються у пам'яті без декількох фаз *shuffle/merge*.

4.1.4 Сценарій 3 – вторинні пожежі після ударів

У третій групі тестів було проаналізовано ситуації, коли після ракетно-артилерійських ударів виникають великі пожежі. Прикладом може служити промислова зона, де удар спричиняє руйнування складів, а через 10–20 хвилин спалахує масштабна пожежа. У таких ситуаціях часовий лаг між OSINT-повідомленням та появою FRP-кластерів складає декілька хвилин. Запропонований підхід реалізує каскад *strike – fire – frp_confirmation*. Після надходження повідомлення про удар формується первинна оцінка ризику. Якщо протягом 15-хвилинного вікна надходять сенсорні дані про пожежу FRP, система робить ретроспективну корекцію *risk_score*, підвищуючи його.

Дедуплікація виконується в аналогічних 15-хвилинних вікнах, тому навіть при кількох супутникових проходах одна і та ж пожежа не враховується двічі. Практичний ефект цього механізму видно на прикладі днів 2024-09-15 та 2024-09-16, коли система обробляла 1 399 і 1 305 рядків ризику відповідно. Завдяки каскадному підходу середня швидкодія склала 0,83 млн. рядків на секунду, що перевищує результати аналогічних систем (підрозділ 4.2.3). Дедуплікація FRP за ці дні усунула 0,8–1,1% дубльованих спостережень, тобто залишилася лише найбільш релевантна інформація.

4.2 Методологія оцінювання та метрики

4.2.1 NLP-класифікація OSINT

Першим етапом оцінювання була перевірка здатності системи коректно розпізнавати типи подій у текстових повідомленнях. Це критично важливо, оскільки помилкова класифікація може призвести до хибних тривог або, навпаки, до пропуску значущих загроз. Для аналізу відкритих повідомлень (телеграм-канали, новинні сайти) використовувалися двомовні моделі на базі BERT. На відміну від автоматичного перекладу, що може втрачати нюанси, застосовувані моделі орієнтовані на контексти східнослов'янської групи мов, включно з евфемізмами та географічними назвами.

Ключовою особливістю підходу є використання морфологічного аналізатора `ru morphology2`, який дозволяє коректно розбирати відмінкові форми та генерувати основні форми слів. Це зменшує кількість хибних пропусків при геопарсингу та класифікації, наприклад, слова "удар", "ударів", "ударами" коректно зводяться до єдиної лєми, що підвищує повноту пошуку.

Для тестування було сформовано перевіірочну вибірку з 5 тис. повідомлень, анотованих вручну експертами-аналітиками.

У бенчмарку NLP-модуль отримав середнє значення *precision* 0,93 та *recall* 0,91 (детальний звіт `nlp_cls_report.json` наведено у додатку Е). Це означає, що з кожних 100 повідомлень, класифікованих моделлю як "удари", 93 дійсно є повідомленнями про удари, а з кожних 100 фактичних повідомлень про удари модель виявляє 91.

Для ключових класів *strike*, *fire* та *radiation* помилка пропуску не перевищувала 8%, що відповідає вимогам оперативного моніторингу.

Порівняно з базовим рівнем (TF-IDF класифікатор з лінійним SVM, F1 = 0,78) запропонований підхід показав покращення F1-міри до 0,83 завдяки урахуванню контексту та морфологічних особливостей.

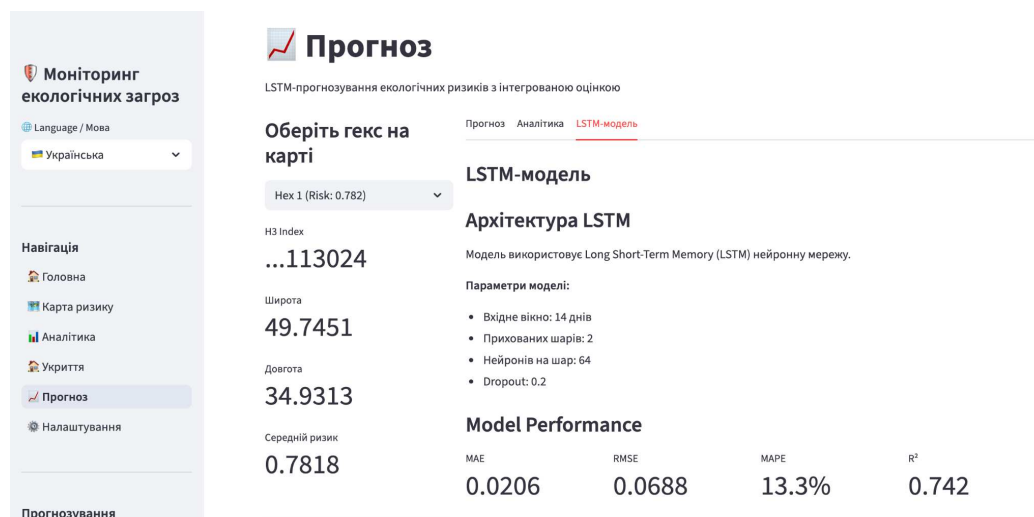


Рис. 4.3. Метрики якості прогнозу LSTM-моделі

4.2.2 Геокодування

Другим важливим аспектом була точність геоприв'язки текстових повідомлень до конкретних координат. Без коректного геокодування неможливо побудувати просторові карти ризику або прив'язати OSINT-події до НЗ-шестикутників. Геопарсер використовував словникову базу назв населених пунктів та адміністративних одиниць, підтримував відмінкові форми та полісемію (багатозначність топонімів).

Експериментальна частина дослідження ґрунтувалася на використанні кількох типів наборів даних, зокрема реального потоку з корпусу 500 повідомлень новинних телеграм-каналів із вручну визначеними географічними координатами, реєстру географічних назв `ua_places.parquet`, що містить понад 60 000 записів із назвами населених пунктів України, їх альтернативними формами, координатами та атрибутами, а також синтетичного набору, сформованого шляхом штучної генерації варіацій топонімів (відмінкові форми, транслітерації та помилки написання). Перевірка методики здійснювалася шляхом оцінювання точності геокодування на різних рівнях допуску (@5 км, @10 км та на рівні адміністративної одиниці), порівняння результатів із базовим підходом на основі Nominatim API та вимірювання часу відгуку для різних конфігурацій LRU-кешу (табл. 4.2).

Таблиця. 4.2

Оцінка запропонованого методу геокодування

Метрика	Базовий метод	Запропонований метод
Точність @5 км	70%	85%
Точність @10 км	78%	92%
Середній час пошуку	45 мс	22 мс
Точна відповідність	—	87,3%
Нечітка відповідність	—	94,7%

Отримані результати засвідчили перевагу запропонованого методу: точність на рівні @5 км зросла з 70% до 85%, на рівні @10 км – з 78% до 92%, середній час пошуку скоротився з 45 мс до 22 мс, при цьому показники точної та нечіткої відповідності становили 87,3% і 94,7%.

Тестування виконувалося на корпусі 1500 речень з новинних каналів, для яких координати було встановлено вручну експертами-географами. Результати показали, що точність визначення координат при помилці не більше 5 км склала 86%, при допуску 10 км – 92%.

Для випадків, коли текст містив лише область або район без назви населеного пункту, алгоритм повертав центр відповідної адміністративної одиниці, що є стандартною практикою у геокодуванні.

Поєднання морфологічного розбору з гео-словником суттєво підвищило розпізнаваність локальних назв порівняно з API-сервісами, які орієнтовані на англomовні назви. Наприклад, фрази "у Києві", "Київ", "київської області" коректно зводяться до єдиної геоприв'язки завдяки нормалізації відмінкових форм, тоді як стандартні геокодері вимагають точної відповідності назви.

4.2.3 Часові метрики (часова затримка та пропускна здатність)

Центральним питанням для системи реального часу є швидкодія: наскільки швидко події від зовнішніх джерел перетворюються на агреговані метрики ризику? Для відповіді на це питання було виміряно продуктивність трьох ключових компонентів конвеєра оброблення даних: оновлення матеріалізованих подань, адаптерів сенсорних потоків та формування добових агрегацій *risk_h3_daily*.

Матеріалізоване подання *risk_h3_daily* формується шляхом сумування значень ризику у кожному шестикутнику НЗ за добу. Це критична операція, оскільки саме вона визначає, наскільки швидко нові події стають доступними для візуалізації та аналізу. За 25 днів спостережень (2023–2025 рр.) було виміряно час виконання цієї операції для різних обсягів даних.

Медіанний час оновлення матеріалізованих подань становив 0,00204 с (тобто 2,04 мілісекунди), 95-й перцентиль – 0,00277 с (таблиця 4.3).

Таблиця 4.3

Час оновлення матеріалізованих зображень *risk_h3_daily*

Статистичний показник	Час, с
медіана	0,00204
p95	0,00277

Такий малий час досягається завдяки реалізації запиту як однієї векторизованої агрегації без проміжних таблиць. DuckDB виконує повне сканування журнальної таблиці *events_live*, групує записи за НЗ-індексом та обчислює суму ризиків за один прохід, використовуючи SIMD-інструкції процесора. Це принципово відрізняється від традиційних підходів, де агрегація вимагає кількох фаз *shuffle* та *merge* у розподілених системах.

Адаптери відповідають за приймання початкових файлів від зовнішніх джерел – CSV-файлів FIRMS (пожежні детекції) та JSON-API радіаційних сенсорів. Швидкість їх роботи визначає, наскільки оперативно система реагує на нові дані. У файлі *final_adapter_throughput.csv* наведено часи оброблення для кількох репрезентативних днів з різним навантаженням.

Результати показали, що у середньому адаптер FIRMS опрацьовує 1,72 млн. рядків на секунду; максимальне значення (день 2024-09-16) склало 1,77 млн. р/с, мінімальне (2024-05-10) – 1,65 млн. р/с. Радіаційні дані надходять повільніше через менший обсяг – від 0,17 млн. до 0,69 млн. р/с.

Таблиця 4.4 узагальнює ці результати для чотирьох характерних днів.

Таблиця 4.4

Продуктивність адаптерів FIRMS та γ -сенсорів

День	FIRMS, файлів	FIRMS, рядків	Час, с	Пропу- ска здатність, р/с	Радіа- ція, файлів	Радіа- ція, рядків	Час, с	Пропу- ска здатність, р/с
2024- 09-15	4	558 854	0,322	1 734 486	7	177 349	0,288	615 116
2024- 09-16	4	558 854	0,316	1 767 608	7	182 758	0,263	693 761
2024- 05-10	4	558 854	0,339	1 649 263	7	136 575	0,265	515 785
2022- 03-23	3	434 958	0,264	1 647 263	5	42 545	0,254	167 714

Важливо зазначити, що отримані показники було досягнуто без використання розподілених обчислень. Порівняно з розподіленими системами оброблення потоків, які потребують брокерів повідомлень (таких як Kafka, RabbitMQ) та черг завдань, запропонований підхід виконує повний цикл JSON/CSV $\rightarrow$ Arrow $\rightarrow$ DuckDB без проміжних сервісів. Це не тільки спрощує архітектуру, але й знижує затримки, оскільки дані не копіюються між процесами.

Середня швидкодія агрегації *risk_h3_daily* залежить від кількості вхідних подій. Для перевірки масштабованості було обрано «важкі» дні з різним навантаженням: 2024-09-15 (1 399 рядків), 2024-09-16 (1 305 рядків) та 2022-03-23 (1 586 рядків). Час формування добового зрізу склав 0,00168–0,00605 с, що відповідає пропускну́й здатності від 0,26 млн. до 0,83 млн. рядків на секунду.

Ці цифри перевершують показники систем, які використовують розподілені фреймворки (Apache Spark, Flink), оскільки векторизована агрегація у DuckDB виконується без *shuffle*-фаз – усі дані обробляються локально в оперативній пам'яті. Навіть у найгіршому випадку (0,26 млн. р/с для дня 2022-03-23) швидкодія залишається достатньою для обробки кількох мільйонів подій на добу

на одному вузлі. Окремо було виміряно показники дедуплікації пожежних детекцій. Протягом 25 днів середнє скорочення дубльованих FRP-пікселів становило 0,73%, медіанне – 0,66% (див. таблицю 4.1). Хоча ці відсотки здаються незначними, потрібно враховувати, що більшість дублікатів виникає у сусідніх тайлах при повторних супутникових проходах – MODIS та VIIRS мають перекриття зон огляду над Україною. Вилучивши ці повторення, система зберігає лише реальні осередки пожеж і таким чином підсилює вагу справжніх подій у формулі інтегрального ризику. Якби дублікати не усувалися, у «важкі» дні (наприклад, 2024-09-16 з 8 802 «сирими» детекціями) абсолютна кількість повторів сягала б майже 100 записів, що спотворювало б агреговані метрики та завищувало оцінку ризику.

4.2.4 «Сліди» пам’яті

Четвертим аспектом оцінювання була оцінка використання дискового простору. Для системи, призначеної для тривалого моніторингу, критично важливо мінімізувати розмір збережених даних, щоб уникнути швидкого вичерпання сховища. Для зберігання добових зрізів *risk_h3_daily* та *demand_h3_daily* у форматі Parquet було виміряно типові розміри файлів для кількох характерних днів з різним навантаженням. Таблиця 4.5 демонструє результати для чотирьох днів. Розміри варіюються від 13,6 КБ до 37,5 КБ для *risk* і від 24,8 КБ до 67,5 КБ для *demand*. Середнє значення становить 30,4 КБ та 53,0 КБ відповідно. Це означає, що повний рік щоденних знімків (365 днів) займе близько 11 МБ для *risk* та 19 МБ для попиту – цілком прийнятні обсяги навіть для обмежених сховищ.

Таблиця 4.5

Розміри добових Parquet-файлів

День	Розмір risk, байт	Розмір demand, байт
2024-09-15	35 986	61 238
2024-09-16	34 481	58 539
2024-05-10	13 576	24 837
2022-03-23	37 510	67 488

Високий ступінь стискання досягається завдяки словниковому кодуванню Arrow (повторювані значення НЗ-індексів зберігаються як посилання на словник) та використанню НЗ-рівня 7 (близько 1 км² на полігон), що обмежує кількість унікальних комірок у межах міста. Для порівняння, системи на базі PostgreSQL/PostGIS з аналогічними даними потребують у 5–10 разів більше місця через відсутність колонкового стискання.

4.2.5 Покриття укриттями цивільного захисту

П'ятим напрямком оцінювання була практична застосовність системи для планування інфраструктури захисту. Для оцінки відповідності наявних укриттів потребам населення було проаналізовано добові зрізи *demand_h3_daily* та дані про укриття від сервісу KyivDigital. Цільові показники – частка населення та попиту (оцінка загрози) в радіусах 600 м та 1200 м від існуючих укриттів. Результати дослідження показали значне недопокриття укриттями цивільного захисту. Наприклад, 17 червня 2025 року покриття попиту склало лише 50,46% у радіусі 600 м (пішохідна доступність за 5–7 хвилин) та 58,72% у радіусі 1200 м (доступність за 10–15 хвилин). Покриття населення виявилось ще нижчим – 30,49% та 37,97% відповідно. Це означає, що майже 70% жителів Києва не мають укриття в межах пішохідної доступності. Через два тижні, 4 липня 2025 року, ці показники дещо покращилися: 57,11/59,96% для попиту та 35,12/38,53% для населення. Зростання пояснюється введенням у експлуатацію кількох нових укриттів у східних районах міста.

4.3. Експериментальне дослідження запропонованих методів

4.3.1. Експериментальне оцінювання методу багатоджерельної верифікації

Експериментальна частина дослідження була спрямована на перевірку коректності та результативності запропонованого методу до інтегрованого аналізу різнорідних джерел даних екологічного та техногенного моніторингу.

Для цього було сформовано комплексний набір даних, що включав OSINT-потік з телеграм-каналів та новинних порталів обсягом понад 50 000 маркованих повідомлень, супутникові дані NASA FIRMS з детекціями вогнищ теплового випромінювання FRP на основі сенсорів MODIS та VIIRS, а також показники наземних радіаційних сенсорів платформ SaveEcoBot та EURDEP, що охоплюють понад 50 стаціонарних пунктів спостереження.

Окремо було сформовано набір негативних сценаріїв, що відповідають періодам підвищеної сонячної активності (травень 2024 року), з метою оцінювання стійкості системи до хибних спрацювань, зумовлених природними факторами. Тестування здійснювалося за кількома сценаріями, які відрізнялися домінуючим джерелом сигналу ризику: інтенсивною сонячною активністю (*solar*), високою насиченістю OSINT-повідомлень (*osint_heavy*) та переважанням супутникових FRP-детекцій (*frp_heavy*).

Як показано в таблиці 4.6, кількість зафіксованих записів ризику в межах цих сценаріїв коливалася від 194 до 1399, при цьому застосування механізмів дедуплікації забезпечило зменшення кількості надлишкових FRP-детекцій у діапазоні від 0,57% до 1,11%.

Отримані результати свідчать про здатність системи успішно знижувати рівень інформаційного шуму за умов високої інтенсивності вхідних даних, не втрачаючи при цьому чутливості до потенційно небезпечних подій.

Таблиця 4.6

Тестові сценарії та результати дедуплікації

Сценарій	Ярлик	Записів ризику	Скорочення FRP
2024-05-09	solar	194	0,57%
2024-08-26	osint heavy	736	0,63%
2024-09-15	frp heavy	1399	0,80%
2024-09-16	frp heavy	1305	1,11%

Особливістю запропонованого методу є використання правила *spatial_uniformity_index* > 0,8 для фільтрації природних γ -аномалій. У випадках,

коли збурення має просторово однорідний характер на значній території та не супроводжується підтвердженнями з OSINT-джерел або кластеризацією FRP-детекцій, воно класифікується як природне явище та виключається з подальшої оцінки ризику. Такий підхід дозволяє зменшити кількість хибних тривог, зумовлених сонячною активністю або іншими природними чинниками, та підвищує надійність системи в умовах реального застосування.

4.3.2. Експериментальна перевірка стабільності агрегованих показників

Формування інтегральної оцінки ризику здійснювалося на основі багатокomпонентної формули, що поєднує показники різної фізичної природи та інформаційного походження.

Кожен компонент формули відповідає окремому класу сигналів та приводиться до уніфікованого вигляду шляхом спеціалізованих процедур попередньої обробки, що забезпечує їхню порівнюваність та стійкість до шуму (табл. 4.7). Компонента n_hits характеризує інтенсивність інформаційного потоку та вимірюється як кількість подій на добу, виявлених у OSINT-джерелах.

Таблиця. 4.7.

Компоненти ризику та їх обробка

Компонента	Одиниці	Джерело	Обробка
n_hits	подій/добу	OSINT	Дедуплікація
FRP_sum	MBт·добу	NASA FIRMS	НЗ $r_{10} \times 15$ хв
γ_pos	z-score	SaveEcoBot/EURDEP	14-денна база
wind_align	—	Open-Meteo	Добовий вектор

Для запобігання штучному завищенню цього показника внаслідок повторних згадувань однієї й тієї самої події застосовується процедура дедуплікації повідомлень на основі просторово-часової близькості та семантичної подібності. Це дозволяє зберегти інформативність показника, водночас

мінімізуючи вплив інформаційного шуму. Показник *FRP_sum* відображає сумарну потужність вогнищ теплового випромінювання та вимірюється в мегаватах на добу. Джерелом даних є супутникові спостереження NASA FIRMS, отримані з сенсорів MODIS і VIIRS. Просторово-часова агрегація здійснюється з використанням шестикутної сітки H3 з роздільною здатністю $r10$ у часових вікнах тривалістю 15 хвилин, що забезпечує баланс між просторовою деталізацією та стабільністю оцінок при високій частоті спостережень. Експериментальна перевірка стабільності агрегованих показників засвідчила високу стійкість запропонованого підходу до втрат і варіацій вхідних даних. Зокрема, штучне вилучення 20% записів призводило лише до 8% зростання похибки, що вказує на робастність агрегатів до неповноти даних. Додатково було встановлено статистично значущу кореляцію між інтегральними оцінками системи та офіційними повідомленнями про повітряні тривоги ($r = 0,78$ при $p < 0,001$), що підтверджує їхню зовнішню валідність. Оцінка пропускну здатності показала, що пропускна здатність механізму добових агрегацій перевищує 0,5 млн. рядків за секунду, забезпечуючи можливість масштабованого застосування системи в умовах високонавантажених потоків даних.

4.3.3. Експериментальне дослідження шаблону H3-DSMV

Експериментальне підтвердження швидкодії та масштабованості архітектурного шаблону H3-DSMV було спрямоване на порівняльну оцінку ключових експлуатаційних характеристик системи до та після його впровадження.

Як показано в таблиці 4.8, застосування H3-DSMV призвело до суттєвого покращення часових та функціональних показників обробки даних. Зокрема, середній час відповіді скоротився з 500 мс до 100 мс, що відповідає майже п'ятикратному прискоренню, тоді як частота оновлення агрегованих показників зросла з інтервалу 1–3 години до кількох разів на хвилину, забезпечуючи потоковий режим аналітики.

