

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

На правах рукопису

Певнев В.І.

УДК 519.8

МЕТОДИ ОПТИМІЗАЦІЇ ІНТЕГРОВАНОЇ СИСТЕМИ
ВІДЕОСПОСТЕРЕЖЕННЯ З ЕЛЕМЕНТАМИ АНАЛІЗУ ДАНИХ

Наукова спеціальність – 172 «Телекомунікація та радіотехніка»

Дисертація на здобуття наукового ступеня доктора філософії

Науковий керівник:
Одарченко Р.С.
доктор. техн. наук, професор

Київ – 2026

АННОТАЦІЯ

Певнев В.І. Методи оптимізації інтегрованої системи відеоспостереження з елементами аналізу даних. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 172 «Телекомунікація та радіотехніка». - Державний Університет «Київський Авіаційний Інститут», Київ, 2026.

Дисертаційну роботу присвячено розв’язанню актуального науково-прикладного завдання, яке полягає у розробці методу багатокритеріального оцінювання ефективності інтегрованої системи відеоспостереження з елементами аналізу даних, що дозволяє визначити інтегральну ефективність системи та оптимізувати її конфігурацію з урахуванням трьох складових параметрів ефективності: технічної, економічної, функціональної.

Необхідність в такому підході, ґрунтується на специфіці сучасних систем безпеки, які розглядається не як окремі системи, а як цілісна система, що складається з підсистем, які не просто здатні взаємодіяти між собою, а є взаємно залежними між собою для ефективного виконання поставлених перед системою задач. Це означає, що система безпеки може бути більш ефективна і розв’язувати складніші та глибші інциденти, або ж автоматизувати бізнес-процеси через модульний підхід до структури програмно-апаратного комплексу, їх інтеграції з програмних забезпеченням верхнього рівня, а також системи скриптів, за якими виконується взаємодія складних підсистем безпеки, та опрацьовується результат їх роботи.

В даній роботі проаналізовано сучасні підходи до оцінювання та оптимізації інтегрованих систем безпеки. Встановлено, що існуючі методи, такі як оцінки та оптимізація за технологічною базою і характеристиками програмно-апаратного комплексу, економічною-фінансовими підходами зі скороченням капітальних витрат та операційних, а також функціональні або експертні, які переважно орієнтовані на окремі

групи показників і не забезпечують комплексний підхід до оцінювання системи. Таким чином неможливо оцінити якість системи до впровадження і прозорий підхід до порівняння альтернативних пропозицій. Обґрунтовано необхідність переходу до багатокритеріального аналізу конфігурації та визначення впливу її складових на різні групи ефективностей.

У дисертаційному дослідженні розроблено новий метод для оцінки ефективності, що включає в себе ряд важливих параметрів з технічних груп, економічних та функціональних. Вперше розроблено та формалізовано математичну модель, яка дозволяє провести розрахунок ефективності з використанням груп параметрів. Вперше розроблено метод оптимізації для таких конфігурацій, які враховують різні показники архітектури, вартості впровадження, якості аналітики та інших вхідних даних проекту. Набули подальшого розвитку програмний прототип розрахунку ефективності, візуалізації значень ефективності та оптимізації базової конфігурації до більшого інтегрального показника.

Теоретичні результати, отримані в дисертаційному дослідженні, відкривають можливість виявити і запропонувати для впровадження систем безпеки їх коректну оцінку та раціональну оцінку альтернативних конфігурацій для зваженого прийняття рішення щодо проектування, впровадження, масштабування та модернізації систем безпеки для використання в комерційних та державних структурах України.

При цьому отримані результати дозволяють: провести детальний аналіз конфігурації базової запропонованої системи безпеки, оцінити її ефективність щодо визначених критеріїв, сформулювати вплив якості роботи аналітичних інструментів на технічну ефективність, вплив на втому оператора (функціональну ефективність) та вплив кількості пропущених інцидентів на збитки операційних затрат (економічну ефективність). Даний підхід дозволяє оцінити слабкі критерії системи та за допомогою методу оптимізації підібрати необхідну оптимальну конфігурацію в заданому необхідному діапазоні вхідних величин.