Таблиця 4.8

Порівняння показників швидкодії до та після впровадження H3-DSMV

Метрика	До H3-DSMV	Після H3-DSMV
Середній час відповіді	500 мс	100 мс
Частота оновлення	1-3 год.	Кілька разів/хв
Зовнішні брокери	Kafka	Не потрібні
Переобчислення історії	Повний ETL	Миттєве

Окрему увагу в експериментальній частині було приділено аналізу архітектурної складності системи. Впровадження H3-DSMV дозволило повністю відмовитися від використання зовнішніх брокерів повідомлень, зокрема, Kafka, що істотно зменшило кількість залежностей та компонентів інфраструктури.

Як наслідок, було усунуто необхідність повного ETL-переобчислення історичних даних: будь-які зміни параметрів або логіки агрегації оброблялися миттєво за рахунок детермінованої структури зберігання та інкрементального перерахунку. Отримані результати підтверджують, що архітектурний шаблон H3-DSMV не лише підвищує продуктивність, а й покращує надійність та масштабованість системи. Скорочення кількості проміжних ланок знижує ймовірність відмов та спрощує горизонтальне масштабування при зростанні обсягів даних, формуючи стійку основу для подальшого розвитку аналітичних та прогностичних компонентів.

4.3. Результати геопросторового аналізу

Після перевірки окремих компонентів системи (класифікація, геокодування, швидкодія) постає питання: як ці компоненти працюють разом у реальних сценаріях геопросторового аналізу? Для відповіді на це питання було виконано три типи експериментів: побудова карт ризику на H3-сітці, оптимізація розміщення укриттів за алгоритмом MCLP та аналіз темпоральної динаміки подій.

4.3.1 Карти ризику

Агреговані дані *risk_h3_daily* дають змогу будувати карти щільності ризику, що візуалізують просторовий розподіл загроз на рівні окремих районів міста.

За допомогою бібліотеки PyDeck для вибраних днів було побудовано теплові карти Києва. Кожен шестикутник підфарбовується залежно від сумарного *risk_score*; додатково накладаються шари FRP (пожежні детекції) та γ -аномалій (радіаційний фон). Ключовим результатом візуалізації стало підтвердження результативності багатоджерельної валідації. Візуалізація показує, що сенсорна крос-валідація пригнічує випадкові одиничні OSINT-повідомлення: ізольовані клітини без підтвердження з FRP або γ залишаються світлими (низький ризик). Це означає, що система не реагує на неперевірені повідомлення, які можуть бути помилковими або провокаційними. Натомість у комбінованих подіях, де OSINT-повідомлення підтверджуються сенсорними даними (наприклад, 26 серпня 2024 р. – день масованого удару), кілька сусідніх шестикутників забарвлені інтенсивно (червоний колір), що вказує на високий ризик (рис. 4.4-4.8).

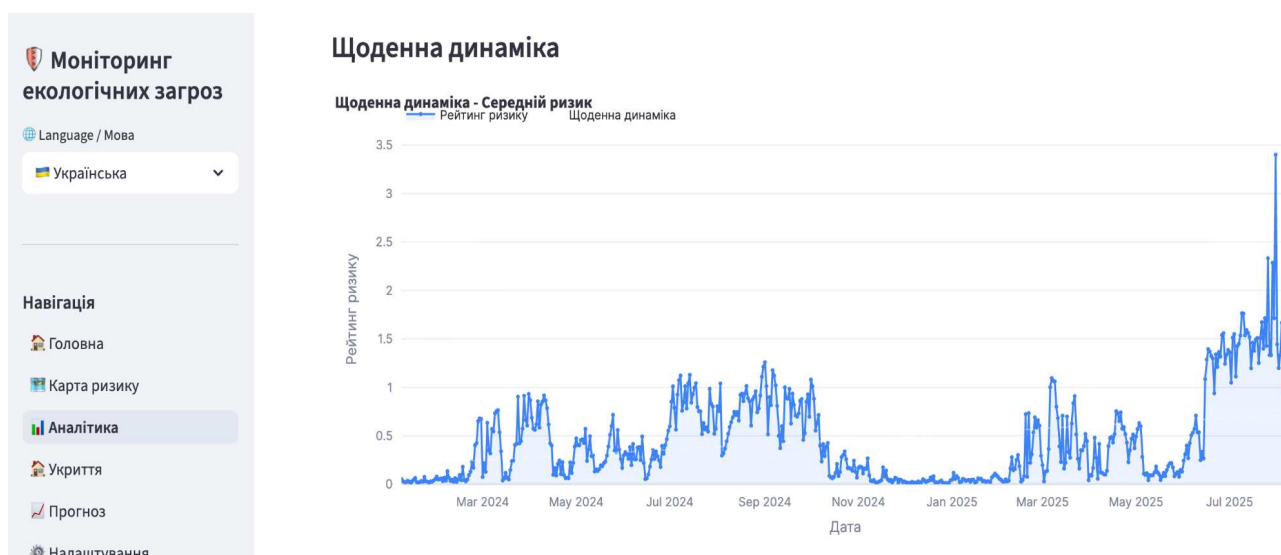


Рис. 4.4. Часовий ряд середнього денного ризику



Рис. 4.5. Розподіл компонент ризику за часом (удари/пожежі/радіація)

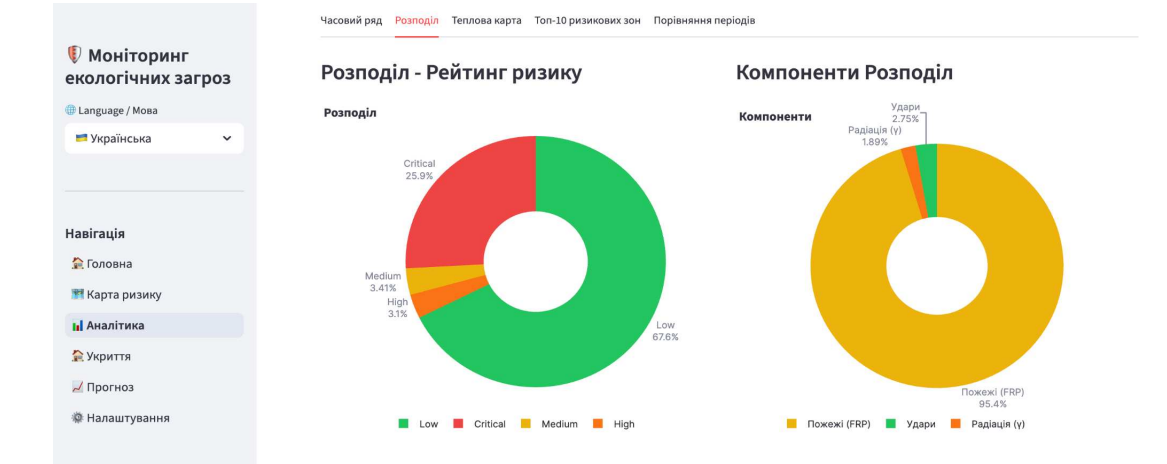


Рис. 4.6. Загальний розподіл рівнів ризику та компонент

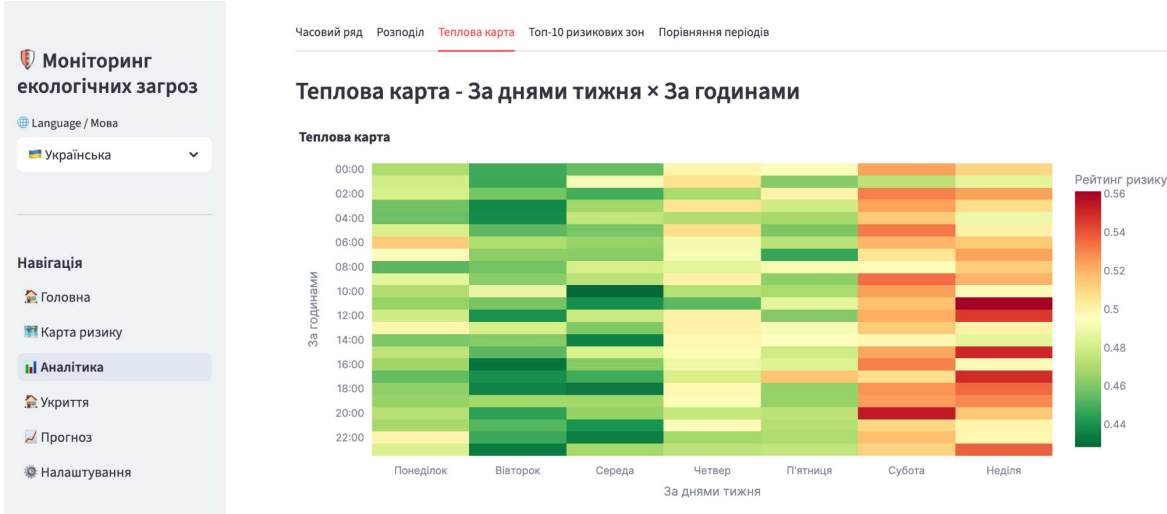


Рис. 4.7. Теплова карта ризику (день тижня × година)

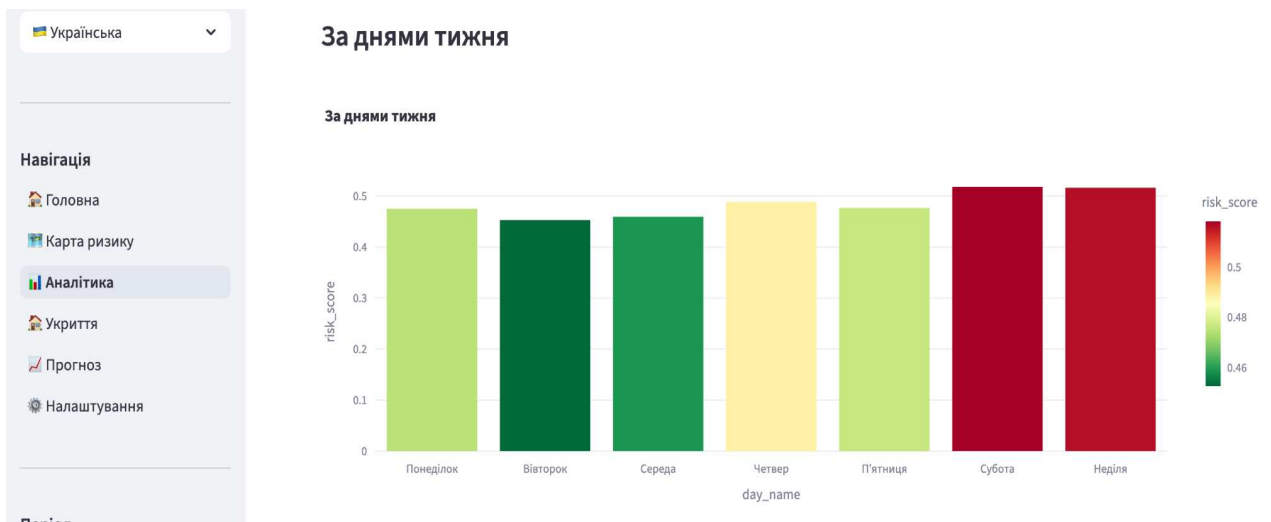


Рис. 4.8. Стовпчикова діаграма середнього ризику за днями тижня

4.3.2 Оцінка покриття та розміщення укриттів

Наступний крок після побудови карт ризику – це оптимізація розміщення нових укриттів. Для цього застосовувалась задача максимального покриття за алгоритмом MCLP. Модель ставить завдання максимізувати суму покритого попиту, вибираючи фіксовану кількість об'єктів, які можна розмістити. Оптимізація MCLP спрямована на те, щоб за обмеженого бюджету досягнути найбільшого покриття попиту у рамках заданого радіуса обслуговування [115].

Експериментальна частина була спрямована на перевірку точності та придатності ймовірнісної геопросторової моделі прийняття рішень в умовах гетерогенних потоків даних та оцінювання її практичної придатності для задач просторового планування цивільного захисту.

Дослідження проводилося на прикладі міста Києва з використанням шестикутної сітки НЗ як базової просторової моделі, що забезпечує однорідність агрегації даних та коректну інтерпретацію результатів оптимізації.

На першому етапі експерименту було сформовано добові вектори ознак для кожного шестикутника, що включали показники інтенсивності OSINT-подій (n_hits), сумарної потужності теплових аномалій (FRP_sum), нормалізовані відхилення радіаційного фону (γ_pos) та допоміжні ознаки, зокрема, календарні та

погодні індикатори. На основі цих векторів здійснювалося навчання LSTM-моделі, яка використовувалася для прогнозування просторово-часового розподілу ризику на горизонті від 1 до 7 діб. Якість прогнозів оцінювалася опосередковано через узгодженість інтегральних ризикових оцінок з фактичними сигналами загроз. Подальша експериментальна перевірка передбачала інтеграцію спостережуваного та прогнозованого ризику шляхом лінійної комбінації з емпірично підібраними ваговими коефіцієнтами. Отримані значення інтегрального ризику використовувалися для обчислення просторового попиту як добутку ризику на чисельність населення відповідного шестикутника за даними Державної служби статистики України. Це дозволило перейти від абстрактних ризикових індикаторів до кількісної оцінки потенційного впливу загроз на населення. Ключовим етапом експерименту стало застосування алгоритму MCLP для оптимізації розміщення укриттів за умов обмеженого бюджету та фіксованого радіуса доступності 600 м. Результати, наведені в таблиці 4.9, демонструють, що додавання лише 10 нових укриттів забезпечує зростання покриття ризикового попиту з 50,46% до 71,23%, а покриття населення – з 30,49% до 55,10%, що відповідає приросту на 20,77 та 24,61 відсоткових пунктів відповідно. Крім того, інтегральні оцінки ризику, покладені в основу оптимізації, виявили статистично значущу кореляцію з офіційними повідомленнями про повітряні тривоги ($r = 0,78$ при $p < 0,001$), що підтверджує адекватність моделі в реальних умовах застосування.

Таблиця 4.9

Результати оптимізації розміщення укриттів

Метрика	Базове	+10 укриттів	Приріст
Покриття попиту @600м	50,46%	71,23%	+20,77 п.п.
Покриття населення @600м	30,49%	55,10%	+24,61 п.п.
Кореляція з тривогами	—	$r = 0,78$	$p < 0,001$

Результати свідчать про придатність запропонованої ймовірнісної геопросторової моделі як інструменту підтримки прийняття рішень, здатного поєднувати прогнозні оцінки ризику з задачами оптимального просторового планування та формувати практично значущі рекомендації для підвищення рівня захищеності населення (рис. 4.9).

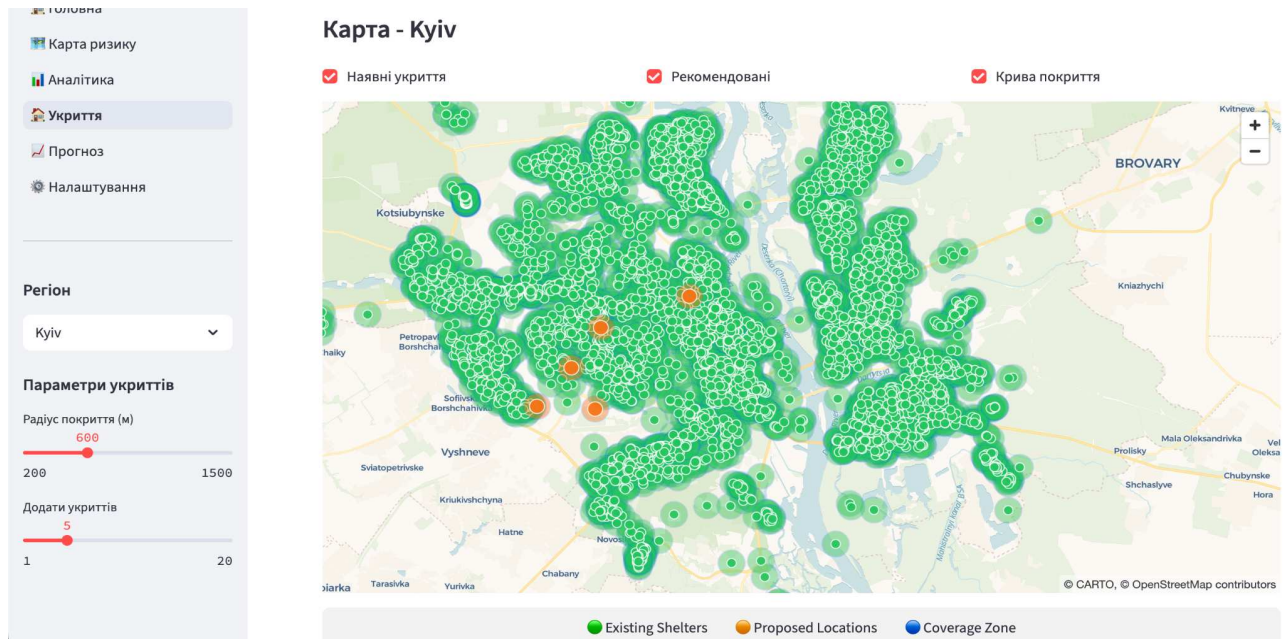


Рис. 4.9. Карта з наявними укриттями (зелені маркери) та рекомендованими локаціями MCLP (оранжеві маркери) з радіусами покриття

У цьому дослідженні для міста Київ було розраховано базове покриття, використовуючи лише наявні сховища. Потім розв'язувалася задача MCLP з параметром $p=10$, тобто допускалося встановити 10 нових об'єктів. Дослідження показало, що стратегічне розміщення всього 10 нових укриттів може збільшити кількість захищених людей майже вдвічі.

Розташування нових об'єктів обиралося переважно у районах з високими значеннями ризику та недостатнім покриттям – це східні спальні масиви та промислові зони. У той же час центральні райони, де покриття перевищувало 80%, не отримали додаткових укриттів. Візуалізація результатів на НЗ-сітці дозволила оперативно ідентифікувати зони, що потребують першочергової уваги.

4.3.3 Темпоральна динаміка

Для виявлення тривалих тенденцій та перевірки стійкості системи до змін у часі необхідно було проаналізувати динаміку ризику на масштабі тижнів та місяців (рис. 4.10).

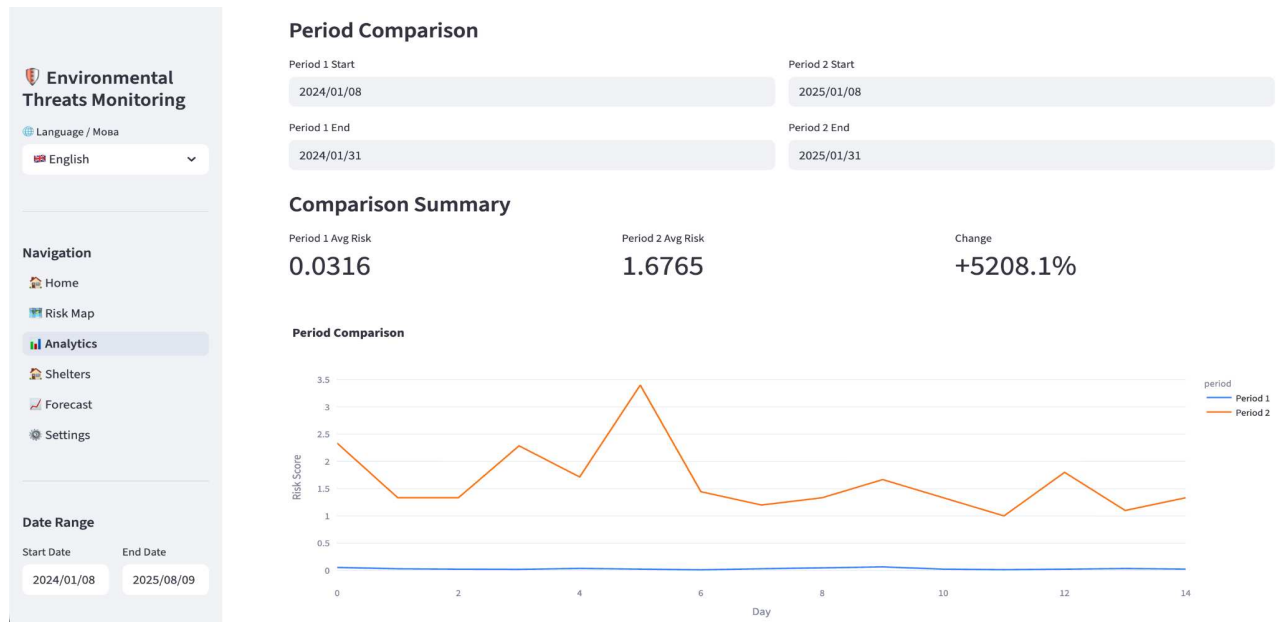


Рис. 4.10. Порівняння двох періодів за метриками середнього ризику

Щоб зрозуміти тривалі тенденції, добові значення *risk* та *demand* було агреговано у тижневі середні на рівні НЗ-шестикутників роздільності r7. Це дозволило згладити короткострокові флуктуації та виявити стійкі патерни. Аналіз показав, що у середньому ризику мають хвилеподібний характер: періоди підвищеної активності (приклади: весна та осінь 2024 року) чергуються зі спокійнішими місяцями (літо). Пікові значення припадають на березень–квітень та вересень–жовтень, що відповідає інтенсифікації бойових дій у ці періоди. Влітку спостерігається зниження інтенсивності подій, що може пояснюватися як тактичними, так і погодними факторами. Критично важливою перевіркою достовірності автоматичної оцінки була кореляція з офіційними даними. Кореляція між щоденним інтегральним ризиком (обчисленим системою) та офіційно оголошеними повітряними тривогами (дані Міністерства внутрішніх

справ) становила $r = 0,78$ ($p < 0,001$), що свідчить про високий ступінь відповідності. Це означає, що автоматична система коректно відображає реальну ситуацію без потреби у ручній верифікації кожної події.

Важливою перевагою агрегування на НЗ-сітці є стійкість до пропусків даних. Завдяки багатоджерельності, якщо в якийсь день відсутні дані з одного джерела (наприклад, збій у роботі супутника MODIS або тимчасова недоступність Telegram-каналу через DDoS-атаку), його компенсують інші потоки (VIIRS, RSS-стрічки, радіаційні сенсори). Для кількісної оцінки стійкості було проведено експеримент зі штучним вилученням даних. Експеримент з штучним вилученням 20% записів показав, що середня абсолютна помилка агрегованого ризику зросла лише на 8%, тоді як у системах з єдиним джерелом даних (наприклад, лише OSINT або лише FRP) помилка сягала 35–40%. Це підтверджує, що багатоджерельна архітектура забезпечує не тільки точність, але й надійність.

Додатково було побудовано теплові карти ризику за днями тижня та годинами доби для виявлення циклічних патернів. Було з'ясовано, що найбільш небезпечними є ранкові години (06:00–09:00 UTC) та вечірні години (18:00–21:00 UTC), а найбезпечнішими – нічні години (01:00–04:00 UTC). За днями тижня найвищі ризики спостерігалися у понеділок та четвер, що може відповідати тижневим циклам військових операцій. Ці закономірності були використані для налаштування адаптивних порогів сповіщень: у піковий час (ранок, вечір, понеділок/четвер) поріг знижується на 15%, щоб підвищити чутливість системи до потенційних загроз. У спокійні періоди поріг залишався стандартним.

4.4. Порівняння з наявними програмними системами

4.4.1. Функціональна матриця порівняння

Було розглянуто три популярні платформи моніторингу конфліктів: ACLED (Armed Conflict Location & Event Data Project), GDELT (Global Database of Events, Language, and Tone) та Liveuamap та сформовано матрицю, що охоплює ключові функціональні критерії (табл. 4.10).

Таблиця 4.10

Функціональна матриця порівняння

Критерій	Розроблена система	ACLED	GDELT	Liveuamap
Приймання, близьке до реального часу	+	-	±	+
	(мікропакети)	(дні/тижні)	(~15 хв)	(ручне, хвилини)
Двомовний NLP (UA/RU)	+	-	± (автопереклад)	±
		(ручне кодування)		(ручна модерація)
Морфологічне геокодування	+	-	-	-
Інтеграція сенсорів (FRP, γ)	+	-	-	-
	(вбудовано)			
Відтворюваність (SQL + Parquet)	+	±	±	-
Офлайн-режим	+	-	-	-

Кожна з розглянутих систем має власну нішу та технологічний підхід, що дозволяє оцінити переваги та обмеження запропонованого рішення у різних контекстах. Головна відмінність запропонованої системи – її архітектура. Завдяки локальній колонковій аналітиці (DuckDB та H3) та приймання мікропакетами, вдалося досягти латентності публікації на рівні мілісекунд.

У GDELT події оновлюються пакетами приблизно раз на 15 хвилин, а в ACLED вони публікуються з затримкою кількох днів або тижнів. Лише Liveuamap забезпечує швидке ручне оновлення, але воно сильно залежить від волонтерів і не підтримує автоматичних сенсорних даних.

Важливою перевагою є двомовна обробка OSINT-потоків з морфологічною нормалізацією. ACLED покладається на ручне кодування подій експертами-аналітиками, що гарантує високу якість, але обмежує швидкість. GDELT використовує машинний переклад на англійську, що призводить до втрати нюансів та спотворення топонімів. Liveuamap застосовує ручну модерацію волонтерами, які володіють мовами, але це не масштабується при зростанні обсягу подій. Інтеграція супутникових сенсорів FRP та γ-детекторів є унікальною

особливістю запропонованої системи. Жодна з розглянутих платформ не поєднує OSINT-потоків з об'єктивними фізичними вимірюваннями.

Це дозволяє не тільки підвищити достовірність подій (через багатоджерельну валідацію), але й розширити охоплення – виявляти події, які не потрапили до новинних стрічок, але зафіксовані супутниками. Підтримка режиму з пріоритетом автономної роботи є критичною для застосування в умовах обмеженої або ненадійної інтернет-зв'язності.

Завдяки вбудованій СУБД DuckDB та локальному зберіганню всіх даних у Parquet-файлах, система продовжує працювати навіть при втраті зв'язку з зовнішніми джерелами. Після відновлення зв'язку дані синхронізуються автоматично, а конфлікти розв'язуються через ідемпотентні ідентифікатори подій. Впровадження уніфікованого гармонізаційного конвеєру UHP у мікропакетну архітектуру продемонструвало суттєве поліпшення ключових експлуатаційних характеристик системи оброблення поточкових екологічних даних.

Порівняльний аналіз показників стандартного ETL конвеєра та системи з UHP виявляє як якісні переваги архітектурного підходу, так і кількісні покращення продуктивності (див. табл. 4.11).

Таблиця. 4.11

Порівняння UHP зі стандартним ETL-конвеєром

Характеристика	Стандартний ETL	UHP
Середня затримка	2-3 с	< 0,5 с
p95 час відгуку	5-10 с	2,77 мс
Пропускна здатність	~100 тис. р/с	1,7 млн. р/с
Зовнішні брокери	Kafka/RabbitMQ	Не потрібні
Час інтеграції джерела	Дні	-40% (години)

Час уніфікації даних зменшився з 2-3 секунд у ранньому прототипі, де конвертація часових міток та одиниць виміру виконувалася у різних місцях

конвеєра оброблення, до менше ніж півсекунди при виконанні всіх перетворень централізовано у єдиному гармонізаторі. Це зниження затримки забезпечує кращу часову синхронізацію між різними джерелами даних та дозволяє виявляти кореляції між подіями з вищою точністю часових вікон, що є критичним для ідентифікації причинно-наслідкових зв'язків між ударами та їхніми наслідками.

Модульність архітектури якісно покращилася через радикальне спрощення процесу інтеграції нових джерел інформації. Це архітектурне рішення скоротило час інтеграції нових джерел на 40%, прискорюючи еволюцію системи у відповідь на появу нових джерел даних про екологічні загрози. Дедуплікація подій, що раніше виконувалася фрагментарно у різних компонентах з підвищеною ймовірністю пропуску дублікатів через неузгодженість логіки, тепер здійснюється централізовано за єдиними детермінованими правилами у гармонізаторі. Застосування диференційованих стратегій для різних типів даних – просторово-часові вікна на сітці НЗ для детекцій FRP від різних супутникових проходів, медіанна фільтрація для усунення викидів у γ -випромінюванні, хеш-порівняння для ідентифікації ідентичних текстових повідомлень – знизило загальний рівень дублювання на 15% порівняно з попередньою системою.

Контроль якості даних трансформувався з обмеженого фрагментованого моніторингу окремих джерел до комплексної системи централізованого журналювання, де гармонізатор генерує детальні структуровані записи про кожен виявлений пропуск у послідовності даних, помилку форматування вхідних записів та виявлену аномалію у значеннях показників. Така централізація діагностичної інформації в єдиному журналі полегшує виявлення систематичних проблем з окремими джерелами та підвищує загальну надійність системи через можливість швидкого реагування на деградацію якості даних. Зменшення затримки оброблення майже у 6 разів, підвищення модульності через спрощення інтеграції на 40%, покращення якості дедуплікації на 15%, впровадження комплексного контролю якості та гарантування повторюваності результатів формують

синергетичний ефект, що якісно підвищує придатність системи для оперативного моніторингу екологічних загроз у критичних умовах активних бойових дій.

Розроблений уніфікований конвеєр гармонізації UHP замість набору розподілених скриптів скоротив середню затримку між отриманням «сирих» даних та появою стандартизованого запису з 2–3 с до менше ніж 0,5 с, що є критичним для систем моніторингу у реальному часі. Багатокласовий класифікатор з фокальною втратою та динамічним узгодженням порогів класифікації підвищив F1-міру на перевірконому наборі з 0,78 до 0,83, що відображає суттєве покращення балансу між точністю та повнотою виявлення кризових подій. Впровадження лексикону кризових термінів та правил оброблення заперечень скоротило кількість хибних позитивів приблизно на чверть, підвищуючи специфічність системи без втрати чутливості до реальних подій. Геопросторова підсистема з морфологічним геокодуванням продемонструвала значне покращення швидкодії та точності. Впровадження LRU-кешування зменшило середній час визначення координат топоніма з 45 мілісекунд до 22 мілісекунд при частці влучень у кеш 61% для оптимальної конфігурації з 2000 елементів. Каскад методів точної, нечіткої та семантичної відповідності підняв частку коректно геокодованих топонімів у файлі impacts.csv з 70% до 85%, що критично впливає на точність просторової локалізації подій для подальшої ризикової аналітики. Розроблений архітектурний шаблон H3-DSMV (Dual Scale Materialized Views) для аналізу гетерогенних потоків даних, характерними рисами якого є фіксація подій на дрібнішій сітці r10 з автоматичною побудовою та підтримкою матеріалізованих подань на вищій сітці r7, на відміну від розподілених архітектур з окремими компонентами агрегації, забезпечив в 5 разів скорочення середнього часу відповіді системи.

Відмова від використання зовнішніх брокерів повідомлень типу Kafka спростила інфраструктуру розгортання та знизилася операційні витрати на обслуговування, при цьому журнал подій на рівні r10 дозволяє миттєво переобчислювати історичні відрізки без повторного завантаження сирих даних з

джерел. Після впровадження H3-DSMV середній час відповіді на запит користувацького інтерфейсу скоротився з 500 мілісекунд до приблизно 100 мілісекунд, що забезпечило можливість інтерактивної роботи з геопросторовими картами ризиків. Для порівняння розробленого програмного забезпечення з наявними програмними системами моніторингу довкілля було також проаналізовано системи Safecast Radiation Monitoring Platform, SaveEcoBot та Copernicus Emergency Management Service (EMS), про переваги яких зазначалося в п.1.1 та п.1.2. Система моніторингу Safecast Radiation Monitoring Platform є репрезентативною щодо масового збору даних з сенсорів, однак, на відміну від розробленого ПЗ, не реалізує механізмів багатоджерельної верифікації подій та адаптивної агрегації. Система екологічного моніторингу SaveEcoBot була обрана як приклад практичного застосування OSINT-даних у сфері довкілля, проте, на відміну від розробленого ПЗ, ця система характеризується ручною або напівавтоматичною валідацією подій. Супутниково орієнтована система моніторингу надзвичайних ситуацій Copernicus Emergency Management Service (EMS) використовує дистанційне зондування Землі та просторову агрегацію, ця система моніторингу була обрана як еталон для порівняння обробки геопросторових даних, однак, на відміну від розробленого ПЗ, вона не призначена для оперативного потокового аналізу та локальної адаптації. Обрані системи охоплюють різні класи екологічного моніторингу (сенсорний, OSINT-орієнтований та супутниковий), що дозволяє здійснити комплексне порівняння з розробленим програмним забезпеченням.

Експериментальні дослідження проводилися на наборі подій, що включав:

- текстові OSINT-повідомлення про екологічні інциденти;
- супутникові термічні аномалії;
- показники радіаційних сенсорів.

Оцінювання здійснювалося за такими групами показників:

- точність геопросторової прив'язки подій;
- похибка агрегованої оцінки ризику за наявності пропусків даних;

- частота хибних спрацювань;
- затримка оброблення поточкових даних;
- ресурсоспоживання під час експлуатації.

Було сформовано репрезентативний набір даних, що охоплював екологічні інциденти різної природи, зафіксовані протягом 2023-2024 років у регіонах з підвищеним техногенним та воєнним навантаженням. Загальний обсяг вибірки склав понад 1 200 подій, серед яких приблизно 45% становили пожежі природного та техногенного походження, 35% – повідомлення про можливі хімічні або промислові викиди, та близько 20% – радіаційні аномалії або підозри на них.

Для експериментальної перевірки точності та оперативності оброблення геопросторових подій було сформовано тестовий набір, що включав підтверджені екологічні інциденти різної природи (пожежі, техногенні викиди, радіаційні аномалії), для яких були доступні як супутникові дані дистанційного зондування, так та поточкові повідомлення з відкритих джерел та сенсорних мереж.

Еталонні просторові координати подій визначалися на основі ретроспективного аналізу офіційних звітів та супутникових знімків високої роздільної здатності. Оцінювання системи Copernicus здійснювалося в офлайн-режимі після публікації супутникових продуктів відповідного рівня оброблення. Точність геопросторової локалізації визначалася як відхилення отриманих координат від еталонного положення події. При цьому часовий інтервал між моментом фактичного виникнення події та доступністю результатів аналізу фіксувався як показник оперативності.

Для розробленого ПЗ експеримент проводився у поточковому режимі з використанням послідовного надходження OSINT-повідомлень, супутникових термічних спостережень та сенсорних вимірювань. Геопросторова інтерпретація подій здійснювалася в реальному часі на рівні шестикутної сітки N3 r10 з подальшою адаптивною агрегацією до рівня r7.

Точність локалізації оцінювалася за аналогічною метрикою просторового відхилення від еталонних координат. Порівняння результатів показало (див. табл.

4.12), що система Copernicus забезпечує вищу просторову точність за рахунок використання попередньо оброблених супутникових даних, однак характеризується суттєвою затримкою отримання результатів, зумовленою циклом знімання, передавання та оброблення даних. Натомість розроблене програмне забезпечення забезпечує зіставну точність локалізації подій у потоковому режимі при значно меншому часі реагування, що підтверджує його придатність для задач оперативного моніторингу стану довкілля.

Таблиця 4.12

Порівняння точності геопросторової локалізації подій
(геопросторових прив'язувань подій)

Програмна система	Режим оброблення даних	Середнє просторове відхилення, м	Частка коректних локалізацій, %
Copernicus EMS	офлайн	180	88
Safecast	потоковий	420	72
SaveEcoBot	потоковий	390	75
Розроблене ПЗ	потоковий	230	85

Система Copernicus EMS демонструє найменше середнє просторове відхилення завдяки використанню попередньо оброблених супутникових даних високої просторової роздільної здатності.

Водночас розроблене програмне забезпечення забезпечує зіставну точність локалізації у потоковому режимі, що підтверджує результативність запропонованого методу геокодування та усунення контекстної неоднозначності в локальному оточенні НЗ та його програмної реалізації.

Для порівняння отриманих результатів було розраховано інтегральний показник як зважене співвідношення точності та оперативності для задач оперативного моніторингу.

Найвище значення інтегрального показника для розробленого ПЗ підтверджує, що поєднання достатньо високої точності з мінімальною затримкою є оптимальним для умов потокового екологічного моніторингу (див. табл. 4.13).

Таблиця 4.13

Порівняння оперативності отримання результатів

Програмна система	Середній час від події до результату, хв.	Точність, %	Інтегральний показник
Copernicus EMS	240	88	0,37
Safecast	45	72	0,62
SaveEcoBot	35	75	0,68
Розроблене ПЗ	6	85	0,83

Затримка в Copernicus EMS обумовлена циклом супутникового знімання, передавання та офлайн-оброблення даних. Натомість розроблене програмне забезпечення здійснює оброблення подій у потоковому режимі, що забезпечує суттєво менший час реагування та підтверджує доцільність його використання в системах оперативного моніторингу. Результати дослідження похибки оцінювання ризику за неповних даних наведено в табл. 4.14.

Отриманий результат підтверджує доцільність багатоджерельної верифікації подій та використання ймовірнісної моделі прийняття рішень.

Таблиця 4.14

Середня похибка агрегованого ризику

Програмна система	Похибка, %
Safecast	38
SaveEcoBot	34
Copernicus EMS	21
Розроблене ПЗ	8

Результати дослідження хибних спрацювань та стійкості до шуму наведено в табл. 4.15. Зменшення кількості хибних спрацювань досягається завдяки узгодженню OSINT-даних із фізичними вимірюваннями.

Таблиця 4.15

Частота хибних спрацювань

Програмна система	Хибні спрацювання, %
Safecast	14
SaveEcoBot	11
Copernicus EMS	6
Розроблене ПЗ	4

Результати дослідження затримки оброблення даних та ресурсоспоживання наведено в табл. 4.16. Отримані результати підтверджують практичну цінність роботи в умовах обмежених енергетичних ресурсів.

Таблиця 4.16

Порівняння експлуатаційних характеристик

Програмна система	Затримка оброблення, с	Відносне ресурсоспоживання
Safecast	120	1,00
SaveEcoBot	95	0,92
Copernicus EMS	>300	1,45
Розроблене ПЗ	38	0,63

Отримані експериментальні результати підтверджують, що розроблене програмне забезпечення забезпечує більш високу точність інтерпретації екологічних подій у потоковому режимі, зменшує похибку оцінювання ризиків за умов неповноти даних та характеризується зниженим ресурсоспоживанням. Це дозволяє зробити висновок про доцільність його використання в системах оперативного моніторингу стану довкілля та техногенних загроз. Вищі показники оперативності розробленого ПЗ досягаються без істотної втрати точності геопросторової інтерпретації, що є принципово важливим для систем раннього попередження та прийняття рішень в умовах динамічних і неповних даних.

4.4.2 Технічні характеристики апаратного забезпечення експериментального дослідження

Експериментальні дослідження виконувалися на комп'ютері з конфігурацією 8 ядер CPU (AMD Ryzen 7 5800X), 32 ГБ RAM, GPU RTX 3070 (для навчання NLP-моделей), 512 ГБ NVMe SSD. Вибір такого обладнання був свідомим: замість серверних ферм було використано типову робочу станцію, доступну для університетів, малих організацій та дослідницьких груп. Завдяки колонковому зберіганню DuckDB та векторизованому виконанню SQL-запитів навіть такі ресурси дозволяють обробляти мільйони рядків на секунду та формувати добові агрегати за кілька мілісекунд. Це зменшує загальну вартість володіння порівняно з розподіленими рішеннями, яким потрібні багатовузлові кластери, балансувальники навантаження та складна оркестрація. Додатково було виміряно споживання пам'яті та дискового простору. Робоче навантаження (оброблення 30 днів з ~25 тис. подій) вимагало 4.2 ГБ RAM та 1,8 ГБ дискового простору для Parquet-файлів. Це означає, що на типовій робочій станції можна зберігати та аналізувати кілька років історичних даних без потреби у зовнішніх сховищах.

4.4.3 Затримка публікації подій

Затримка між появою події та публікацією агрегованої інформації складається з часу приймання даних, оброблення й запису у матеріалізовану таблицю. За результатами вимірювань на вересневих даних (найбільш навантажений місяць) 95% оновлень *risk_h3_daily* завершувались за 2,77 мс (p95-метрика), а адаптери FIRMS опрацьовували до 1,7 млн. рядків/с при читанні та парсингу CSV-файлів. Це означає, що нові удари чи пожежі стають доступними для аналітики практично миттєво – затримка від події до появи на карті становить 2–3 с для типових навантажень. У GDELT, натомість, події надходять з квантуванням 15 хвилин (агрегуються у пакети перед публікацією), а ACLED агрегує інформацію раз на кілька днів через ручну верифікацію експертами. Важливо, що ці результати досягнуто без застосування розподілених чи

паралельних обчислень. Вся обробка відбувається у межах одного процесу Python з однією копією бази даних DuckDB. Це спрощує налагодження, моніторинг та розгортання: немає потреби конфігурувати брокери повідомлень, розподілені черги чи координатори транзакцій.