Практична цінність дисертаційної роботи полягає в наступному: вперше розроблено методику багатокритеріальної оцінки ефективності інтегрованої системи відеоспостереження з елементами аналізу даних; розроблено відповідне алгоритмічне програмне забезпечення, яке застосовується для автоматизованої оцінки таких систем за допомогою використання математичної моделі багатокритеріальної оцінки ефективності і їх оптимізації за показником інтегрального критерія ефективності; матеріали дисертаційної роботи впроваджено в роботу підприємства ТОВ «Аксонсофт», що безпосередньо працює та впроваджує такі системи в Україні та на міжнародному ринку, також результати дисертаційного дослідження впроваджено в навчальний процес Державного Університету «Київського Авіаційнийного Інституту». Використання результатів дисертаційної роботи підтверджено відповідними актами впровадження

Ключові слова: Slice-мережі, динамічне управління, статистична обробка даних, оптимізація структури, системи зв'язку БПЛА, технології SDR, емпіричне вейвлет-перетворення, машинне навчання, IoT пристрої.

Список публікацій здобувача

1. Певнев. В.І. Аналіз можливостей інтегрованих систем безпеки з використанням інструментів відеоаналітики / В.І. Певнев // Політ. Сучасні проблеми науки. Телекомунікації та захист інформації в авіаційному зв'язку: матеріали міжнародної науково-практичної конференції здобувачів вищої освіти і молодих учених (м. Київ, 4-7 квітня 2023 р.). – Київ : НАУ, 2023. – С.60-61.

2. Odarchenko R. Optimization of the integrated video surveillance system with elements of data analysis / R. Odarchenko, V. Pevnev, A. Pinchuk, O. Polihenko // CEUR Workshop Proceedings. Vol. 3925 : Proceedings of the 3rd International Conference on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN 2024), Kyiv, Ukraine, January 24-27, 2024. – 2024. – PP. 47-63.

3. Pinchuk A. Functional dependency graphs for cyber risk assessment in 5G-enabled ICS/OT networks / A. Pinchuk, R. Odarchenko, O. Polihenko, O. Dobrynychuk, V. Pevnev // CEUR Workshop Proceedings. Vol. 4024 : Proceedings of the 4th International Conference on

Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN 2025), Kyiv, Ukraine, June 20-22, 2025. – 2025. – PP. 92-105.

4. Pevnev V. Determining the effectiveness criteria for intelligent detectors in integrated video surveillance systems / V. Pevnev, R. Odarchenko // Electronics and Control Systems. – 2026. – № 1(87). – PP. 64-71

ABSTRACT

Pevnev V.I. Methods for Optimization of an Integrated Video Surveillance System with Elements of Data Analysis. – Qualifying scientific work as a manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 172 Telecommunications and Radio Engineering. – State University Kyiv Aviation Institute, Kyiv, 2026.

The dissertation is devoted to solving an actual scientific and applied problem consisting in the development of a method for multicriteria evaluation of the efficiency of an integrated video surveillance system with elements of data analysis, which makes it possible to determine the integral efficiency of the system and optimize its configuration with due regard to three groups of efficiency parameters: technical, economic, and functional.

The need for such an approach is determined by the specific nature of modern security systems, which should be considered not as separate subsystems but as a holistic system composed of interconnected and interdependent subsystems that interact for the effective fulfillment of the tasks assigned to the overall system. This means that a security system can become more efficient, resolve more complex and deeper incidents, and automate business processes through a modular structure of the hardware and software complex, its integration with higher-level software, and the use of script-based interaction between complex security subsystems with subsequent processing of the obtained results.

The thesis analyzes modern approaches to the evaluation and optimization of integrated security systems. It has been established that existing methods, such as assessments and optimization based on the technological basis and characteristics of the hardware and software complex, economic and financial approaches aimed at reducing capital and operating expenditures, as well as functional or expert-based approaches, are mostly focused on separate groups of indicators and do not provide a comprehensive approach to system evaluation. As a consequence, they do not make it possible to assess system quality before implementation or

to ensure a transparent comparison of alternative proposals. Therefore, the need to move toward a multicriteria analysis of system configuration and to determine the influence of its components on different groups of efficiency indicators has been substantiated.

A new method for efficiency assessment has been developed in the dissertation research. It includes a set of important parameters from the technical, economic, and functional groups. For the first time, a mathematical model has been developed and formalized that enables efficiency calculation using grouped parameters. For the first time, an optimization method has been proposed for such configurations, taking into account different architectural indicators, implementation costs, analytics quality, and other project input data. The software prototype for efficiency calculation, efficiency visualization, and optimization of the baseline configuration toward a higher integral indicator has been further developed.

The theoretical results obtained in the dissertation open up possibilities for correct evaluation of security systems and rational assessment of alternative configurations in order to support balanced decision-making regarding the design, implementation, scaling, and modernization of security systems for use in commercial and governmental sectors of Ukraine.

The obtained results make it possible to perform a detailed analysis of the baseline configuration of the proposed security system, evaluate its efficiency according to the defined criteria, determine the impact of the performance quality of analytical tools on technical efficiency, assess the influence on operator fatigue as a component of functional efficiency, and estimate the effect of missed incidents on operating loss indicators within economic efficiency. This approach makes it possible to identify weak criteria of the system and, by means of the optimization method, select the required optimal configuration within a specified range of input values.

The practical value of the dissertation lies in the following: for the first time, a methodology for multicriteria efficiency assessment of an integrated video surveillance system with elements of data analysis has been developed; appropriate algorithmic software has been created and is used for automated assessment of such systems through the application of a mathematical model of multicriteria efficiency evaluation and their optimization according to

the integral efficiency criterion; the dissertation materials have been implemented in the work of AxxonSoft LLC, which directly develops and deploys such systems in Ukraine and in the international market; in addition, the results of the dissertation research have been implemented in the educational process of the State University Kyiv Aviation Institute. The use of the dissertation results is confirmed by the relevant implementation acts.

Keywords: Slice-networks, dynamic management, statistical data processing, optimization of the structure, UAV communication systems, SDR technologies, empirical wavelet transform, machine learning, IoT devices.

Applicant's publications:

1. Pevnev, V.I. Analysis of the capabilities of integrated security systems using video analytics tools / V.I. Pevnev // *Polit. Modern Problems of Science. Telecommunications and Information Protection in Aviation Communication: Proceedings of the International Scientific and Practical Conference of Higher Education Applicants and Young Scientists* (Kyiv, April 4–7, 2023). – Kyiv: NAU, 2023. – pp. 60–61.
2. Odarchenko, R., Pevnev, V., Pinchuk, A., Polihenko, O. Optimization of the integrated video surveillance system with elements of data analysis // *CEUR Workshop Proceedings*. Vol. 3925: Proceedings of the 3rd International Conference on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN 2024), Kyiv, Ukraine, January 24–27, 2024. – 2024. – pp. 47–63.
3. Pinchuk, A., Odarchenko, R., Polihenko, O., Dobrynychuk, O., Pevnev, V. Functional dependency graphs for cyber risk assessment in 5G-enabled ICS/OT networks // *CEUR Workshop Proceedings*. Vol. 4024: Proceedings of the 4th International Conference on Cyber Hygiene & Conflict Management in Global Information Networks (CH&CMiGIN 2025), Kyiv, Ukraine, June 20–22, 2025. – 2025. – pp. 92–105.

4. Pevnev, V., Odarchenko, R. Determining the effectiveness criteria for intelligent detectors in integrated video surveillance systems // *Electronics and Control Systems*. – 2026. – No. 1(87). – pp. 64–71.

ЗМІСТ

ВСТУП	15
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ ІНТЕГРОВАНИХ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ ТА ПІДХОДІВ ДО ЇХ ПОБУДОВИ ІВПРОВАДЖЕННЯ...	19
1.1. Сучасні інтегровані системи відеоспостереження.....	19
1.2. Архітектури побудови систем відеоспостереження.....	22
1.2.1. Централізована архітектура.....	23
1.2.2. Розподілена архітектура.....	24
1.2.3. Гібридна архітектура	25
1.2.4. EDGE-рішення та NVR	26
1.2.5. Хмарна архітектура та vSaaS	28
1.2.6. Порівняння типів архітектур	29
1.3. Елементи аналізу даних у системах відеоспостереження.....	31
1.3.1. Роль відеоаналітики у сучасних системах та базові аналітичні інструменти.....	32
1.3.2. Нейромережеві методи аналізу даних	34
1.3.3. Вплив точності аналітичних інструментів на ефективність системи	36
1.4. Аналіз існуючих підходів оцінювання системи відеоспостереження	39
1.5. Моделювання функціональних залежностей та каскадних впливів у кіберфізичних системах відеоспостереження	41
1.5.1. Критичний аналіз існуючих аналітичних підходів.....	42
1.5.2. Архітектура та класифікація вузлів у моделях функціональних залежностей.....	43
1.5.3. Типологія функціональних взаємозв'язків (Топологія FDG).....	44
1.5.4. Семантична атрибуція як основа для кількісного оцінювання.....	45
1.6. Класифікація підходів оцінювання за групами критеріїв	46
1.6.1. Технічні критерії	47
1.6.2. Економічні критерії	48