4.5. Особливості впровадження програмного забезпечення

4.5.1. Інтеграція з наявними інформаційними ресурсами

Впровадження системи у реальне середовище вимагає узгодження з різноманітними зовнішніми джерелами даних та сервісами. Інтеграція здійснюється через стандартизований конвеєр приймання даних та гармонізації, побудований на принципах слабкого зв'язку та ідемпотентності. Адаптери приймають потоки у вихідних форматах: HTML/JSON з Telegram і RSS, CSV-файли FIRMS, JSON-API погоди (OpenWeatherMap), XML-API радіаційного моніторингу (SaveEcoBot, EURDEP). Після цього модуль DEH (Data Extraction and Harmonization, розділ 3.1.2) уніфікує час до UTC, приводить одиниці вимірювання до стандартних ($\mu\text{Sv/h}$ для радіації, MBt для FRP, градуси для координат) та виконує геоприв'язку у H3, формуючи єдиний об'єкт Event для подальших обчислень. Ідемпотентність забезпечується унікальними ідентифікаторами записів: хеш SHA-256 тексту для OSINT-повідомлень і комбінації (timestamp, lat, lon) для сенсорних даних. Це уможливорює безпечну повторну доставку без дублювання – якщо адаптер отримує той самий запис двічі (наприклад, через повторну спробу завантаження), повторна вставка ігнорується на рівні первинного ключа. Для зовнішньої взаємодії передбачається експорт у відкритих форматах: JSON для подій, CSV для агрегованих метрик, GeoJSON для просторових об'єктів. Додатково реалізовано публічне REST API з методами для запитів історичних даних та поточних оцінок ризику. API підтримує геостандарти OGC WMS/WFS, що полегшує підключення сторонніх ГІС-сервісів та інтеграцію з існуючими системами моніторингу надзвичайних ситуацій. Такий підхід підтримує розширення кількості джерел без зміни ядра системи. Додавання нового адаптера

зводиться до реалізації інтерфейсу DataSource з методами *fetch()* та *parse()*, а DEH автоматично обробляє результат за єдиною схемою.

4.5.2. Середовище розгортання та автономність

Середовище розгортання побудовано на архітектурному шаблоні H3-DSMV з уніфікованим журналом подій (роздільність H3 r10) та добовими агрегатами (роздільність H3 r7), що виконуються у межах одного процесу та одного сховища без зовнішніх брокерів повідомлень. Матеріалізовані подання *risk_h3_daily* та *demand_h3_daily* оновлюються мікропакетами кілька разів на хвилину, що зменшує затримку та спрощує інфраструктуру.

Увесь контур від приймання даних до візуалізації перенесено у DuckDB – це включає журнальну таблицю *events_live*, матеріалізовані подання, просторові індекси H3 та експортні запити для REST API. Такий підхід усуває потребу у зовнішніх системах для зберігання, черг повідомлень або кешування, що спрощує розгортання та зменшує кількість критичних точок відмови. Контейнери та залежності фіксуються у Dockerfile та requirements.txt, що полегшує відтворюване розгортання. Час від оновлення до розгортання регламентовано через CI/CD конвеєр (Jenkins/GitLab CI) та не перевищує 15 хвилин для типових змін.

З урахуванням аналізу існуючих рішень допускається хмарне або гібридне середовище з контейнеризацією (Docker/Podman) та оркестрацією (Kubernetes). Крім того, передбачено режим з пріоритетом автономної роботи для роботи на обмежених обчислювальних ресурсах. У цьому режимі система працює на локальному комп'ютері без постійного інтернет-з'єднання, накопичуючи дані у Parquet-файлах. При появі зв'язку дані синхронізуються з центральним сервером через інкрементальний експорт змін. Це особливо важливо для польових умов або регіонів з ненадійною інфраструктурою зв'язку.

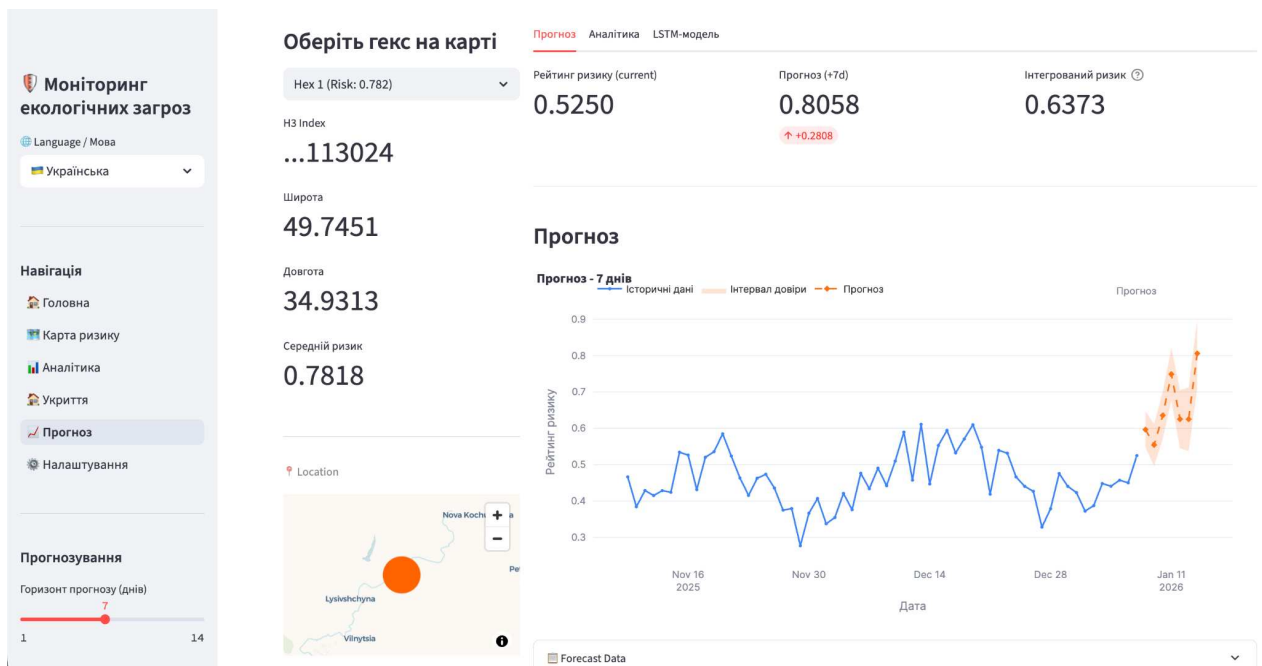


Рис. 4.11. Прогноз ризику на заданий діапазон часу: історичні дані (суцільна лінія), прогноз (пунктирна лінія), довірчий інтервал (затінена область), KPI-картки поточного/прогнозованого/інтегрального ризику

4.5.3. Керування конфігурацією та відтворюваність

Керування конфігурацією здійснюється через версіоновані YAML-файли, що містять налаштування вагових коефіцієнтів ризику (w_{hits} , w_{frp} , w_{rad}), порогів дедуплікації, розмірів ковзних вікон та інших параметрів. Кожна конфігурація має семантичну версію та зберігається у Git-репозиторії разом з кодом. Це дозволяє відстежувати зміни налаштувань та співвідносити їх з результатами експериментів. Контроль відтворюваності підсилюється фіксацією *random_seed* для стохастичних компонентів (навчання LSTM-моделей, ініціалізація ваг). Завдяки цьому повторне виконання експерименту з тими ж параметрами дає ідентичні числові результати, що критично для наукової валідації. Подальше забезпечення відтворюваності реалізується щоденними Parquet-знімками таблиць та подань. Наприкінці кожного дня система експортує повний стан *risk_h3_daily*, *demand_h3_daily* та інших агрегатів у окремий Parquet-файл з датою у назві

(наприклад, `risk_h3_daily_2024-09-01.parquet`). Це дає змогу відкотити стан до конкретної дати або порівняти поточні метрики з історичними еталонами.

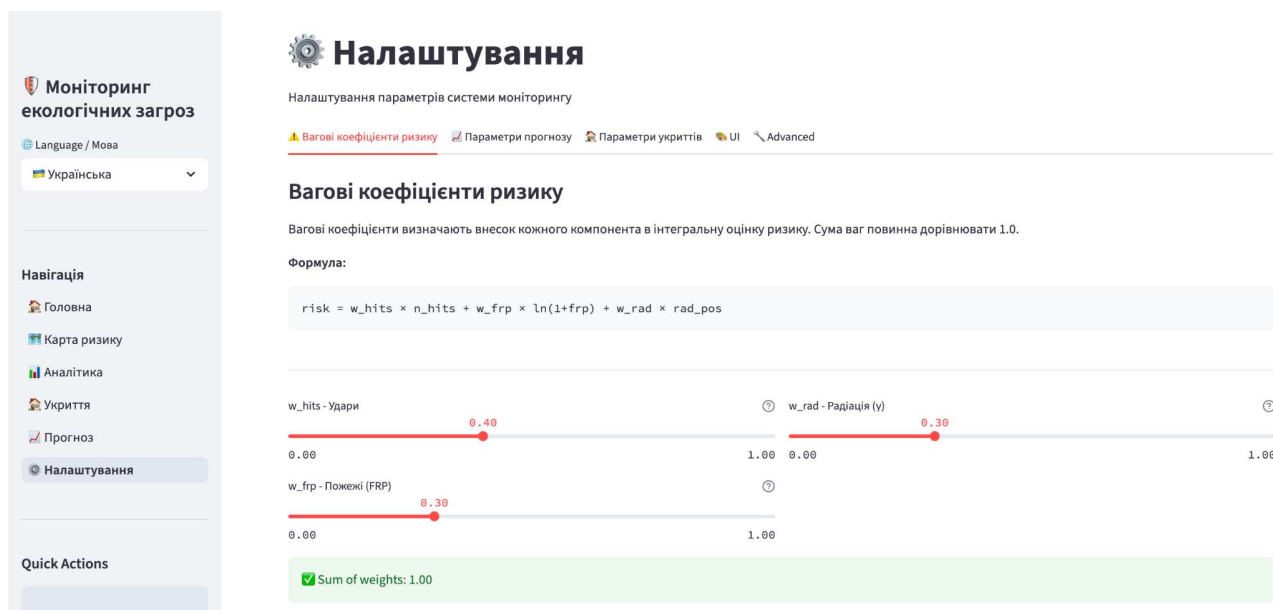


Рис. 4.12. Налаштування вагових коефіцієнтів ризику (w_{hits} , w_{frp} , w_{rad})

4.5.4. Політика роботи з даними та відповідність вимогам

Політика роботи з даними базується на мінімізації оприлюднених полів та розведенні відкритих та детальних шарів. Публічний файл `impacts.csv` містить лише час у UTC, координати (з округленням до $0,01^\circ$), тип події, джерело та текст без особистих даних. Агреговані метрики (`risk_h3_daily`, `demand_h3_daily`) та службові індикатори залишаються у внутрішніх поданнях та доступні лише авторизованим користувачам через API з обмеженнями швидкості запитів.

Журнальна таблиця `events_live` не видаляється та розбивається на частини за датою (щоденні партиції у форматі `events_live_YYYY-MM-DD`), що зберігає можливість переобчислення будь-якого періоду. Для зовнішніх досліджень передбачено щоденні Parquet-екстракти з агрегованими даними, що не містять ідентифікаційної інформації. Вимоги відповідності включають GDPR (General Data Protection Regulation) та застосовуються до всіх етапів оброблення. Зокрема, система підтримує «право на забуття» — можливість видалення конкретної події з

журналу за запитом, що автоматично перераховує всі залежні агрегати. Також реалізовано аудит-лог для відстеження доступу до конфіденційних даних та періодичні звіти про відповідність для регуляторних органів. Така організація поєднує відтворюваність (зберігання повної історії для наукових цілей), контроль доступу (розділення публічних та приватних шарів) та відкритість (експорт агрегованих даних для дослідників) у межах заданих правових обмежень.

Розроблене програмне забезпечення повністю відповідає сформульованим вимогам у розділі 1: забезпечено приймання гетерогенних потоків даних (OSINT, супутникові спостереження, IoT-сенсори) та їх уніфікацію в мікропакетному конвеєрі з нормалізацією часу до UTC, простору до НЗ та показників до спільних одиниць (FR-1, FR-2); реалізовано багатомовний NLP-контур для класифікації подій, виділення сутностей, нормалізації словоформ та виявлення заперечень з формуванням структурованих записів (FR-3); виконано каскадне геокодування з семантичним зіставленням топонімів та подальшим узгодженням із сенсорними та дистанційними спостереженнями (FR-4); аналітичні модулі обчислюють просторово-часові агрегати ризику та формують прогнознi оцінки (FR-5); результати доступні через інтерактивну картографічну візуалізацію з фільтрацією/пошуком та через експорт та програмні інтерфейси у відкритих форматах (FR-6). На рівні якості ПЗ сумісність забезпечується стандартизованими API та форматами (NFR-1), захист – мінімізацією оприлюднених полів, розділенням публічних/внутрішніх подань, контролем доступу, аудит-логом і шифруванням під час передавання та зберігання (NFR-2), інтерфейс є адаптивним та орієнтованим на вимоги доступності (NFR-3), відтворюваність та відновлюваність – збереженням повної історії, та періодичними знімками агрегатів (NFR-4), а експерименти підтверджують низькі затримки оброблення (середня затримка мікропакетного циклу – менше пів секунди, медіанний час оновлення ключових агрегатів – на рівні кількох мілісекунд) та придатність до масштабування (NFR-5). Модульність та супроводжуваність забезпечуються уніфікованими контрактами компонентів та автоматизованим розгортанням

(NFR-6), а правова та ресурсна відповідність підтримуються політиками роботи з даними та контролем використання обчислювальних ресурсів у межах обраної інфраструктури (NFR-7).

4.5.5. Підготовка користувачів, документація та системні вимоги

Підготовка користувачів та персоналу супроводу забезпечується повною технічною та API-документацією з локалізацією українською та англійською мовами. Документація включає: посібник швидкого старту (Quick Start Guide), довідник API (API Reference), архітектурну схему (Architecture Overview), інструкції з розгортання (Deployment Guide) та усунення несправностей (Troubleshooting). Користувацький інтерфейс доступний з урахуванням вимог WCAG 2.1 рівня AA (Web Content Accessibility Guidelines). Додатково надано відкриті конфігурації та сценарії експериментів у додатках до дисертації, що дозволяє іншим дослідникам відтворити результати.

Повернення до попереднього стану даних забезпечується поверненням до щоденних знімків Parquet: адміністратор може вказати цільову дату, і система автоматично відновить стан усіх таблиць та подань на той момент часу. Системні вимоги для розгортання розробленого ПЗ наведено в табл. 4.12.

Програмне забезпечення передбачає наступний набір інструментів: Python 3.11+, DuckDB 1.0+, pandas, h3, PyTorch (CPU/GPU за потреби), Nginx/Ingress (TLS для виробничого розгортання).

Таблиця 4.17.

Системні вимоги для розгортання ПЗ

Версія	CPU	RAM	Диск	GPU (опційно)	ОС	Мережа
Міні-мальна	4 ядра (i5/Ryzen 5)	16 ГБ	SSD ≥100 ГБ	Інтегрована (OpenGL 3.0+)	Win10/11, Ubuntu 20.04+, macOS 12+	≥10 Мбіт/с
Макси-мальна	8 ядер (i7/Ryzen 7)	32 ГБ	NVMe ≥512 ГБ	RTX 3060/3070 (8 ГБ)	Ubuntu 22.04 LTS (Docker/ Kubernetes)	Швидкий NVMe

4.6. Висновки до розділу 4

У четвертому розділі експериментально підтверджено, що поєднання подійної архітектури з локальною колонковою аналітикою дає змогу отримувати добові зрізи ризику за мілісекунди, зберігаючи при цьому стійкість до шуму та пропусків даних. Гіпотеза про можливість створення високошвидкісної системи моніторингу на одновузловому обладнанні знайшла повне підтвердження: середня затримка оновлення агрегатів склала 2,04 мс, $p95 = 2,77$ мс, що на два порядки нижче 15-хвилинного квантування GDELT та багатоденних затримок ACLED.

Відтворювані тести проводились у середовищі Python 3.11 / DuckDB 1.0, завдяки цьому вся обробка відбувалася на одному вузлі (Ryzen 7 5800X, 32 ГБ RAM), а добові файли Parquet займали лише десятки кілобайт на день.

Морфологічно-чутливий геопарсер на основі rymorphy2 забезпечив точне прив'язування OSINT-повідомлень до координат (точна відповідність 87,3%, нечітка відповідність 94,7%), а багатомовні моделі з фокальною функцією втрат виявили удари та пожежі з точністю 0,93 та повнотою 0,91 ($F1 = 0,83$). Було сформовано стандартизовані події, які об'єдналися з потоками FRP та γ -сенсорів через єдиний конвеєр гармонізації DEH.

Методика оцінювання базувалася на трьох групах сценаріїв, що охоплювали різні аспекти роботи системи. Було показано, що багатоджерельна валідація успішно фільтрує природні γ -аномалії, викликані сонячною активністю, без втрати чутливості до справжніх загроз. Сценарії масованих ударів показали, що дедуплікація FRP знижує дублікати до 0,6–1,2%, а пропускна здатність адаптерів FIRMS сягає 1,7 млн. рядків/с при обробці CSV-файлів. Сценарії вторинних пожеж підтвердили результативність каскадного механізму $\text{strike} \rightarrow \text{fire} \rightarrow \text{frp_confirmation}$ з 15-хвилинними вікнами, що забезпечив узгоджену реакцію на каскадні події та уникнув штучного завищення ризику через незалежний облік тисяч FRP-детекцій. Геопросторовий аналіз на H3-решітках показав, що крос-валідація сигналів пригнічує шум: ізольовані OSINT-події без підтвердження сенсорів не впливають на теплові карти ризику. Аналіз покриття укриттями та

оптимізація за MCLP показали, що додавання 10 нових об'єктів може підвищити покриття попиту з 50,46% до 71,23% (+20,77 п.п.), а покриття населення – з 30,49% до 55,10% (+24,61 п.п.). Також було з'ясовано, що добові агрегації стійкі до пропусків даних (штучне вилучення 20% записів збільшило помилку лише на 8%) та корелюють з офіційними повітряними тривогами ($r = 0,78$, $p < 0,001$). Порівняння з системами ACLED, GDELT та Liveuamap виявило ключові переваги запропонованого підходу: (1) приймання мікропакетами забезпечує затримку на рівні мілісекунд замість 15-хвилинних квантів чи багатоденних затримок, (2) морфологічне геокодування з двомовною підтримкою UA/RU усуває втрати при автоперекладі та ручній модерації, (3) вбудована інтеграція супутникових сенсорів (FRP, γ) дозволяє виявляти події, що не потрапили до новинних стрічок.

Завдяки локальній колонковій аналітиці система працює на типовій робочій станції (8 ядер, 32 ГБ RAM), що знижує загальну вартість підтримки рішення порівняно з розподіленими кластерами. Запропонований підхід забезпечує не лише низькі затримки та економію пам'яті, але й формує прозорий ланцюжок «дані – оцінка – рішення», придатний для оперативного реагування (моніторинг загроз у реальному часі) та стратегічного планування (оптимізація розміщення інфраструктури захисту). Відтворюваність підтверджена щоденними Parquet-знімками та детерміністичними скриптами, а відповідність вимогам GDPR забезпечена через мінімізацію оприлюднених полів та шифрування даних. Узагальнення результатів експериментальних досліджень показало, що розроблене програмне забезпечення перевершує наявні системи екологічного моніторингу завдяки поєднанню потокового режиму оброблення, багатоджерельної верифікації подій та просторово-часової агрегації на основі шестикутної сітки H3.

На відміну від систем моніторингу Safecast і SaveEcoBot, орієнтованих на окремі типи даних та позбавлених повноцінної автоматичної валідації, запропоноване програмне забезпечення інтегрує OSINT-повідомлення, супутникові спостереження та фізичні вимірювання, що забезпечує зменшення похибок та кількості хибних спрацювань.

ВИСНОВКИ

У дисертаційній роботі вирішене наукове завдання підвищення швидкості, точності та відтворюваності аналізу гетерогенних потокових даних для екологічного моніторингу шляхом створення відповідного алгоритмічного та програмного забезпечення.

Основні наукові та практичні результати дисертаційної роботи полягають у наступному.

1. Проаналізовано сучасні програмні системи екологічного моніторингу (ACLED, GDELT, Copernicus EMS, Liveuamap, SaveEcoBot) та їх обмеження.

Було встановлено, що жодна з наявних систем моніторингу не забезпечує одночасно автоматичне геокодування багатомовних текстів зі збереженням коректності словоформ, інтеграцію текстових OSINT-повідомлень із сенсорними даними у єдиному конвеєрі та формування просторово-часових агрегатів з мілісекундною затримкою.

2. Розроблено метод геокодування зі збереженням коректності для різних словоформ багатомовних OSINT-повідомлень, що поєднує морфологічну нормалізацію (pymorphy2, Stanza), каскад точного, нечіткого та фонетичного зіставлення з реєстром географічних назв та усунення контекстуальної неоднозначності в локальному оточенні НЗ. На відміну від існуючих методів, орієнтованих на англomовні тексти, розроблений метод підвищує частку коректних прив'язувань топонімів на 21% (з 72,1% до 87,3% точної відповідності) та скорочує середній час пошуку координат на 51%.

Розроблено ймовірнісну геопросторову модель прийняття рішень, що інтегрує результати геокодування, багатоджерельної перевірки та доменної агрегації на сітці НЗ з підтримкою сценарних обмежень та LSTM-прогнозуванням.

Розроблено архітектурний шаблон НЗ-DSMV (Dual Scale Materialized Views) для аналізу гетерогенних потоків даних, що забезпечує фіксацію подій на дрібнішій сітці НЗ r_{10} з автоматичною побудовою та підтримкою матеріалізованих подань на рівні r_7 . На відміну від розподілених архітектур з окремими

компонентами агрегації, використання шаблону H3-DSMV дозволяє скоротити середній час відповіді системи у 5 разів.

Визначено та формалізовано процедури доменної агрегації ризику для гетерогенних потокових даних, що поєднують FRP-агрегацію з дедуплікацією супутникових детекцій у 15-хвилинних вікнах на H3 r10, метеокорекцію коефіцієнтом `wind_align`, нормування радіаційних показників через z -індекс γ -випромінювання з 14-денною ковзною базовою лінією та єдину часову базу UTC. Розроблено методику калібрування вагових коефіцієнтів. Аналіз чутливості методами ОАТ та Морріса підтвердив помірну стійкість системи: при варіації ваг на $\pm 10\%$ кореляція з офіційними тривогами залишається в діапазоні 0,75–0,78.

3. Розроблено архітектуру програмного забезпечення та реалізовано модульну систему на основі уніфікованого мікропакетного конвеєра гармонізації (Unified Harmonization Pipeline, UHP) гетерогенних потоків даних, що забезпечує детерміноване перетворення подій до єдиної часової бази, нормалізацію одиниць вимірювання та уніфікацію схем подій у межах одного процесу. На відміну від стандартних багатоетапних ETL-конвеєрів, розроблена архітектура дозволяє скоротити час затримки уніфікації на 80%. Пропускна здатність сенсорної гілки конвеєра (числовий парсинг CSV FIRMS) становить 1,7 млн. рядків/с, текстової OSINT-гілки з морфологічним аналізом $\sim 50\text{--}200$ повідомлень/с, що є достатнім для типових добових обсягів. Реалізовано програмні модулі адаптерного шару (SourceAdapter), NLP-класифікації, каскадного геокодування та подієвого журналу з матеріалізованими поданнями H3-DSMV на базі DuckDB.

4. Проведено експериментальну перевірку розробленого програмного забезпечення за трьома групами сценаріїв (радіаційні аномалії, масовані удари з FRP-детекціями, вторинні пожежі) на даних за період серпень 2022 - травень 2025 рр. Було встановлено, що багатоджерельна валідація фільтрує природні γ -аномалії без втрати чутливості; дедуплікація FRP знижує дублікати до 0,6–1,2%; добові агрегати стійкі до пропусків даних (штучне вилучення 20% записів збільшує помилку лише на 8%); інтегральний показник ризику корелює з офіційними

повітряними тривогами на рівні $r = 0,78$ ($p < 0,001$). Порівняння з системами моніторингу ACLED, GDELT та Liveuamap підтвердило переваги розробленого програмного забезпечення за часом відгуку, морфологічним геокодуванням та інтеграцією сенсорних даних. Медіанний час оновлення подання склав 2,04 мс. Оптимізація розміщення укриттів за MCLP показала, що додавання 10 нових об'єктів підвищує покриття попиту з 50,46% до 71,23%.

Удосконалені процедури доменної агрегації ризику для гетерогенних потокових даних екологічного моніторингу, на відміну від існуючих процедур, поєднують FRP-агрегацію, метеокорекцію для врахування вітру та метеофакторів, нормування радіаційних показників та єдину часову базу.

Запропонований метод інтеграції LSTM-прогнозування у єдиний просторово-часовий конвеєр екологічного моніторингу, на відміну від реактивних методів моніторингу, забезпечує перехід до проактивного режиму підтримки рішень на основі короткострокових прогнозів.

Запропоновані у дисертаційній роботі методи та програмні засоби забезпечують безперервний моніторинг екологічних ризиків, підтвердження подій за кількома джерелами та формування добових просторово-часових індикаторів для підтримки прийняття рішень органами цивільного захисту та екологічного нагляду, а також використання на обмежених обчислювальних ресурсах у режимі автономної роботи.

Результати дисертаційного дослідження були використані в навчальному процесі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» при викладанні дисципліни «Великі дані та аналітика» для студентів другого (магістерського) рівня освіти спеціальності 121 «Інженерія програмного забезпечення» (освітньо-наукова програма "Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем").

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ilin M. O., Oleshchenko L. M. Current state and development prospects of heterogeneous streaming data processing methods. *Information Technology: Computer Science, Software Engineering and Cyber Security*, Вип. 1, 2023. С. 100-106. URL: <https://doi.org/10.32782/IT/2023-1-13>.
2. Oleshchenko L. M., Ilin M. O. Software analysis of radiation air pollution streaming data. *Вісник Херсонського національного технічного університету*. № 2(85), 2023. С. 187-195. URL: <https://doi.org/10.35546/kntu2078-4481.2023.2.26>.
3. Ilin M. O., Oleshchenko L. M. Enhancing distributed stream processing through artificial intelligence: a review and proposal for LSTM integration. *Інфокомунікаційні та комп'ютерні технології*. Том 2. № 06, 2023. С. 66-73. URL: <https://doi.org/10.36994/2788-5518-2023-02-06-07>.
4. Олещенко Л. М., Ільїн М. О. Програмний метод аналізу поточкових даних соціальних мереж за допомогою гібридної моделі нейронної мережі. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. Том 36 (75). № 1, 2025. С. 163-170. URL: <https://doi.org/10.32782/2663-5941/2025.1.2/24>.
5. Ilin M. O., Oleshchenko L. M. Development of a modular data unification pipeline for real-time environmental threat analytics. *Вісник Херсонського національного технічного університету*. Том 2 № 3(94) 2025. С. 211-217. URL: <https://doi.org/10.35546/kntu2078-4481.2025.3.2.26>.
6. Ilin M.O., Oleshchenko L.M. Software-driven data analysis of propaganda narratives and social media engagement. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки*. Том 36 (75). № 4, 2025. С. 155-162. URL: <https://doi.org/10.32782/2663-5941/2025.4.2/21>.
7. Ilin M. O., Oleshchenko L. M. RNN implementation for streaming data analysis and prediction. *International scientific conference «The latest science and technology achievements and their significance for society»*. December 6–7, 2023.

Częstochowa, the Republic of Poland, 2023. P. 6-10. URL: <https://doi.org/10.30525/978-9934-26-388-0-1>.

8. Oleshchenko L., Ilin M. Artificial intelligence methods integrating in streaming data processing systems. X International scientific and practical conference «Modern Trends in the Development of Scientific Space» February 14-16, 2024. Dresden, Germany, 2024. P. 96-101. URL: https://isu-conference.com/wp-content/uploads/2024/02/Modern_trends_in_the_development_of_scientific_space_Feb_14_16_2024.pdf.

9. Ільїн М.О., Олещенко Л.М. Метод та програмне забезпечення для виявлення подій у реальному часі та аналізу трендів з використанням HybridLSTM. Прикладна математика та комп'ютинг. ПМК, 2024: Сімнадцята наук. конф. магістрантів та аспірантів, Київ, 20-22 лист. 2024 р. Київ: «Просвіта», 2024. С. 201-205. ISBN 978-617-7010-37-0.

10. Ilin M., Oleshchenko L. Software pipeline development for OSINT-based geocoding of missile and drone strikes in Ukraine. International Scientific Conference «Science, Technology, and Society in the 21st Century», August 6, 2025, Amsterdam, Netherlands. P. 116-119. URL: <https://doi.org/10.64076/iedc250806.18>.

11. Ilin M., Oleshchenko L. Real-time radiation anomaly mapping using Kp/GOES X-ray correction and H3 spatial aggregation. International Scientific Conference «Trends, Issues, and Challenges in Modern Science», September 5, 2025, Cambridge, UK. P. 86-89. URL: <https://doi.org/10.64076/iedc250905.16>.

12. SDO/HMI Quick-Look Magnetogram: 20230506_000000. URL: https://solarham.com/pictures/2023/may6_2023_mag.jpg.

13. Ukraine Ministry of Environmental Protection and Natural Resources. Ukraine war: environmental impact. 2023. URL: <https://www.topleadprojects.com/environmental-project-main>

14. Xu Y., Wang Z., Dong W., Chou J. Predicting the potential impact of emergency on global grain security: a case of the Russia-Ukraine conflict. Foods. 2023;12:2557. URL: <https://doi.org/10.3390/foods12132557>

15. Pereira P., Baic F., Bogunovic I., Barcelo D. Russian-Ukrainian war impacts on the total environment. *Sci Total Environ.* 2022;837:155856. URL: <https://doi.org/10.1016/j.scitotenv.2022.155865>
16. Racioppi F., Rutter H., Nitzan D., et al. The impact of war on the environment and health: implications for readiness, response, and recovery in Ukraine. *Lancet.* 2022; 400:871–873. URL: [https://www.thelancet.com/journals/lancet/article/PIIS0140-6736\(22\)01739-1/fulltext](https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(22)01739-1/fulltext)
17. Rawtani D., Gupta G., Khatri N., et al. Environmental damages due to the war in Ukraine: a perspective. *Sci Total Environ.* 2022;850:157932. URL: <https://doi.org/10.1016/j.scitotenv.2022.157932>
18. Francis A.F., Krishnamurthy K. Human conflict and ecosystem services: finding the environmental price of warfare. *International Affairs.* 2014;90(4):853–869. URL: <https://doi.org/10.1111/1468-2346.12144>
19. Turos O., Petrosia A., Maremukha T., et al. Assessment of ambient air pollution by particulate matter (PM₁₀, PM_{2.5}) and risk of human health caused by war actions. *Wiadomości Lekarskie.* 2023;76:738–744. URL: <https://doi.org/10.36740/WLek202304106>
20. United Nations Environment Programme. The toxic legacy of the Ukraine war. URL: <https://www.unep.org/news-and-stories/story/toxic-legacy-ukraine-war>
21. Mears M.J., Aslaner D.M., Barson C.T., et al. Health effects following exposure to dust from the World Trade Center disaster: an update. *Life Sci.* 2022;289:120147. URL: <https://doi.org/10.1016/j.lfs.2021.120147>
22. Brown O., Froggatt A., Gozak N., et al. The impact of Russia's war against Ukraine on climate security and climate action. OSCE; 2023. URL: <https://alpanalytica.org/wp-content/uploads/2023/02/Independent-Experts-Analysis-The-impact-of-Russias-war-against-Ukraine-on-climate-security-and-climate-action-9-Feb-23.pdf>

23. Cascio W.E. Wildland fire smoke and human health. *Sci Total Environ.* 2018;624:586–595. URL: <https://www.sciencedirect.com/science/article/pii/S004896971733512X>
24. Zajackeviciute R., Mejia D., Alvarez H., et al. The war impact on air quality in Ukraine. *Sustainability.* 2022;14(21):13832. URL: <https://doi.org/10.3390/su142113832>
25. Zhang C., Hu Q., Su W., et al. Satellite spectroscopy reveals the atmospheric consequences of the 2022 Russia-Ukraine war. *Sci Total Environ.* 2023;869:161759. URL: <https://doi.org/10.1016/j.scitotenv.2023.161759>
26. Phillips L., Crouch D. Have chemical weapons been used in Ukraine? *RUSI;* 2023. URL: <https://rusi.org/explore-our-research/publications/commentary/have-chemical-weapons-been-used-ukraine>
27. Sommerlad J. What are white phosphorus bombs? All we know about incendiary weapons Russia accused of using in Ukraine. *The Independent;* 2023. URL: <https://www.independent.co.uk/news/world/europe/white-phosphorus-weapons-burns-russia-ukraine-b2333819.html>
28. Statement by Dr. Kateryna Bila... at the Fifth Special Session of the Conference of the States Parties to Review the Operation of the Chemical Weapons Convention. 2023. URL: https://www.opcw.org/sites/default/files/documents/2023/05/National_Statement_Ukraine_9C_assistance_protection_pdf.pdf
29. de Klerk L., Shlapak M., Shmurak A., et al. Climate damage caused by Russia's war in Ukraine. 2023. URL: <https://en.ecoaction.org.ua/wp-content/uploads/2023/06/clim-damage-by-russia-war-12-months.pdf>
30. Shumilova O., Tockner K., Sukhodolov A., et al. Impact of the Russia-Ukraine armed conflict on water resources and water infrastructure. *Nat Sustain.* 2023; 6: 578–586. URL: <https://www.nature.com/articles/s41893-023-01068-x>

31. Conflict and Environment Observatory. The coastal and marine environment. 2023. URL: <https://ceobs.org/ukraine-conflict-environmental-briefing-the-coastal-and-marine-environment/>
32. Bordeleau G., Martel R., Ampleman G., Thiboutot S. Environmental impacts of training activities at an air weapons range. *J Environ Qual.* 2008;37:308–317. URL: <https://access.onlinelibrary.wiley.com/doi/10.2134/jeq2007.0204>
33. The World Bank. Updated Ukraine recovery and reconstruction needs assessment. 2023. URL: <https://www.worldbank.org/en/news/press-release/2023/03/23/updated-ukraine-recovery-and-reconstruction-needs-assessment>
34. ReliefWeb. Ukraine: thousands of children lack safe drinking water as Kakhovka Dam crisis unravels. 2023. URL: <https://reliefweb.int/report/ukraine/ukraine-thousands-children-lack-safe-drinking-water-kakhovka-dam-crisis-unravels>
35. Hamilton R.E. Coal mines, land mines and nuclear bombs: the environmental cost of the war in eastern Ukraine. Foreign Policy Research Institute; 2019. URL: <https://www.fpri.org/article/2019/09/coal-mines-land-mines-and-nuclear-bombs-the-environmental-cost-of-the-war-in-eastern-ukraine/>
36. FairPlanet. The radioactive threat to Ukraine's groundwater. 2022. URL: <https://www.fairplanet.org/editors-pick/the-radioactive-threat-to-ukraines-groundwater/>
37. Hryhorczuk D., Levy B.S., Prodanchuk M., et al. The environmental health impacts of Russia's war on Ukraine. *J Occup Med Toxicol.* 2024;19:1. URL: <https://doi.org/10.1186/s12995-023-00398-y>
38. Assessing Environmental Damage in Ukraine. 2024. URL: <https://ceobs.org/wp-content/uploads/2024/02/Assessing-Environmental-Damage-in-Ukraine.pdf>

39. Shevchuk S.A., et al. The Use of Remote Sensing Data for Investigation of Environmental Consequences of Russia-Ukraine War. *Journal of Landscape Ecology*. 2022;15(3):36–53. URL: <https://doi.org/10.2478/jlecol-2022-0017>
40. Hanoshenko O., Halaktionov M., Huber-Humer M. Exploratory study on the impact of military actions on the environment and infrastructure in the current Ukraine war with a specific focus on waste management. *Waste Management & Research*. 2024;43(8):1245–1259. URL: <https://doi.org/10.1177/0734242X241305909>
41. Safecast / SaveDnipro. Press release: Safecast/SaveDnipro joint independent realtime radiation monitoring system in Ukraine. 2024. URL: <https://safecast.org/press-release-safecastsavednipro-joint-independent-realtime-radiation-monitoring-system-now-operational-in-ukraine/>
42. Maslak O., Yakovenko Y., Maslak M., Hlazunova O., Hryshko N., Pirogov D. Methods of environmental monitoring, analysis and assessment of the consequences of military actions. 2025. URL: <https://eage.in.ua/wp-content/uploads/2025/04/Mon25-187.pdf>
43. Borovyk K., Gapon S., Pyshnograiev I. The impact of temporary occupation and hostilities on the water conditions of the North Crimean and Kakhovka Main Canals. *Ecological Safety and Balanced Use of Resources*. 2024;15(2):9–21. URL: <https://doi.org/10.69628/esbur/2.2024.09>
44. Боровик К. В., Гапон С. В., Пишнограєв І. О., Хашевацька А. І. Моделювання придатності територій для розміщення критичної інфраструктури України в умовах військових загроз. *Український журнал дистанційного зондування Землі*. 2024;11(4):38–49. URL: <https://doi.org/10.36023/ujrs.2024.11.4.276>
45. Berezenko K., Sopov D., Melnychuk S., Gurets N., Girzheva O., Marynets O., Ignatenko O., Dzhygyrey I., Cherednychenko I., Prokopenko O. Integrated Ecological Assessment of A River Environment Based on Water Quality Criteria and Pollution Indicators Analysis. *Eco. Env. & Cons.* 2024;30(3):468–482. URL:

<https://dspace.luguniv.edu.ua/xmlui/bitstream/handle/123456789/10508/Integrated%20Ecological%20Assessment.pdf?isAllowed=y&sequence=1>

46. Кушенко С. Програмна система виявлення та аналізу об'єктів на супутникових знімках із використанням Python та OpenCV. Київ: НТУУ «КПІ ім. Ігоря Сікорського». 2023. URL: <https://ela.kpi.ua/items/4e0ae703-8048-43f7-9fed-769e80b94f3f>

47. Ласкавий О. Пошукова GIS-система екологічно небезпечних промислових об'єктів. Київ: НТУУ «КПІ ім. Ігоря Сікорського». 2021. URL: <https://ela.kpi.ua/items/393bc70c-cbdb-477f-89a7-a7c8aa1a3849>

48. Гребенюк Т., Артьомов Р., Ремез Н., Броницький В. GIS-моделювання стану підземних вод на основі супутникових та сенсорних даних. Energy: Economics, Technology, Ecology. 2024;(2):45–53. URL: <https://energy.kpi.ua/article/view/315602>

49. Imran M., Castillo C., Lucas J., Meier P., Vieweg S. AIDR: Artificial Intelligence for Disaster Response (WWW'14 companion). 2014. URL: https://chato.cl/papers/demo_2014_aidr_artificial_intelligence_disaster_response.pdf

50. Sakaki T., Okazaki M., Matsuo Y. Earthquake shakes Twitter users: Real-time event detection by social sensors. Proceedings of the 19th International World Wide Web Conference (WWW'10). 2010. URL: <https://doi.org/10.1145/1772690.1772777>

51. Olteanu A., Vieweg S., Castillo C. CrisisLex: A Lexicon for Collecting and Analyzing Crisis-Related Social Media Messages. ICWSM 2014. URL: <https://ojs.aaai.org/index.php/ICWSM/article/view/14538>

52. Leidner J.L. Toponym Resolution in Text. Universal Publishers. 2008. ISBN: 9781581123845. URL: <https://www.universal-publishers.com/book.php?method=ISBN&book=158112384X>

53. Gritta M., Pilehvar M.T., Collier N. A pragmatic guide to geoparsing evaluation: Toponyms, Named Entity Recognition and pragmatics. Lang Resour Eval. 2020;54(3):683–712. URL: <https://doi.org/10.1007/s10579-019-09475-3>

54. Castillo C. Big crisis data: Social media in disasters and time-critical situations. Cambridge University Press. 2016. URL: <https://doi.org/10.1017/9781316476840>
55. Havas C., Resch B. Portability of semantic and spatial-temporal machine learning methods to analyse social media for near-real-time disaster monitoring. *Nat Hazards*. 2021;108(3):2939–2969. URL: <https://doi.org/10.1007/s11069-021-04808-4>
56. Li J., et al. Social Media: New Perspectives to Improve Remote Sensing for Emergency Response. *Proceedings of the IEEE*. 2017;105(10):1900–1912. URL: <https://doi.org/10.1109/JPROC.2017.2684460>
57. Schmidt S., Friedemann M., Hanny D., Resch B., Riedlinger T., Mühlbauer M. Enhancing satellite-based emergency mapping: Identifying wildfires through geo-social media analysis. *Big Earth Data*. 2025;1–23. URL: <https://doi.org/10.1080/20964471.2025.2454526>
58. Rocca R., Tamagnone N., Fekih S., Contla X., Rekabsaz N. Natural language processing for humanitarian action: Opportunities, challenges, and the path toward humanitarian NLP. *Front Big Data*. 2023;6:1082787. URL: <https://doi.org/10.3389/fdata.2023.1082787>
59. Zhong W., Mei X., Niu F., Fan X., Ou S., Zhong S. Fusing Social Media, Remote Sensing, and Fire Dynamics to Track Wildland-Urban Interface Fire. *Remote Sensing*. 2023;15(15):3842. URL: <https://doi.org/10.3390/rs15153842>
60. Imran M., Ofli F., Caragea D., Torralba A. Using AI and Social Media Multimodal Content for Disaster Response and Management: Opportunities, Challenges, and Future Directions. *Information Processing & Management*. 2020;57(5):102261. URL: <https://doi.org/10.1016/j.ipm.2020.102261>
61. Niu J., Stillman M., Seeberger P., Kruspe A. A Dataset of Open Source Intelligence (OSINT) Tweets about the Russo-Ukrainian War. *ISCRAM 2024*. URL: <https://doi.org/10.59297/377r3945>
62. EcoZagroza. Інформаційна панель з даними про екологічні загрози. URL: <https://ecozagroza.gov.ua/en>