	11
1.6.3. Експлуатаційні критерії	48
1.6.4. Аналітичні критерії	49
1.6.5. Функціональні критерії	49
1.6.6. Ризикові критерії	50
1.7. Постановка наукової проблеми та обґрунтування напряму дослідження	51
Висновки до розділу1.....	54
РОЗДІЛ 2. РОЗРОБКА МЕТОДУ КОМПЛЕКСНОГО ОЦІНЮВАННЯ ТА.ОПТИМІЗАЦІЇ ІНТЕГРОВАНОЇ СИСТЕМИ ВІДЕОСПОСТЕРЕЖЕННЯ З ЕЛЕМЕНТАМИ АНАЛІЗУ ДАНИХ	56
2.1. Формування концепції ефективності інтегрованої системи відеоспостереження....	56
2.1.1. Вектор технічної ефективності	60
2.1.2. Вектор економічної ефективності	60
2.1.3. Вектор функціональної ефективності	62
2.1.4. Математична постановка концепції	63
2.2. Визначення системи критеріїв оцінювання методу ефективності	65
2.2.1. Класифікація критеріїв	66
2.2.2. Опис технічних критеріїв	66
2.2.3. Опис економічних критеріїв	67
2.2.4. Опис функціональних критеріїв	68
2.2.5. Визначення ваги для критеріїв	70
2.2.6. Структура критеріїв та їх поєднання	70
2.3. Побудова інтегрального критерію ефективності	70
2.4 Розробка методу оптимізації	73
2.4.1. Збір даних	76
2.4.2. Фільтрація допустимих рішень	76
2.4.3. Оцінка критеріїв ефективності	77
2.4.4. Інтегральна оцінка ефективності	77
2.4.5. Вибір оптимальної конфігурації	78

	12
2.5. Формалізація правил вибору конфігурації	78
2.5.1. Оцінка критеріїв ефективності	78
2.5.2. Порівняння альтернатив	79
2.5.3. Обмеження на вибір конфігурації	79
2.5.4. Розрахунок і вибір найкращої конфігурації	80
2.5.5. Формалізація правил пріоритетів	80
2.6. Наукова інтерпретація методу	81
2.6.1 Комплексний підхід до оптимізації	81
2.6.2. Ваги критеріїв та гнучкість методу	82
2.6.3. Адаптивність і універсальність методу	82
2.6.4. Математична точність і методологічна коректність	83
2.6.5. Наукове значення методу	83
2.6.6. Вплив на практичне застосування	83
Висновки до розділу 2	85
РОЗДІЛ 3. МАТЕМАТИЧНА МОДЕЛЬ ТА АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ МЕТОДУ ОПТИМІЗАЦІЇ ІНТЕГРОВАНОЇ СИСТЕМИ ВІДЕОСПОСТЕРЕЖЕННЯ З ЕЛЕМЕНТАМИ АНАЛІЗУ ДАНИХ	87
3.1. Формалізація параметрів системи	87
3.1.1. Загальна структура параметричного опису	88
3.1.2. Формалізація технічних параметрів	89
3.1.3. Формалізація економічних параметрів	93
3.1.4. Формалізація функціональних параметрів	95
3.1.5. Узагальнений параметричний простір конфігурацій	98
3.1.6. Зовнішні обмеження та критерії допустимості	98
3.1.7. Значення виконаної формалізації для подальшого дослідження	100
3.2. Математична модель інтегральної ефективності	101
3.2.1. Загальна постановка моделі інтегральної ефективності	101
3.2.2. Локальні функції ефективності	102

	13
3.2.3. Нормування параметрів	104
3.2.4. Система вагових коефіцієнтів	105
3.2.5. Агрегування локальних критеріїв	106
3.2.6. Багатокритеріальна форма моделі	107
3.2.7. Обмеження та штрафна форма інтегральної моделі	107
3.2.8. Інтерпретація інтегрального показника	108
3.3. Математичне врахування впливу модулів аналітики на інтегральну ефективність.	108
3.3.1. Базові показники якості роботи аналітичних інструментів	109
3.3.2. Стохастичний характер помилок аналітики	110
3.3.3. Вплив помилок аналітики на технічну ефективність	112
3.3.4. Вплив хибних спрацювань на навантаження оператора	113
3.3.5. Вплив пропусків інциденту на функціональну та економічну ефективність	114
3.3.6. Скориговані локальні функції ефективності	115
3.3.7. Скоригована інтегральна ефективність з урахуванням аналітики	116
3.3.8. Аналіз чутливості інтегральної ефективності до якості аналітики	117
3.4. Побудова алгоритму вибору раціональної конфігурації	118
3.4.1. Формальна постановка задачі вибору	118
3.4.2. Вхідні дані та формування множини допустимих рішень	119
3.4.3. Розрахунок локальних критеріїв	120
3.4.4. Урахування компонента відеоаналітики	121
3.4.5. Обчислення інтегрального показника та ранжування альтернатив	122
3.4.6. Вибір оптимального рішення та обчислювальна складність алгоритму	123
Висновки до розділу 3	124
РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА ВАЛІДАЦІЯ ЗАПРОПОНОВАНОГО МЕТОДУ ОЦІНКИ ІНТЕГРАЛЬНОЇ ЕФЕКТИВНОСТІ ТА ОПТИМІЗАЦІЇ КОНФІГУРАЦІЇ	126
4.1. Методика та організація експериментального дослідження	126
4.2. Формування вхідних даних і базової конфігурації системи	131

	14
4.3. Демонстраційний розрахунок базової конфігурації	134
4.4. Програмна реалізація методу	141
4.5. Результати оптимізації конфігурації системи	144
4.6. Порівняльний аналіз базової та оптимізованої конфігурації	147
4.7. Порівняльний аналіз існуючих підходів і розробленого методу	154
Висновки до розділу 4	159
ВИСНОВОК	160
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	164
Додаток А.....	175
Додаток Б.....	195
Додаток В.....	196

ВСТУП

Актуальність теми. Системи відеоспостереження є один з ключових елементів забезпечення безпеки, які трансформувались з плином часу в комплексні програмно-апаратні комплекси. Наразі, такі системи, не розглядаються як окреме рішення під задачу відеоспостереження, аналітики, автоматизації, контролю доступу, а розглядаються комплексно, з урахуванням інтеграції всіх підсистем. Саме тому такі системи мають проходити повний багатокритеріальний аналіз для оцінки ефективності.

Існуючі підходи до оцінювання таких конфігурацій здебільшого базуються на окремих групах показників, технічних, економічних, функціональних або експертних. У результаті адекватна, коректна оцінка подібних систем неможлива. Крім відсутності такої оцінки також відсутній розрахунок взаємозв'язку між групами які мають залежність між собою через глобальні критерії впливу, як аналітика, стома оператора, економічні затрати на непрацездатність системи та погану якість її роботи.

У зв'язку з цим актуальним є розробка методу, який дозволить комплексно оцінювати систему через локальні критерії ефективності, а також інтегральний показник, який буде враховувати всі переваги та виключати недоліки існуючих методів. За допомогою такого методу багатокритеріальної інтегральної оцінки відбувається оптимізація конфігурації системи до рівня максимальних показників ефективності відповідно до діапазону необхідних величини та обмежень проекту. Таким чином даний підхід і методи дозволяє вирішити вищезгадані проблематики, які вирішені в рамках даної дисертаційної роботи.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана відповідно до напрямку наукових досліджень в сфері комплексних програмно-апаратних систем безпеки з елементами аналізу даних. Тематика даного дослідження відповідає сучасним завданням щодо підвищення ефективності вибору

системи та відповідно інструментів для впровадження та модернізації КСБ за допомогою методу оптимізації.