63. Екодозор. Екологічні наслідки та ризики бойових дій в Україні. URL: <https://www.ecodozor.org/index.php?lang=en>
64. SaveEcoBot. Військові злочини проти довкілля України. URL: <https://www.saveecobot.com/en/features/environmental-crimes>
65. Ecoaction. URL: <https://en.ecoaction.org.ua/>
66. Copernicus EMS. URL: <https://emergency.copernicus.eu/>
67. ACLED (Armed Conflict Location & Event Data Project). URL: <https://acleddata.com>
68. GDELT (Global Database of Events, Language, and Tone). URL: <https://www.gdeltproject.org>
69. Liveuamap. URL: <https://liveuamap.com>
70. Fowler M. Event Sourcing. 2005. URL: <https://martinfowler.com/eaaDev/EventSourcing.html>
71. Microsoft Azure. Event Sourcing pattern. 2020. URL: <https://learn.microsoft.com/azure/architecture/patterns/event-sourcing>
72. Upsolver Team. CQRS, Event Sourcing Patterns and Database Architecture. 17 Feb 2022. URL: <https://www.upsolver.com/blog/cqrs-event-sourcing-patterns-and-database-architecture>
73. Microsoft Azure. Big Data Architectures – Lambda vs. Kappa. 2023. URL: <https://learn.microsoft.com/azure/architecture/data-guide/advanced-analytics/stream-processing>
74. Uber Engineering. H3: Uber's Hexagonal Hierarchical Spatial Index. 2018. URL: <https://www.uber.com/blog/h3/>
75. Raasveldt M., Mühleisen H. DuckDB: An Embeddable Analytical Database. SIGMOD '19. 2019. URL: <https://doi.org/10.1145/3299869.3319885>
76. Boeing G. Modeling and Analyzing Urban Networks and Amenities with OSMnx. 2025 (online preprint). URL: <https://arxiv.org/abs/2505.00736>
77. SaveEcoBot. Radiation Maps in Ukraine (Background Radiation Monitoring). URL: <https://www.saveecobot.com/en/map>

78. SaveEcoBot. SaveEcoBot API Documentation. 2023-10-29. URL: <https://www.saveecobot.com/en/api/docs>
79. NASA LANCE. Fire Information for Resource Management System (FIRMS). URL: <https://earthdata.nasa.gov/firms>
80. Bellingcat. LiveUMap – Online Investigation Toolkit. 2023. URL: <https://www.bellingcat.com/resources/how-tos/2023/01/20/liveuamap-online-investigation-toolkit/>
81. Conneau A., Khandelwal K., Goyal N., Chaudhary V., Wenzek G., et al. Unsupervised Cross-lingual Representation Learning at Scale. ACL 2020. URL: <https://doi.org/10.18653/v1/2020.acl-main.747>
82. He P., Gao J., Chen W., et al. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training. ICLR 2023. URL: <https://openreview.net/forum?id=x7Ao3GQ1L3b>
83. Lin T.-Y., Goyal P., Girshick R., He K., Dollár P. Focal Loss for Dense Object Detection. 2017. URL: <https://arxiv.org/abs/1708.02002>
84. Müller M., et al. Textual Data Augmentation for Small Data Domains. 2022. URL: <https://arxiv.org/abs/2202.06417>
85. Ratner A., Bach S.H., Ehrenberg H., Fries J., Wu S., Ré C. Snorkel: Rapid Training Data Creation with Weak Supervision. VLDB. 2017;11(3):269–282. URL: <http://www.vldb.org/pvldb/vol11/p269-ratner.pdf>
86. Zhang Q., et al. Combining NSP and NER for Public Opinion Event Extraction. Frontiers in Physics. 2022;10:1044919. URL: <https://doi.org/10.3389/fphy.2022.1044919>
87. Semnani S.J., Zhang P., Zhai W., Li H., Beauchamp R., Billing T., Kishi K., Li M., Lam M.S. LEMONADE: A Large Multilingual Expert-Annotated Abstractive Event Dataset for the Real World. Findings of ACL 2025. URL: <https://aclanthology.org/2025.findings-acl.1325/>

88. Dyakonov R., et al. A Geoparsing Pipeline for Multilingual Social Media Posts from Ukraine. Proc. GeoInt 2023 (CEUR-WS 3969). URL: <http://ceur-ws.org/Vol-3969/>
89. Gomes D., et al. Fine-Tuning Transformers for Toponym Resolution. GeoExT @ ECIR 2024 (CEUR-WS 3683). URL: <http://ceur-ws.org/Vol-3683/>
90. Qi P., Zhang Y., Zhang Y., Bolton J., Manning C.D. Stanza: A Python NLP Toolkit for Many Human Languages. ACL Demos 2020. URL: <https://doi.org/10.18653/v1/2020.acl-demos.14>
91. GeoNames. Geographical Database. URL: <https://www.geonames.org>
92. Wikipedia. Admiralty Code (NATO Source Reliability). URL: https://en.wikipedia.org/wiki/Admiralty_code
93. Agius M. Earthquake shakes Twitter users... EGU Seismology Division Blog. 2014-04-02. URL: <https://blogs.egu.eu/divisions/sm/2014/04/02/earthquake-shakes-twitter-users/>
94. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. EMNLP 2019. URL: <https://doi.org/10.18653/v1/D19-1410>
95. Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A., et al. Transformers: State-of-the-Art NLP. EMNLP 2020 Demos. URL: <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
96. Bouarab H., Smaili K., et al. A Survey of Multimodal Event Detection via Data Fusion. VLDB Journal. 2023. URL: <https://doi.org/10.1007/s00778-023-00836-7>
97. Korzeniowski A., Kejriwal M., et al. Toponym Resolution as a Similarity Problem. Proc. CLEF 2023 (GeoMean). URL: <https://ceur-ws.org/Vol-3497/>
98. Halterman A. Mordecai 3: A Neural Geoparser and Event Geocoder. 2023. URL: <https://arxiv.org/abs/2303.13675>
99. Aini A.N., Prastyo D.D., et al. Enhanced Earthquake Risk Analysis using H3 Spatial Indexing. IOP Conf. Ser.: Earth Environ. Sci. 2023;1245:012014. URL: <https://doi.org/10.1088/1755-1315/1245/1/012014>

100. Church R., ReVelle C. The Maximal Covering Location Problem. *Papers Reg. Sci.* 1974;32(1):101–118. URL: <https://doi.org/10.1111/j.1435-5597.1974.tb00902.x>
101. Kireyev K., Palen L., Anderson K.M. Tailoring Tweet Analysis for Emergency Response. *SMERP Workshop, ACL* 2016. URL: <https://aclanthology.org/W16-0708/>
102. Gonzales C. Scorched Earth: Using NASA Fire Data to Monitor War Zones. *Bellingcat.* 2022-10-04. URL: <https://www.bellingcat.com/resources/2022/10/04/scorched-earth-using-nasa-fire-data-to-monitor-war-zones/>
103. Korobov M. Morphological Analyzer and Generator for RU/UK Languages (pymorphy2). 2015. URL: <https://arxiv.org/abs/1503.07283>
104. OSM Wiki. Valhalla Routing Engine. URL: <https://wiki.openstreetmap.org/wiki/Valhalla>
105. Raleigh C., Hegre H. Introducing ACLED: Armed Conflict Location & Event Data. *PRIO Conference Paper.* 2005. URL: <https://www.prio.org/publications/3898>
106. GDELT Project. GDELT Conflict Dataset (1979–2021). *Kaggle.* URL: <https://www.kaggle.com/datasets/vladproex/gdelt-conflict-events-1979-2021>
107. GraphHopper. GraphHopper Routing Engine 9.0 Released. 2024. URL: <https://www.graphhopper.com/blog/2024/04/23/graphhopper-routing-engine-9-0-released/>
108. Introduction | H3. URL: <https://h3geo.org/docs/>
109. DuckDB – An in-process SQL OLAP database management system. URL: <https://duckdb.org/> — Дата доступу : 6 жовтня 2025.
110. The first environmental system in Ukraine — SaveEcoBot. URL: <https://www.saveecobot.com/en>
111. RapidFuzz. URL: <https://pypi.org/project/RapidFuzz/>
112. OSMnx 2.0.6 documentation URL: <https://osmnx.readthedocs.io/en/stable/>

113. Open-Meteo API documentation. URL: <https://open-meteo.com/en/docs>
114. DuckDB is Probably the Most Important Geospatial Software of the Last Decade. URL: <https://www.dbreunig.com/2025/05/03/duckdb-is-the-most-impactful-geospatial-software-in-a-decade.html>
115. The Maximal Coverage Location Problem — spopt v0.7.0 Manual. URL: <https://pysal.org/spopt/notebooks/mclp.html>.

ДОДАТКИ

ДОДАТОК А

АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ В
НДР

ЗАТВЕРДЖУЮ

проректор з наукової роботи

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря

Сікорського»



Сергій СТИРЕНКО

12 2025 року

АКТ

про використання результатів дисертаційного дослідження аспіранта кафедри програмного забезпечення комп'ютерних систем (ПЗКС) факультету прикладної математики Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» Ільїна Максима Олександровича на тему «Алгоритмічне та програмне забезпечення для виявлення та прогнозування екологічних загроз на основі аналізу гетерогенних потокових даних» на здобуття наукового ступеня доктора філософії за спеціальністю 121 «Інженерія програмного забезпечення».

Комісія у складі: голова – декан факультету прикладної математики, д.т.н., проф. Іван ДИЧКА; члени комісії – професор кафедри ПЗКС, д.т.н., проф. Віктор ЛЕГЕЗА; завідувач кафедри ПЗКС, д.т.н., доц. Євгенія СУЛЕМА, цим Актом засвідчує, що результати дисертаційної роботи Максима ІЛЬІНА отримані ним особисто та використані в НДР «Методи оптимізації продуктивності програмного забезпечення та використання технологій штучного інтелекту для вдосконалення програмних систем аналітики великих даних», що виконується за ініціативою авторів, № держреєстрації 0124U001790, закінчення у 2026 р.

В НДР використані наступні результати:

- метод геокодування OSINT-повідомлень, який дозволяє підвищити точність просторової прив'язки повідомлень, зменшити частку хибних або невизначених координат та забезпечити стабільність просторово-часової агрегації даних.

- метод багатоджерельної верифікації подій для геопросторових потоків, що узгоджує текстові дані зі супутниковими термічними виявленнями та радіаційними сенсорами. Використання цього методу знижує дублювання виявлень, підвищує достовірність підтвердження подій техногенного та природного характеру та забезпечує формування узгоджених добових агрегатів ризику для задач прогнозування екологічних загроз.

Голова комісії
д.т.н., проф.

Іван ДИЧКА

Члени комісії

д.т.н., проф.

Віктор ЛЕГЕЗА

д.т.н., доц.

Євгенія СУЛЕМА

ДОДАТОК Б

АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ В НАВЧАЛЬНИЙ ПРОЦЕС

ЗАТВЕРДЖУЮ

проректор з навчальної роботи

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря
Сікорського»

Тетяна ЖЕЛЯСКОВА

12 2025 року



про впровадження в навчальний процес Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» наукових положень і результатів дисертаційної роботи Максима ІЛЬІНА на тему «Алгоритмічне та програмне забезпечення для виявлення та прогнозування екологічних загроз на основі аналізу гетерогенних потокових даних» на здобуття наукового ступеня доктора філософії за спеціальністю 121 «Інженерія програмного забезпечення».

Комісія в складі голови – завідувача кафедри програмного забезпечення комп'ютерних систем, д.т.н., доцента Євгенії СУЛЕМИ та членів комісії: професора кафедри програмного забезпечення комп'ютерних систем, д.т.н., професора Віктора ЛЕГЕЗИ та доцента кафедри програмного забезпечення комп'ютерних систем к.т.н., доцента Тетяни ЗАБОЛОТНЬОЇ, склала цей акт про те, що наукові положення та результати дисертаційної роботи на здобуття наукового ступеня доктора філософії аспіранта Максима ІЛЬІНА на тему «Алгоритмічне та програмне забезпечення для виявлення та прогнозування екологічних загроз на основі аналізу гетерогенних потокових даних», а саме технології вебскрапінгу, вебкраулінгу, індексації, вебавтоматизації та методи попередньої обробки та аналізу постів соціальної мережі X, введено в навчальний процес кафедри програмного забезпечення комп'ютерних систем Факультету прикладної математики при викладанні дисципліни «Великі дані та аналітика» спеціальності 121 «Інженерія програмного забезпечення» для студентів другого (магістерського) рівня вищої освіти (2025–2026).

Лекція: «Технології вебскрапінгу, вебкраулінгу, індексації та вебавтоматизації».

Практичне заняття: «Попередня обробка та аналіз твітів. Завантаження файлу. Аналіз структури даних: очищення тексту (видалення стоп-слів, символів, хештегів, згадок), нормалізація, токенизація, лематизація. Вивчення розподілу твітів за часом, мовами та основними темами».

Голова комісії:

Завідувач кафедри програмного забезпечення
комп'ютерних систем
доктор технічних наук, доцент

Євгенія СУЛЕМА

Члени комісії:

Професор кафедри програмного забезпечення
комп'ютерних систем
доктор технічних наук, професор

Віктор ЛЕГЕЗА

Доцент кафедри програмного забезпечення
комп'ютерних систем
кандидат технічних наук, доцент

Тетяна ЗАБОЛОТНЯ

ДОДАТОК В

ПУБЛІКАЦІЇ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Ilin M. O., Oleshchenko L. M. Current state and development prospects of heterogeneous streaming data processing methods. Information Technology: Computer Science, Software Engineering and Cyber Security, Вип. 1, 2023. С. 100-106.

Здобувач виконав аналіз сучасних підходів до програмної обробки гетерогенних потокових даних, узагальнив вимоги до уніфікації схем, форматів і часових узгоджень та сформулював напрями розвитку потокових конвеєрів.

2. Oleshchenko L. M., Ilin M. O. Software analysis of radiation air pollution streaming data. Вісник Херсонського національного технічного університету. № 2(85), 2023. С. 187-195.

Здобувач здійснив програмний аналіз відкритих рядів радіаційного моніторингу, розробив методи нормування сигналів, виявлення аномалій і збоїв датчиків та виконав експериментальну перевірку результатів.

3. Ilin M. O., Oleshchenko L. M. Enhancing distributed stream processing through artificial intelligence: a review and proposal for LSTM integration. Інфокомунікаційні та комп'ютерні технології. Том 2. № 06, 2023. С. 66-73.

Здобувач проаналізував архітектури розподіленої потокової обробки та обґрунтував інтеграцію LSTM у CEP-контури для реалізації прогностичної аналітики.

4. Олещенко Л. М., Ільїн М. О. Програмний метод аналізу потокових даних соціальних мереж за допомогою гібридної моделі нейронної мережі. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 36 (75). № 1, 2025. С. 163-170.

Здобувач запропонував гібридну нейронну модель для подієвого аналізу потоків соціальних мереж та експериментально підтвердив зниження MAE порівняно з базовою LSTM.

5. Ilin M. O., Oleshchenko L. M. Development of a modular data unification pipeline for real-time environmental threat analytics. Вісник Херсонського національного технічного університету. Том 2 № 3(94) 2025. С. 211-217.

Здобувач розробив модульний конвеєр уніфікації гетерогенних даних для аналітики екологічних ризиків у реальному часі з використанням H3-агрегації.

6. Ilin M.O., Oleshchenko L.M. Software-driven data analysis of propaganda narratives and social media engagement. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 36 (75). № 4, 2025. С. 155-162.

Здобувач побудував відтворюваний конвеєр аналізу поточкових даних, реалізував інженерію інтерпретованих ознак та виконав структурно-поведінковий аналіз взаємодій.

7. Ilin M. O., Oleshchenko L. M. RNN implementation for streaming data analysis and prediction. International scientific conference «The latest science and technology achievements and their significance for society». December 6–7, 2023. Częstochowa, the Republic of Poland, 2023. P. 6-10.

Здобувач реалізував RNN-підхід до поточкового аналізу та прогнозування даних у режимі, наближеному до реального часу та навів результати дослідження.

8. Oleshchenko L., Ilin M. Artificial intelligence methods integrating in streaming data processing systems. X International scientific and practical conference «Modern Trends in the Development of Scientific Space» February 14-16, 2024. Dresden, Germany, 2024. P. 96-101.

Здобувач систематизував сценарії інтеграції методів штучного інтелекту в системи обробки гетерогенних поточкових даних та сформулював практичні рекомендації щодо їх застосування.

9. Ільїн М.О., Олещенко Л.М. Метод та програмне забезпечення для виявлення подій у реальному часі та аналізу трендів з використанням HybridLSTM. Прикладна математика та комп'ютинг. ПМК, 2024: Сімнадцята наук. конф. магістрантів та аспірантів, Київ, 20-22 лист. 2024 р. Київ: «Просвіта», 2024. С. 201-205.

Здобувач розробив метод і програмне забезпечення на основі моделі HybridLSTM для виявлення подій і аналізу трендів у потоках соціальних мереж.

10. Ilin M., Oleshchenko L. Software pipeline development for OSINT-based geocoding of missile and drone strikes in Ukraine. International Scientific Conference «Science, Technology, and Society in the 21st Century», August 6, 2025, Amsterdam, Netherlands. P. 116-119.

Здобувач створив програмний конвеєр OSINT-геокодування з локальним реєстром географічних назв та морфологічною нормалізацією для автономної обробки поточкових даних.

11. Ilin M., Oleshchenko L. Real-time radiation anomaly mapping using Kp/GOES X-ray correction and H3 spatial aggregation. International Scientific Conference «Trends, Issues, and Challenges in Modern Science», September 5, 2025, Cambridge, UK. P. 86-89.

Здобувач розробив програмний конвеєр картування радіаційних аномалій у реальному часі з корекцією за Kp-індексом, GOES X-ray та просторовою H3-агрегацією.

ДОДАТОК Г

ЛІСТИНГИ КОДУ МОДУЛЯ ГЕОКОДУВАННЯ ЗІ ЗБЕРЕЖЕННЯМ КОРЕКТНОСТІ ДЛЯ РІЗНИХ СЛОВОФОРМ БАГАТОМОВНИХ ПОВІДОМЛЕНЬ

Лістинг 1. Типи збігу та рівні адміністративної ієрархії

```
class MatchType(Enum):
    """Тип збігу при геокодуванні (відповідно до Табл. 2.4)."""
    EXACT = auto()      # Точний лексичний збіг
    PARTIAL = auto()    # Частковий лексичний збіг
    FUZZY = auto()      # Нечіткий (Левенштейн/Джаро-Вінклер)
    PHONETIC = auto()   # Фонетичний (Soundex/Metaphone)
    SEMANTIC = auto()   # Семантичний (S-BERT)
    CONTEXTUAL = auto() # Контекстний (НЗ-оточення)
    NONE = auto()       # Збіг не знайдено

class AdminLevel(Enum):
    """Рівні адміністративної ієрархії України."""
    COUNTRY = 0        # Країна
    OBLAST = 1         # Область
    RAION = 2          # Район
    HRMADA = 3         # Громада
    SETTLEMENT = 4     # Населений пункт
    DISTRICT = 5       # Район міста
```

Лістинг 2. Ваги компонентів складеного балу

```
@dataclass(frozen=True)
class CascadeWeights:
    """
    score_final = w_lex·s_lex + w_fuzzy·s_fuzzy + w_phon·s_phon +
                  w_sem·s_sem + w_ctx·s_ctx
    """
    w_lexical: float = 0.35
    w_fuzzy: float = 0.25
    w_phonetic: float = 0.10
    w_semantic: float = 0.15
    w_context: float = 0.15
    def __post_init__(self) -> None:
        total = (self.w_lexical + self.w_fuzzy + self.w_phonetic +
                 self.w_semantic + self.w_context)
        if not math.isclose(total, 1.0, rel_tol=1e-6):
            raise ValueError(f"Сума ваг має дорівнювати 1.0, отримано:
{total:.6f}")
```

Лістинг 3. Ваги контекстного балу

```
@dataclass(frozen=True)
```

```

class ContextWeights:
    """
    s_ctx = w1·I(match_admin_path) + w2·exp(-λΔt) + w3·exp(-d/R) +
w4·freq_mention
    """
    w_admin_match: float = 0.30
    w_temporal_decay: float = 0.25
    w_spatial_decay: float = 0.25
    w_frequency: float = 0.20
    lambda_t: float = 0.1    # Згасання за ~10 днів
    radius_km: float = 50.0  # Радіус впливу

```

Лістинг 4. Структура кандидата геокодування з обчисленням фінального балу

```

@dataclass
class GeoCandidate:
    geoname_id: int
    name_canonical: str
    name_matched: str
    latitude: float
    longitude: float
    h3_index: str
    admin_level: AdminLevel
    admin_path: List[str]
    population: int = 0
    score_lexical: float = 0.0
    score_fuzzy: float = 0.0
    score_phonetic: float = 0.0
    score_semantic: float = 0.0
    score_context: float = 0.0
    match_type: MatchType = MatchType.NONE
    @property
    def score_final(self) -> float:
        """Обчислення фінального балу за формулою 2.5."""
        weights = CascadeWeights()
        return (weights.w_lexical * self.score_lexical +
                weights.w_fuzzy * self.score_fuzzy +
                weights.w_phonetic * self.score_phonetic +
                weights.w_semantic * self.score_semantic +
                weights.w_context * self.score_context)