Мета роботи. Метою дисертаційної роботи є виконання, відповідно до напрямків наукових досліджень у галузі системи безпеки, відеоспостереження, аналізу даних та інтелектуальних аналітичних детекторів. Тематика дослідження відповідає сучасним завданням оцінки ефективності, підвищення ефективності, а також удосконалення сучасних підходів до проектування через розробку інтегрального методу оцінки ефективності та оптимізації інтегрованих систем, що мають специфічні вимоги до якості, стабільності та масштабування.

Об'єкт дослідження є процеси оцінювання та вибору конфігурації інтегрованих систем відеоспостереження.

Предмет дослідження є методи та моделі оцінювання технічної, економічної й функціональної ефективності інтегрованих систем відеоспостереження з урахуванням впливу модулів відеоаналітики.

Методи дослідження. Для досягнення поставлених цілей в дисертаційній роботі використано: методи системного аналізу, математичного моделювання, багатокритеріального оцінювання, нормування та зваження критеріїв, оптимізаційний пошук, комп'ютерного моделювання.

Наукова новизна. У роботі отримані такі наукові результати:

Удосконалено підхід до оцінки ефективності та процесу оптимізації інтегрованих систем відеоспостереження з елементами аналізу даних, яке полягає в переході від врахування одної групи критеріїв до багатокритеріального аналізу, яка поєднує собі економічний, технічний та функціональний критерій оцінювання та відповідні групи.

Вперше розроблено математичну модель для оцінки інтегрованої системи відеоспостереження, яка заснована на трьох групах ефективності, з яких складається проект ІСВ, а саме технічної, економічної та функціональної. Дані локальні ефективності конфігурації утворюють інтегральний показник ефективності системи.

Вперше розроблено метод оптимізації конфігурації системи для інтегрованих

систем відеоспостереження, яке базується на врахуванні вхідних параметрів базової конфігурації, діапазон можливих змін та системи зовнішніх обмежень, які дозволяють виконати перебір допустимих конфігурацій та виявити найбільш оптимальну через алгоритм оцінки ефективності.

Набули подальшого розвитку програмні засоби розрахунку та прийняття рішення, щодо оцінювання ефективності базової конфігурації та знаходження оптимального аналогу в рамках заданих обмежень, що дозволяє з урахуванням високої швидкості обробки оцінити критерії ефективності та порівняти їх параметри. Такий підхід дозволяє зважено та раціонально прийняти рішення щодо побудови системи та вказати на слабкі місця конфігурації.

Практичне значення отриманих результатів полягає в можливості застосування даних розроблених методів оцінки ефективності та оптимізації конфігурацій систем безпеки, для оцінки, порівняння та вибору раціональної конфігурації для об'єктів різного масштабу та вимог. Запропонований підхід враховує всі необхідні величини, як технічні, економічні та функціональні, а також критерії впливу якості роботи аналітичних детекторів на якість та ефективність роботи системи, прогнозувати очікувані збитки, та оцінювати навантаження на оператора, що дозволяє прийняти правильне раціональне та практичне рішення, щодо використання системи.

Практична цінність. дисертаційної роботи полягає в тому, що розроблений програмний прототип автоматичного розрахунку ефективності конфігурації системи та її оптимізації. Даний прототип побудований на основі розробленого методу оцінки ефективності та оптимізації з врахуванням точності аналітичних інструментів та низки критеріїв системи. Його використання дозволяє швидко та цілісно виконувати оцінку ефективності та доцільності конфігурації ІСВ, а також знаходити підходящі, кращі за значеннями, альтернативи системи. Отримані результати можуть бути використані під час проектування систем, впровадженні, масштабуванні та модернізації, або ж знаходженні слабких місць з точки зору ефективності системи та для чесної оцінки пропозицій.

Особистий внесок автора. Усі результати даної дисертаційної роботи отримані автором самостійно. Автором було проведено аналіз існуючих підходів та методів оцінки та оптимізації інтегрованих систем відеоспостереження, було розроблено новий метод, розроблено вперше математичну модель методу інтегральної оцінки ефективності та оптимізації, а також виконано програмну реалізацію і проведено експериментальне дослідження.

Апробація результатів дисертаційного дослідження.

Основні положення та результати дисертаційної роботи доповідались, обговорювались та апробувались на наукових конференціях, семінарах, фахових лабораторіях та на комерційних проектах. Результати дослідження також використовувались при підготовці наукових публікацій, проведенні теоретичних лекційних та практичних занять в лабораторіях.

Публікації. За матеріалами дисертаційної роботи опубліковано 3 наукових праць, у тому числі: 1 у фаховому виданні, які входять в перелік наукових видань, затверджений МОН України матеріали доповідей на науково-технічних конференціях.

Структура і зміст роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи складає 148с. друкованого тексту, у тому числі містить 27 рисунків та 12 таблиць. Список використаних джерел на 10 сторінках. містить 113 найменувань.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНИХ ІНТЕГРОВАНИХ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ ТА ПІДХОДІВ ДО ЇХ ПОБУДОВИ І ВПРОВАДЖЕННЯ

1.1. Сучасні інтегровані системи відеоспостереження

Існує декілька підходів до побудови систем безпеки та відеоспостереження і до недавнього часу актуальними були системи VMS (Video Managment System), роль яких полягає в вирішенні задач вузького профілю, а саме підключення відеокамер, робота з відеопотоками, аналітикою та зберіганням даних. Враховуючи неповноцінність такого рішення з огляду загальної системи безпеки було створено ряд програмно-апаратних комплексів, що мають назву PSIM (Physical security information management). Такі системи відрізняються як підходом так і задачами, які вона вирішує, оскільки даний тип має мікромодільну архітектуру, що дозволяє вибудувати якісну та ефективну топологію роботи програмного забезпечення, яка складається з ядра системи та його модулів. Кожен модуль в цій системі відповідає за свою задачу та дозволяє не тільки розмежувати їх з точки зору безпеки, а й з точки зору відмовостійкості. У випадку припинення роботи одного модулю система продовжує свою роботу використовуючи наявні інші модулі та ядро системи поки недієздатний модуль не відновить працездатність.

Набір модулів, в таких платформах, групуються для вирішення певного конкретного типу задачі, як розпізнавання обличчя, розпізнавання номерних знаків, додаткова аналітика на основі штучного інтелекту, інтеграція з різним обладнанням, як система контролю доступу, охоронна сигналізація та інше. Таким чином платформа дозволяє об'єднати на програмно-апаратному рівні декілька різних підсистем безпеки та налаштувати їх на сумісну взаємодію від чого такі платформи мають в назві “інтегровані”. Крім можливості інтеграції в даних системах має бути присутня можливість побудови різних автоматичних правил та скриптів, що вимагає наявності

вбудованого інтерпретатора. Саме за допомогою такого інструменту утворюється можливість побудови різнорідних сценаріїв взаємодії для різних підсистем в одному інформаційному середовищі.

Такий підхід зумовлений запитом на автоматизацію та спрощення роботи спеціалістів служби безпеки та ІТ-служб на різних підприємствах, де вибудований послідовний, логічний сценарій роботи, який може виконувати не людина, а система автоматично. Яскравим прикладом може бути пропускна система з контролем вивантаження сировини на елеваторі, де об'єм робочих задач достатньо великий і рівень відповідальності такий самий. Побудова даної системи дозволяє не тільки зменшити кількість помилок, які робить одна або декілька людей на різних етапах, а й скоротити число таких працівників. В окремих випадках є можливість взагалі замінити людський персонал на автоматизовану систему, оскільки крім випадкових ситуацій де інформація чи подія може бути втрачена або спотворена, існують випадки навмисної недоброчесної поведінки працівників, які впливають на саме підприємство. В такому випадку, сценарій взаємодії компонентів системи безпеки буде наступний:

1. Автомобіль під'їжджає до Контрольно Пропускного Пункту;
2. Система розпізнавання номерних знаків визначає номер транспортного засобу;
3. Розпізнаний номер ТЗ порівнюється з еталонним номером, що записаний в списку доступу БД та визначається рівень допуску, якщо такий було надано;
4. Для додаткової верифікації може додатково використовуватись система контролю та управління доступом, яка після успішного завершення п.2 дозволить провести аутентифікацію водія через персональну карту;
5. Система отримує детальну інформацію про ТЗ, водія, сировину та інші параметри і друкує їх автоматично на принтері разом з інструкцією для водія та пам'яткою;

6. Світлофор сигналізує відповідним кольором про можливість проїзду. На інформаційне табло виводиться наступна точка до якої має рухатись водій та автоматично, по команді системи, відкривається шлагбаум;

7. Після проїзду на першу точку - ваговий комплекс, відбувається автоматичне зважування. Система безпеки отримує результати зважування та перевіряє чи допустима така вага у врахуванням похибки;

8. Наступна точка - лабораторія. На цьому етапі відбувається перевірка сировини і визначається наступна точка для маршруту;

9. За результатами лабораторії автомобіль рухається на вказаний комплекс та виконує розвантаження;

10. Після завершення процедури автомобіль повторює всі пункти крім лабораторії в зворотному напрямку, та всі дані фіксуються в обліковій системі. Після завершення процесу водій може залишити територію.