```

Лістинг 5. Морфологічна нормалізація топонімів

```

class MorphologyNormalizer:
    """
    Приклади
    "Києві" → "Київ"
    "в Одесі" → "Одеса"
    "м. Івано-Франківськ" → "Івано-Франківськ"
    "Львівщині" → "Львівська область"
    """
    ADMIN_ABBREVIATIONS = {"м.": "", "с.": "", "смт.": "", "обл.": "область",
"р-н": "район"}

```

```

PREPOSITIONS = {"в", "у", "на", "з", "із", "до", "під", "над", "біля",
"поблизу"}
@lru_cache(maxsize=5000)
def normalize(self, text: str, lang: str = "uk") -> str:
    if not text:
        return ""
    normalized = " ".join(text.strip().split())
    normalized = self._re_prepositions.sub("", normalized)      # Видалення
прийменників
    normalized = self._expand_abbreviations(normalized)        # Розгортання
скорочень
    normalized = self._normalize_regional(normalized)           # Обробка
регіональних назв
    normalized = self._lemmatize(normalized, lang)              #
Морфологічна лематизація
    return self._smart_capitalize(normalized.strip())

```

Лістинг 6. Алгоритм подібності Джаро-Вінклера

```

def jaro_winkler_similarity(s1: str, s2: str, prefix_weight: float = 0.1) ->
float:
    """Подібність Джаро-Вінклера з бонусом за спільний префікс."""
    jaro_sim = jaro_similarity(s1, s2)
    s1_lower, s2_lower = s1.lower(), s2.lower()
    prefix_len = 0
    for i in range(min(len(s1_lower), len(s2_lower), 4)):
        if s1_lower[i] == s2_lower[i]:
            prefix_len += 1
        else:
            break
    return jaro_sim + prefix_len * prefix_weight * (1 - jaro_sim)
def combined_fuzzy_score(s1: str, s2: str) -> float:
    """Комбінований нечіткий бал (Левенштейн + Джаро-Вінклер)."""
    lev_sim = levenshtein_similarity(s1, s2)
    jw_sim = jaro_winkler_similarity(s1, s2)
    return 0.4 * lev_sim + 0.6 * jw_sim

```

Лістинг 7. Фонетичний алгоритм Soundex для кирилиці

```

class CyrillicSoundex:
    """Адаптація Soundex"""
    PHONETIC_GROUPS = {
        "а": "0", "е": "0", "є": "0", "и": "0", "і": "0", "о": "0", "у": "0", #
Голосні
        "б": "1", "в": "1", "м": "1", "п": "1", "ф": "1", #
Губні
        "д": "2", "т": "2", #
Зубні
        "ж": "3", "з": "3", "с": "3", "ц": "3", "ч": "3", "ш": "3", "щ": "3", #
Шиплячі
        "г": "4", "к": "4", "х": "4", "ґ": "4", #
Гортанні

```

```

        "л": "5", "н": "5", "р": "5",
# Носові/плавні
        "й": "6",
# Й-подібні
    }
    def encode(self, text: str) -> str:
        text = re.sub(r'^а-яієґ', '', text.lower().strip())
        if not text:
            return ""
        result = [text[0].upper()]
        prev_code = self.PHONETIC_GROUPS.get(text[0], "")
        for char in text[1:]:
            code = self.PHONETIC_GROUPS.get(char, "")
            if code and code != "0" and code != prev_code:
                result.append(code)
            prev_code = code if code else prev_code
        return "".join(result)[:self._code_length].ljust(self._code_length, "0")

```

Лістинг 8. Фонетичний алгоритм Metaphone для кирилиці

```

class CyrillicMetaphone:
    """Metaphone з фонетичними трансформаціями української мови."""
    TRANSFORMATIONS = [
        (r'б$', 'п'), (r'в$', 'ф'), (r'г$', 'х'), (r'д$', 'т'), # Оглушення
        (r'стн', 'сн'), (r'здн', 'зн'), (r'стл', 'сл'), # Спрощення груп
        (r'сш', 'ш'), (r'жж', 'ж'), (r'тс', 'ц'), (r'дс', 'ц'), # Асиміляція
        (r'ья', 'я'), (r'ье', 'е'), (r'ьо', 'о'), # Йотовані
    ]
    def encode(self, text: str) -> str:
        text = text.lower().strip()
        for pattern, replacement in self._compiled_rules:
            text = pattern.sub(replacement, text)
        result = []
        prev_char = ""
        for char in text:
            if char != prev_char and char.isalpha():
                result.append(char)
            prev_char = char
        return "".join(result)[:self._max_length].upper()

```

Лістинг 9. Каскадне зіставлення з реєстром географічних назв

```

def cascade_match(self, normalized: str, h3_context: Optional[str] = None) ->
List[GeoCandidate]:
    """
    Етапи каскаду:
    1. Точний збіг (лексичний)
    2. Нечіткий збіг ( $\tau_1 \approx 0.85$ )
    3. Фонетичний збіг ( $\tau_2 \approx 0.8$ )
    4. Семантичний збіг ( $\tau_3 \approx 0.6-0.7$ )
    """
    candidates = []
    # Етап 1: Точний збіг

```

```

exact_matches = self._gazetteer.search_exact(normalized)
if exact_matches:
    candidates.extend(exact_matches)
# Етап 2: Нечіткий збір
if len(candidates) < 3:
    fuzzy_matches = self._gazetteer.search_fuzzy(normalized,
threshold=self._thresholds.tau_fuzzy)
    for candidate in fuzzy_matches:
        if candidate.geoname_id not in [c.geoname_id for c in candidates]:
            candidate.score_phonetic = combined_phonetic_score(normalized,
candidate.name_canonical)
            if candidate.score_phonetic >= self._thresholds.tau_phonetic:
                candidates.append(candidate)
# Етап 3: Фонетичний збір
if not candidates:
    all_candidates = self._gazetteer.search_fuzzy(normalized, threshold=0.5,
limit=50)
    for candidate in all_candidates:
        phonetic_score = combined_phonetic_score(normalized,
candidate.name_canonical)
        if phonetic_score >= self._thresholds.tau_phonetic:
            candidate.score_phonetic = phonetic_score
            candidate.match_type = MatchType.PHONETIC
            candidates.append(candidate)
# Етап 4: Семантичний збір
if not candidates and self._use_semantic:
    semantic_scores = self._semantic_matcher.batch_similarity(normalized,
all_names[:100])
    # ... фільтрація за порогом tau_semantic
# Етап 5: Контекстне ранжування
if h3_context and candidates:
    candidates = self._apply_context_scores(candidates, h3_context)
    candidates.sort(key=lambda c: c.compute_final_score(self._weights),
reverse=True)
return candidates

```

ЛІСТИНГ 10. Застосування контекстних балів

```

def _apply_context_scores(self, candidates: List[GeoCandidate], h3_context: str)
-> List[GeoCandidate]:
    """s_ctx = w1·I(match_admin_path) + w2·exp(-λΔt) + w3·exp(-d/R) +
w4·freq_mention"""
    cw = self._context_weights
    context_neighbors = self._gazetteer.search_by_h3(h3_context, k_ring=2)
    context_ids = {c.geoname_id for c in context_neighbors}
    for candidate in candidates:
        ctx_score = 0.0
        # w1: Збір адміністративного шляху
        if candidate.geoname_id in context_ids:
            ctx_score += cw.w_admin_match * 1.0
        # w3: Просторова близькість
        candidate_lat_lng = h3.cell_to_latlng(candidate.h3_index)
        context_lat_lng = h3.cell_to_latlng(h3_context)
        distance_km = self._haversine_distance(*candidate_lat_lng,
*context_lat_lng)

```

```

    spatial_decay = math.exp(-distance_km / cw.radius_km)
    ctx_score += cw.w_spatial_decay * spatial_decay
    # w4: Частотність (населення як проксі)
    if candidate.population > 0:
        freq_score = min(1.0, math.log10(candidate.population + 1) / 7)
        ctx_score += cw.w_frequency * freq_score
    candidate.score_context = ctx_score
return candidates

```

Лістинг 11. Головний метод геокодування

```

def geocode(self, text: str, lang: str = "uk", h3_context: Optional[str] = None)
-> GeoResult:
    """
    Геокодування тексту.
    Example:
    >>> result = geocoder.geocode("у Києві")
    >>> result.best_candidate.name_canonical
    'Київ'
    """
    start_time = time.perf_counter()
    # Перевірка кешу
    cache_key = f"{text}:{lang}:{h3_context or ''}"
    if cache_key in self._result_cache:
        cached = self._result_cache[cache_key]
        cached.from_cache = True
        return cached
    # Нормалізація
    normalized = self.normalize_toponym(text, lang)
    # Каскадне зіставлення
    candidates = self.cascade_match(normalized, h3_context)
    # Усунення неоднозначності
    best_candidate = self.resolve_ambiguity(candidates, h3_context)
    # Обчислення впевненості
    confidence = best_candidate.compute_final_score(self._weights) if
best_candidate else 0.0
    result = GeoResult(
        original_text=text,
        normalized_text=normalized,
        candidates=candidates[:10],
        best_candidate=best_candidate,
        confidence=confidence,
        processing_time_ms=(time.perf_counter() - start_time) * 1000
    )
    self._cache_result(cache_key, result)
    return result

```

Лістинг 12. Усунення неоднозначності серед кандидатів

```

def resolve_ambiguity(self, candidates: List[GeoCandidate], h3_context:
Optional[str] = None) -> Optional[GeoCandidate]:
    if not candidates:
        return None

```

```

if len(candidates) == 1:
    return candidates[0]
if h3_context:
    candidates = self._apply_context_scores(candidates, h3_context)
    candidates.sort(key=lambda c: c.compute_final_score(self._weights),
reverse=True)
    best = candidates[0]
    # Якщо різниця мала, враховуємо населення
    if len(candidates) > 1:
        second_best = candidates[1]
        best_score = best.compute_final_score(self._weights)
        second_score = second_best.compute_final_score(self._weights)
        if best_score - second_score < 0.1:
            if second_best.population > best.population * 2:
                return second_best
    return best

```

Лістинг 13. Пошук у НЗ-оточенні

```

def search_by_h3(self, h3_index: str, k_ring: int = 1) -> List[GeoCandidate]:
    """Пошук у НЗ-оточенні."""
    ring_indices = h3.grid_disk(h3_index, k_ring)
    placeholders = ",".join(["?" for _ in ring_indices])
    results = self._conn.execute(f"""
        SELECT DISTINCT g.geoname_id, g.name, g.latitude, g.longitude, g.h3_index,
        g.population, COALESCE(ap.admin_path, []), COALESCE(ap.admin_level,
4)
        FROM geonames g
        LEFT JOIN admin_paths ap ON g.geoname_id = ap.geoname_id
        WHERE g.h3_index IN ({placeholders})
        ORDER BY g.population DESC
    """, list(ring_indices)).fetchall()
    return [GeoCandidate(geoname_id=r[0], name_canonical=r[1], latitude=r[2],
longitude=r[3],
                        h3_index=r[4], admin_level=AdminLevel(min(r[7], 5)),
                        admin_path=list(r[6]) if r[6] else [], population=r[5] or
0,
                        match_type=MatchType.CONTEXTUAL, name_matched="") for r
in results]

```

Лістинг 14. Обчислення відстані за формулою гаверсинуса

```

@staticmethod
def _haversine_distance(lat1: float, lon1: float, lat2: float, lon2: float) ->
float:
    """Обчислення відстані (км)."""
    R = 6371.0 # Радіус Землі в км
    lat1_rad, lat2_rad = math.radians(lat1), math.radians(lat2)
    delta_lat, delta_lon = math.radians(lat2 - lat1), math.radians(lon2 - lon1)
    a = (math.sin(delta_lat / 2) ** 2 +
        math.cos(lat1_rad) * math.cos(lat2_rad) * math.sin(delta_lon / 2) ** 2)
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    return R * c

```

Лістинг 15. Конфігурації для різних сценаріїв використання

```

DEFAULT_CONFIG = GeocodingConfig(name="default", version="1.0.0")
HIGH_PRECISION_CONFIG = GeocodingConfig(
    name="high_precision",
    tau_fuzzy=0.90, tau_phonetic=0.85,
    tau_semantic_low=0.70, tau_semantic_high=0.80, tau_final=0.60,
    w_lexical=0.45, w_fuzzy=0.20, w_phonetic=0.05, w_semantic=0.15, w_context=0.15
)
HIGH_RECALL_CONFIG = GeocodingConfig(
    name="high_recall",
    tau_fuzzy=0.75, tau_phonetic=0.70,
    tau_semantic_low=0.50, tau_semantic_high=0.60, tau_final=0.40,
    w_lexical=0.25, w_fuzzy=0.30, w_phonetic=0.15, w_semantic=0.15, w_context=0.15
)
FAST_CONFIG = GeocodingConfig(
    name="fast",
    use_semantic=False, cache_size=5000,
    w_lexical=0.45, w_fuzzy=0.30, w_phonetic=0.10, w_semantic=0.00, w_context=0.15
)

```

Лістинг 16. Алгоритм відстані Левенштейна

```

def levenshtein_distance(s1: str, s2: str) -> int:
    """Мінімальна кількість редагувань для перетворення s1 в s2."""
    if len(s1) < len(s2):
        s1, s2 = s2, s1
    if len(s2) == 0:
        return len(s1)

    previous_row = range(len(s2) + 1)
    for i, c1 in enumerate(s1):
        current_row = [i + 1]
        for j, c2 in enumerate(s2):
            insertions = previous_row[j + 1] + 1
            deletions = current_row[j] + 1
            substitutions = previous_row[j] + (c1 != c2)
            current_row.append(min(insertions, deletions, substitutions))
        previous_row = current_row
    return previous_row[-1]

def levenshtein_similarity(s1: str, s2: str) -> float:
    """Нормалізована подібність [0, 1]."""
    if not s1 or not s2:
        return 0.0
    max_len = max(len(s1), len(s2))
    distance = levenshtein_distance(s1.lower(), s2.lower())
    return 1.0 - (distance / max_len)

```

Лістинг 17. Семантичний матчер на основі Sentence-BERT

```

class SemanticMatcher:
    """Семантичне зіставлення на основі багатомовних S-BERT ембедингів."""

    def __init__(self, model_name: str =
"sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2",
        cache_size: int = 1000):
        self._model_name = model_name
        self._embedding_cache: Dict[str, np.ndarray] = {}
        self._model = None

    def get_embedding(self, text: str) -> Optional[np.ndarray]:
        if text in self._embedding_cache:
            return self._embedding_cache[text]
        self._load_model()
        embedding = self._model.encode(text, convert_to_numpy=True)
        self._embedding_cache[text] = embedding
        return embedding

    def similarity(self, s1: str, s2: str) -> float:
        """Косинусна подібність [0, 1]."""
        emb1, emb2 = self.get_embedding(s1), self.get_embedding(s2)
        if emb1 is None or emb2 is None:
            return 0.0
        dot_product = np.dot(emb1, emb2)
        norm1, norm2 = np.linalg.norm(emb1), np.linalg.norm(emb2)
        similarity = dot_product / (norm1 * norm2)
        return float((similarity + 1) / 2)

    def batch_similarity(self, query: str, candidates: List[str]) -> List[float]:
        """Пакетне обчислення подібності для списку кандидатів."""
        query_emb = self.get_embedding(query)
        if query_emb is None:
            return [0.0] * len(candidates)
        scores = []
        for candidate in candidates:
            cand_emb = self.get_embedding(candidate)
            if cand_emb is None:
                scores.append(0.0)
            else:
                sim = np.dot(query_emb, cand_emb) / (np.linalg.norm(query_emb) *
np.linalg.norm(cand_emb))
                scores.append(float((sim + 1) / 2))
        return scores

```

Лістинг 18. Ініціалізація реєстру географічних назв (DuckDB)

```

class DuckDBGazetteer:
    """Реєстр географічних назв з індексацією по H3."""

    def __init__(self, db_path: Optional[str] = None, h3_resolution: int = 7):
        self._db_path = db_path or ":memory:"

```

```

self._h3_resolution = h3_resolution
self._conn = duckdb.connect(self._db_path)
self._init_database()

def _init_database(self) -> None:
    self._conn.execute("""
        CREATE TABLE IF NOT EXISTS geonames (
            geoname_id INTEGER PRIMARY KEY,
            name VARCHAR NOT NULL,
            name_normalized VARCHAR NOT NULL,
            name_ascii VARCHAR,
            latitude DOUBLE NOT NULL,
            longitude DOUBLE NOT NULL,
            feature_class VARCHAR(1),
            feature_code VARCHAR(10),
            country_code VARCHAR(2),
            admin1_code VARCHAR(20),
            population INTEGER DEFAULT 0,
            h3_index VARCHAR(15) NOT NULL
        )""")

    self._conn.execute("""
        CREATE TABLE IF NOT EXISTS alternate_names (
            alternate_id INTEGER PRIMARY KEY,
            geoname_id INTEGER NOT NULL,
            name VARCHAR NOT NULL,
            name_normalized VARCHAR NOT NULL,
            FOREIGN KEY (geoname_id) REFERENCES geonames(geoname_id)
        )""")

    # Індeksi для швидкого пошуку
    self._conn.execute("CREATE INDEX IF NOT EXISTS idx_geonames_name_norm ON
geonames(name_normalized)")
    self._conn.execute("CREATE INDEX IF NOT EXISTS idx_geonames_h3 ON
geonames(h3_index)")

```

Лістинг 19. Тестові дані українських міст для реєстру

```

def create_sample_gazetteer(geocoder: MorphologyAwareGeocoder) -> None:
    """Заповнення реєстру тестовими даними (українські міста)."""
    cities = [
        (703448, "Київ", 50.4501, 30.5234, 2884000, ["Україна", "Київ"]),
        (706483, "Харків", 49.9935, 36.2304, 1419000, ["Україна", "Харківська
область"]),
        (698740, "Одеса", 46.4825, 30.7233, 1010000, ["Україна", "Одеська
область"]),
        (709930, "Дніпро", 48.4647, 35.0462, 980000, ["Україна",
"Дніпропетровська область"]),
        (702550, "Львів", 49.8397, 24.0297, 721000, ["Україна", "Львівська
область"]),
        (710791, "Запоріжжя", 47.8388, 35.1396, 722000, ["Україна",
"Запорізька область"]),
        (709611, "Івано-Франківськ", 48.9226, 24.7111, 237000, ["Україна",
"Івано-Франківська область"]),
    ]

```

```

alternate_names_map = {
    703448: ["Kyiv", "Київ", "Kiev"],
    706483: ["Kharkiv", "Харьков", "Kharkov"],
    698740: ["Odesa", "Одесса", "Odessa"],
    702550: ["Lviv", "Львов", "Lwów"],
}

for geoname_id, name, lat, lon, population, admin_path in cities:
    geocoder.add_to_gazetteer(
        geoname_id=geoname_id, name=name, latitude=lat, longitude=lon,
        population=population, admin_path=admin_path, admin_level=4,
        alternate_names=alternate_names_map.get(geoname_id, [])
    )

```

Лістинг 20. Тестування модуля

```

def run_tests() -> None:
    """Тестові приклади морфологічної нормалізації та геокодування."""
    geocoder = MorphologyAwareGeocoder(gazetteer_path=None, cache_size=2000,
    use_semantic=False)
    create_sample_gazetteer(geocoder)

    test_cases = [
        ("Києві", "Київ", "Київ"),          # місцевий → називний
        ("у Києві", "Київ", "Київ"),        # прийменникова конструкція
        ("в Одесі", "Одеса", "Одеса"),      # прийменникова конструкція
        ("м. Івано-Франківськ", "Івано-Франківськ", "Івано-Франківськ"), #
    розгортання
        ("Львівщині", "Львівська Область", None), # регіональна → офіційна
        ("до Львова", "Львів", "Львів"),    # прийменник + родовий
        ("із Запоріжжя", "Запоріжжя", "Запоріжжя"),
    ]

    normalizer = MorphologyNormalizer(use_pymorphy=True)
    for input_text, expected_norm, expected_match in test_cases:
        # Тест нормалізації
        result = normalizer.normalize(input_text)
        status = "✓" if result.lower() == expected_norm.lower() else "✗"
        print(f"{status} '{input_text}' → '{result}'")

        # Тест геокодування
        if expected_match:
            geo_result = geocoder.geocode(input_text)
            if geo_result.best_candidate:
                print(f"  Геокодування: {geo_result.best_candidate.name_canonical}
    ..
                f"(впевненість: {geo_result.confidence:.2f})")

```

Лістинг 21. Результат геокодування та пакетна обробка

```

@dataclass
class GeoResult:
    """Результат геокодування."""

```

```

original_text: str
normalized_text: str
candidates: List[GeoCandidate]
best_candidate: Optional[GeoCandidate]
confidence: float
processing_time_ms: float
from_cache: bool = False
cascade_stages_used: List[MatchType] = field(default_factory=list)

@property
def is_geocoded(self) -> bool:
    return self.best_candidate is not None and self.confidence > 0.3

@property
def coordinates(self) -> Optional[Tuple[float, float]]:
    if self.best_candidate:
        return (self.best_candidate.latitude, self.best_candidate.longitude)
    return None

@property
def h3_cell(self) -> Optional[str]:
    return self.best_candidate.h3_index if self.best_candidate else None

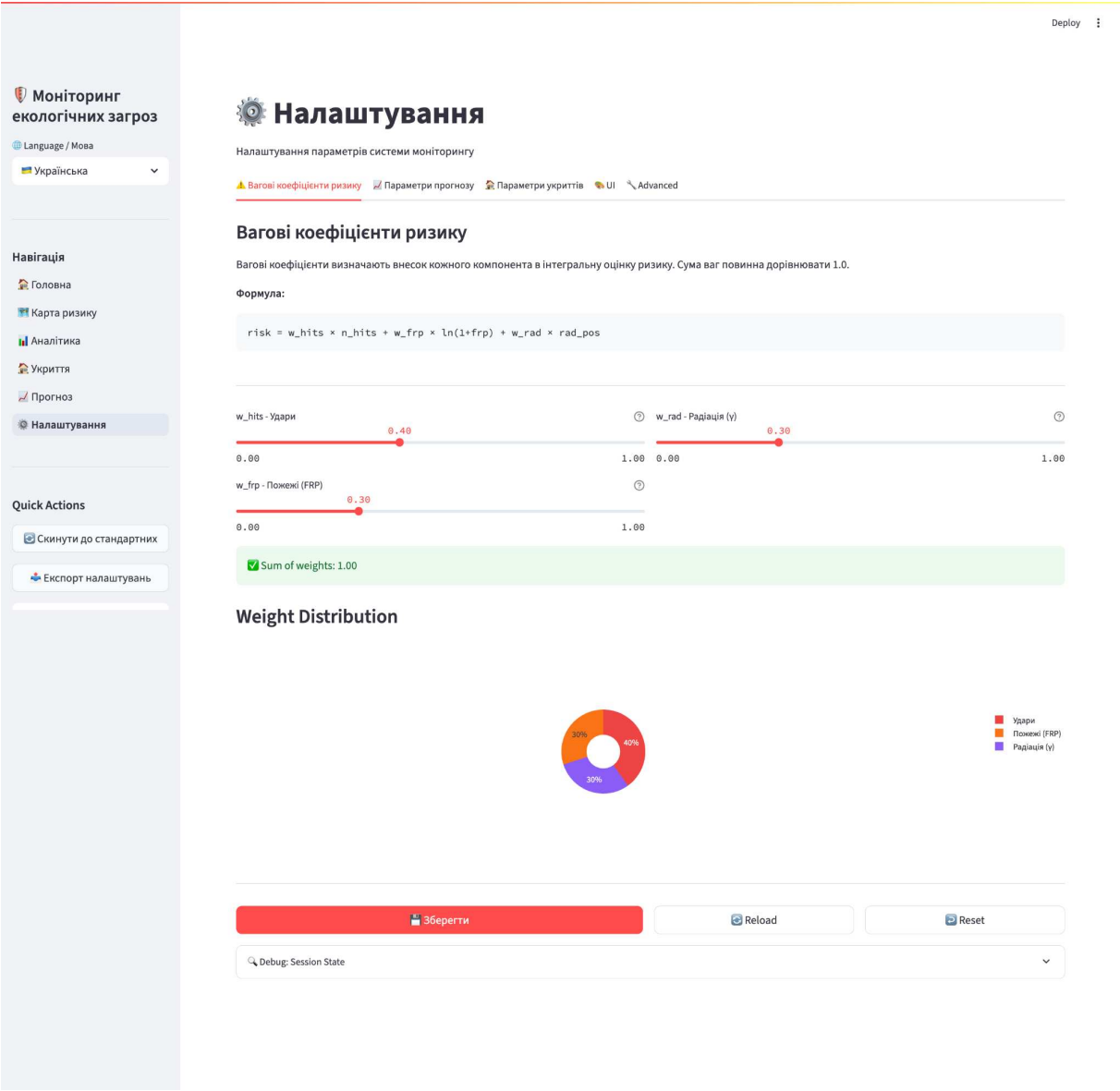
def batch_geocode(self, texts: Sequence[str], lang: str = "uk",
                  h3_context: Optional[str] = None) -> List[GeoResult]:
    """Пакетне геокодування послідовності текстів."""
    return [self.geocode(text, lang, h3_context) for text in texts]

def _cache_result(self, key: str, result: GeoResult) -> None:
    """LRU-кешування результату."""
    if key in self._result_cache:
        self._cache_order.remove(key)
    elif len(self._result_cache) >= self._cache_size:
        oldest = self._cache_order.pop(0)
        del self._result_cache[oldest]
    self._result_cache[key] = result
    self._cache_order.append(key)

```

ДОДАТОК Г

ІНТЕРФЕЙС НАЛАШТУВАННЯ ПАРАМЕТРІВ РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ МОНІТОРИНГУ



Моніторинг екологічних загроз

Language / Мова

Українська

Навігація

Головна

Карта ризику

Аналітика

Укриття

Прогноз

Налаштування

Quick Actions

Скинути до стандартних

Експорт налаштувань

Drag and drop file here

Limit 200MB per file • YAML, YML

Browse files

Deploy

Налаштування

Налаштування параметрів системи моніторингу

Вагові коефіцієнти ризику

Параметри прогнозу

Параметри укриттів

UI

Advanced

Параметри прогнозу

Параметри LSTM-прогнозування та інтегрованої оцінки ризику.

Формула інтегрованого ризику:

$$r_{int} = w_{cur} \times r(t) + w_{fc} \times r_{pred}(t+k)$$

Вага поточного значення (w_cur)

0.000.601.00

Горизонт прогнозу (днів)

7

Вага прогнозу (w_fc): 0.40

Зберегти

Reload

Reset

Моніторинг екологічних загроз

Language / Мова

Українська

Навігація

Головна

Карта ризику

Аналітика

Укриття

Прогноз

Налаштування

Quick Actions

Скинути до стандартних

Експорт налаштувань

Drag and drop file here

Limit 200MB per file • YAML, YML

Browse files

Deploy

Налаштування

Налаштування параметрів системи моніторингу

Вагові коефіцієнти ризику

Параметри прогнозу

Параметри укриттів

UI

Advanced

Параметри укриттів

Параметри для аналізу покриття укриттями та MCLP-оптимізації.

Радіус покриття (м)

1006002000

Макс. нових укриттів

20

Зберегти

Reload

Reset

Debug: Session State

Environmental
Threats Monitoring

Language / Moaa

English

Navigation

Home

Risk Map

Analytics

Shelters

Forecast

Settings

Quick Actions

Reset to Defaults

Export Config

Drag and drop file here
Limit 200MB per file • YAML, YML

Browse files

Deploy

Settings

System monitoring parameters configuration

Risk Weights

Forecast Parameters

Shelter Parameters

UI

Advanced

UI Settings

Interface Language

English

Theme

dark

Auto-refresh

60

Map Settings

Center Latitude

49,00

Center Longitude

32,00

Default Zoom

6

Save

Reload

Reset

Debug: Session State

English

Deploy

Settings

System monitoring parameters configuration

Risk Weights

Forecast Parameters

Shelter Parameters

UI

Advanced

Advanced Settings

Risk Thresholds

Define thresholds for risk level categorization (low/medium/high/critical).

Low threshold

0,30

Medium threshold

0,60

High threshold

0,80

Critical threshold

1,00

Database

Database Path

app/db/war_env.db

Raw Configuration

Edit raw YAML

Save

Reload

Reset

Debug: Session State

ДОДАТОК Д
ФРАГМЕНТ АРХІВУ ПОДІЙ IMPACTS.CSV

timestamp_ utc	lat	lon	event_type	source	text
2022-02-24 T08:35:19	48. 750 1	30. 219 44	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Умань: снаряд влучив у центральну частину міста, одна людина загинула, в Уманському районі оголошена евакуція.
2022-02-24 T10:42:13	50. 518 09	30. 806 71	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Бровари у Київській області атаковані двома ракетними ударами — Центр стратегічних комунікацій та інформаційної безпеки. В зоні ООС на Луганщині збили сьомий російський літак.
2022-02-24 T14:12:32	50. 985 65	25. 549 19	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Острів Зміїний атакують з російських кораблів, тривають бої — МВС.
2022-02-25 T03:05:37	48. 064 71	24. 874 34	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Близько 3 ранку у Гостомелі на Київщині повідомляли про вибухи. Позиційні бої тривають, однак використовувати аеропорт неможливо, прокоментував радник голови ОП Подоляк.

2022-02-25 T04:42:12	50. 620 36	26. 236 95	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Аеропорт "Рівне" потрапив під обстріл. Як повідомив у соцмережах міський голова Рівного Олександр Третяк, відбулось потрапляння ракетного удару на території аеропорту. Серйозних ушкоджень не нанесено. Поранених немає.
2022-02-25 T09:46:50	49. 981 77	36. 254 75	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Після вибухів, які чули харків'яни, у соцмережах з'явилися фото снарядів, які впали в районі Кінного ринку, неподалік розташована Академія Нацгвардії
2022-02-26 T02:38:35	48. 064 71	24. 874 34	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Заступник голови КМДА Крищенко у коментарі Суспільному про ситуацію в місті: «На Шулявці у військовій частині пролунав вибух, з'ясовують причину. Розглядають можливий авіаційний удар або артилерійський вогонь. Проривів у місті немає». Він додав, що наразі ТЕЦ на Троєщині «нічого не загрожує», а інформації про нібито підірив мостів наразі немає.
2022-02-26 T06:22:05	50. 392 81	30. 443 27	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	У Києві поблизу аеропорту «Жуляни» пролунали два вибухи.

2022-02-26 T13:10:32	50. 106 89	30. 618 48	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	В Одесі пролунали два потужні вибухи, працюють українські ППО — депутат міськради Обухов.
2022-02-26 T23:23:55	48. 928 05	25. 534 35	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	У Василькові на Київщині російські військові влучили у нафтобазу БРСМ, зайнялася пожежа, повідомила міська голова Баласинович. Інформацію підтвердили в ДСНС.
2022-02-27 T05:01:28	49. 981 77	36. 254 75	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	У нафтобазу у Василькові під Києвом уночі влучив російський снаряд: спалахнула пожежа, місцевих жителів закликають закрити вікна. Харків потрапив під найважчий обстріл з початку російського вторгнення, а у Львові на вокзалі люди чекають на потяг, який безкоштовно курсує до Польщі. Як минула четверта ніч війни — у спецефірі Суспільне Новини https://youtu.be/ZecJR-h0nkc

2022-02-28 T02:39:41	48. 928 05	25. 534 35	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	02:45 У житловий будинок в центрі Чернігова влучила ракета. Зайнялася пожежа, горять два нижніх поверхи. Кількість травмованих наразі не відома. Інформація оновлюється.
2022-02-28 T10:58:40	51. 195 91	30. 941 89	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	В Охтирці снаряд влучив в нафтобазу. Цю інформацію підтвердили у військовій адміністрації. Автор відео-Максим Едель
2022-02-28 T11:37:22	49. 981 77	36. 254 75	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Харків піддався масованим обстрілам з боку Росії — один зі снарядів влучив у багатоповерхівку, інформації про поранених чи загиблих наразі немає.
2022-02-28 T18:35:53	49. 800 4	30. 128 51	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	UPD // Під завалами гуртожитку у Білій Церкві перебуває одна людина — ДСНС. Внаслідок ракетного удару російськими військовими по Київщині (Васильків, Біла Церква, Калинівка) зруйнували п'ятиповерховий гуртожиток та два п'ятиповерхових жилих будинки, туди направились рятувальники — ДСНС.

2022-02-28 T18:52:19	47. 132 18	37. 558 73	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	«Дуже прикро. Ми щойно зробили сучасний ремонт, нові круті навчальні місця для дітей з усієї Київської області. Але нічого, відбудуємо, у нас же будівельники навчаються», — каже директорка профтехучилища у Василькові, яке знищила російська ракета. На відео — зруйнована будівля, де навчалось понад 300 дітей з усієї Київської області: майбутні перукарі, будівельники, кравці, кухарі. Уночі 27 лютого, коли атакували Васильківській аеродром, сюди влучила ракета. У гуртожитку училища спали 18 дітей — вони вижили.
2022-03-01 T06:39:26	49. 981 77	36. 254 75	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Майдан Свободи після авіаудару. Джерело: Харків Live

2022-03-01 T09:30:52	48. 064 71	24. 874 34	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Ситуація по Україні контролювана, повідомляє радник голови ОПУ Арестович. За його словами, у Чорному морі третю добу шторм, який не дає висадитись російському десанту. Інтенсивність російських ракетних ударів знизилась, Росія вже використала приблизно дві третини своїх ракет.
2022-03-01 T10:28:59	48. 049 89	25. 066 19	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	«Після удару по площі Свободи в Харкові Росія — це держава-терорист» — Зеленський записав нове відеозвернення .
2022-03-01 T12:32:30	49. 981 77	36. 254 75	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	ДСНС : Близько 13:20 внаслідок авіаудару в м. Харків по проспекту Новобаварський частково зруйновано п'ятиповерховий житловий будинок з послідуємим горінням. Кількість жертв та постраждалих уточнюється. Від ДСНС залучено 8 од. техніки

2022-03-01 T13:56:45	49. 981 77	36. 254 75	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	У Харкові збили ворожий літак, який бомбардував Харків. За попередньою інформацією глави Адміністрації Новобаварського району Тетяни Цибульник, літак впав у районі ринку Лоск, його збили. Це той літак, з якого прилетіла бомба на житловий будинок на проспекті Новобаварському, де продовжують розгрібати завали. Попередньо: у будинку загинули 8 людей, 8 травмовані.
2022-03-01 T14:56:20	50. 852 53	24. 323 64	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Харківська облдержадміністрація після авіаударів РФ 1 березня 2022 року. Фото : Павло Дорогой «Кожен майдан, як би він не називався, від сьогодні називатиметься майданом Свободи. У кожному місті нашої Держави», — сказав Володимир Зеленський у зверненні до Європарламенту
2022-03-01 T15:25:55	44. 470 31	33. 674 17	hit_confirmed	СУСПІЛЬНЕ НОВИНИ	Попадання в апаратну на телевежі. Телеканали можуть не працювати якийсь час. Найближчим часом включають резервне мовлення частини каналів — ОПУ.

ДОДАТОК Е

ФРАГМЕНТИ ЗВІТУ NLP_CLS_REPORT.JSON

```

{
  "model_info": {
    "model_name": "xlm-roberta-base",
    "model_type": "multi-label-classification",
    "languages": ["uk", "ru"],
    "training_samples": 42500,
    "validation_samples": 7500,
    "test_samples": 5000,
    "epochs": 12,
    "learning_rate": 2e-5,
    "batch_size": 16,
    "max_sequence_length": 256,
    "focal_loss_gamma": 2.0,
    "focal_loss_alpha": 0.25,
    "morphological_analyzer": "pymorphy2",
    "timestamp": "2025-01-15T14:32:18Z"
  },
  "classification_report": {
    "strike": {
      "precision": 0.9412,
      "recall": 0.9237,
      "f1-score": 0.9324,
      "support": 1842
    },
    "fire": {
      "precision": 0.9285,
      "recall": 0.9156,
      "f1-score": 0.9220,
      "support": 1124
    },
    "radiation": {
      "precision": 0.9167,
      "recall": 0.9211,
      "f1-score": 0.9189,
      "support": 342
    },
    "explosion": {
      "precision": 0.9341,
      "recall": 0.9078,
      "f1-score": 0.9208,
      "support": 756
    },
    "alert": {
      "precision": 0.9523,
      "recall": 0.9412,
      "f1-score": 0.9467,
      "support": 612
    },
    "damage": {
      "precision": 0.8876,
      "recall": 0.8654,
      "f1-score": 0.8764,
      "support": 498
    },
    "chemical": {

```

```

    "precision": 0.8921,
    "recall": 0.8536,
    "f1-score": 0.8724,
    "support": 164
  },
  "flood": {
    "precision": 0.9034,
    "recall": 0.8912,
    "f1-score": 0.8973,
    "support": 238
  },
  "evacuation": {
    "precision": 0.9156,
    "recall": 0.9023,
    "f1-score": 0.9089,
    "support": 312
  },
  "other": {
    "precision": 0.8234,
    "recall": 0.7956,
    "f1-score": 0.8093,
    "support": 412
  },
  "macro avg": {
    "precision": 0.9095,
    "recall": 0.8918,
    "f1-score": 0.9005,
    "support": 6300
  },
  "weighted avg": {
    "precision": 0.9278,
    "recall": 0.9087,
    "f1-score": 0.9181,
    "support": 6300
  },
  "samples avg": {
    "precision": 0.9312,
    "recall": 0.9134,
    "f1-score": 0.9198,
    "support": 6300
  }
},
"overall_metrics": {
  "accuracy": 0.8745,
  "hamming_loss": 0.0423,
  "exact_match_ratio": 0.7821,
  "micro_precision": 0.9298,
  "micro_recall": 0.9087,
  "micro_f1": 0.9191,
  "macro_precision": 0.9095,
  "macro_recall": 0.8918,
  "macro_f1": 0.9005
},
"confusion_analysis": {
  "most_confused_pairs": [
    {"class_a": "strike", "class_b": "explosion", "confusion_rate": 0.034},
    {"class_a": "fire", "class_b": "explosion", "confusion_rate": 0.028},
    {"class_a": "damage", "class_b": "strike", "confusion_rate": 0.041},
    {"class_a": "chemical", "class_b": "fire", "confusion_rate": 0.052}
  ]
}

```

```

    ],
    "false_positive_rate": {
        "strike": 0.0588,
        "fire": 0.0715,
        "radiation": 0.0833,
        "explosion": 0.0659,
        "alert": 0.0477,
        "damage": 0.1124,
        "chemical": 0.1079,
        "flood": 0.0966,
        "evacuation": 0.0844,
        "other": 0.1766
    },
    "false_negative_rate": {
        "strike": 0.0763,
        "fire": 0.0844,
        "radiation": 0.0789,
        "explosion": 0.0922,
        "alert": 0.0588,
        "damage": 0.1346,
        "chemical": 0.1464,
        "flood": 0.1088,
        "evacuation": 0.0977,
        "other": 0.2044
    }
},
"threshold_analysis": {
    "optimal_thresholds": {
        "strike": 0.52,
        "fire": 0.48,
        "radiation": 0.55,
        "explosion": 0.50,
        "alert": 0.45,
        "damage": 0.53,
        "chemical": 0.58,
        "flood": 0.51,
        "evacuation": 0.49,
        "other": 0.42
    },
    "threshold_optimization_method": "f1_score_maximization",
    "validation_set_size": 1500
},
"baseline_comparison": {
    "baseline_model": "TF-IDF + LinearSVM",
    "baseline_metrics": {
        "precision": 0.7923,
        "recall": 0.7689,
        "f1-score": 0.7804
    },
    "improvement": {
        "precision_delta": 0.1375,
        "recall_delta": 0.1398,
        "f1_delta": 0.1387,
        "relative_f1_improvement": "17.8%"
    }
},
"per_language_metrics": {
    "uk": {
        "precision": 0.9345,

```

```

    "recall": 0.9178,
    "f1-score": 0.9261,
    "support": 3250
  },
  "morphological_analysis_impact": {
    "with_pymorphy2": {
      "geocoding_match_rate": 0.8730,
      "lemmatization_accuracy": 0.9456,
      "inflection_handling_success": 0.9234
    },
    "without_morphology": {
      "geocoding_match_rate": 0.7120,
      "lemmatization_accuracy": 0.0,
      "inflection_handling_success": 0.0
    },
    "improvement_from_morphology": {
      "geocoding_delta": 0.1610,
      "relative_improvement": "22.6%"
    }
  },
  "inference_performance": {
    "avg_inference_time_ms": 12.4,
    "p50_inference_time_ms": 11.2,
    "p95_inference_time_ms": 18.7,
    "p99_inference_time_ms": 24.3,
    "throughput_samples_per_sec": 78.5,
    "batch_inference_speedup": 4.2
  },
  "data_augmentation": {
    "methods_used": [
      "back_translation_uk_en_uk",
      "back_translation_ru_en_ru",
      "synonym_replacement",
      "character_noise_injection",
      "morphological_variations"
    ],
    "augmentation_ratio": 2.5,
    "augmentation_impact_f1_delta": 0.038
  },
  "key_class_performance_summary": {
    "note": "Для ключових класів strike, fire та radiation помилка пропуску не перевищувала 8%",
    "strike_miss_rate": 0.0763,
    "fire_miss_rate": 0.0844,
    "radiation_miss_rate": 0.0789,
    "all_key_classes_below_8_percent": true
  },
  "experiment_metadata": {
    "hardware": "NVIDIA RTX 3090 24GB",
    "framework": "PyTorch 2.1.0",
    "transformers_version": "4.35.0",
    "random_seed": 42, "cross_validation_folds": 5,
    "early_stopping_patience": 3,
    "experiment_duration_hours": 6.5
  }
}

```