Даний приклад був описаний для того, аби показати який великий об'єм роботи виконується людьми і навіть з використанням різних систем безпеки це не мінімізує ризики. Лише правильний підхід до автоматизації і побудови логіки дозволяє побудувати робочу систему з високим індексом ефективності.

В даному прикладі було використані наступні підсистеми, які були інтегровані в систему відеоспостереження, як:

1. Система відеоспостереження;
2. Аналітика та розпізнавання номерних знаків ТЗ;
3. Інтеграція з системою контролю та управління доступу;
4. Інтеграція з контролером шлагбаума;
5. Робота зі сторонніми світловим елементами;
6. Інтеграція з ваговим комплексом;
7. Інтеграція з програмним забезпеченням лабораторії;
8. Інтеграція з обліковою системою;

9. Програмний інтерпретатор для написання автоматичних правил роботи системи та обробки великих обсягів даних (Big Data).

Висвітлений підхід яскраво демонструє, чому інтегровані системи безпеки та відеоспостереження стали актуальнішими за VMS-подібні системи. Для побудови таких комплексних систем необхідно використовувати спеціалізоване програмне забезпечення, а також сервери. Окремі компанії дозволяють розгорнути системи такого масштабу з використанням спеціалізованих пристроїв - відеореєстраторів (NVR), але проблематика таких рішень полягає в монобрендовій політиці компаній виробників, тобто допускається робота суто з обладнання одного й того ж вендору (виробника), що не дозволяє гнучко масштабувати систему та впроваджувати різного роду специфічні задачі.

Беручи до уваги такий підхід можна зазначити, що не правильно розглядати сучасну систему відеоспостереження, як систему з набором камер, базовими детекторами або інтеграціями. Сучасна інтегрована система відеоспостереження - це складний комплекс в якому потрібно врахувати достатньо велику низку аспектів, як при впровадженні, так і в експлуатації рішення, оскільки рішення перед впровадженням має прямий вплив на якість роботи системи в подальшому і впливає на її вартість володіння.

1.2. Архітектури побудови систем відеоспостереження

Вибір архітектури або ж топології системи безпеки - це критично важливе питання, яке впливає на технологічну та економічну доцільність. Сучасна тенденція демонструє ігнорування даного підходу при побудові систем будь-якого масштабу своєчасно, що формує проблематику в майбутньому з якою зіштовхнеться кожен об'єкт раніше чи пізніше. Така тенденція зумовлена підходом до вирішення задачі - "тут і зараз", через що до не береться до уваги подальша імплементація системи в інші структури або ж інтеграція з іншими підсистемами замовника на об'єкті. Як результат вдаються до використання та придбання додаткових рішень, заміни обладнання, через

відсутність в поточній конфігурації виконати якісне масштабування. В окремих випадках єдиним варіантом є повна заміна ІСВ, тому, першочергово, особливу увагу потрібно приділи вибору топології, яка буде відповідати вирішенню поточних задач та дозволить коректно розвивати систему в майбутній експлуатації.

1.2.1. Централізована архітектура

Централізована архітектура розгортається за принципом топології «зірка», де всі виконавчі пристрої, відеокамери, підсистеми, модулі та сервіси підключаються до одного центрального вузла, який може бути реалізований у вигляді окремого серверу або ж бути кластером з декількох серверів (рис.1.1). Такий підхід найбільш актуальний та зручний на об'єктах де відсутнє географічна розподіленість об'єкту, тобто пристрої знаходяться в межах однієї або декількох будівель. Реалізація такої архітектури потребує використання спеціалізованого професійного обладнання та компонентів, які мають високу вартість відносно стандартних рішень, але водночас - надійними і високопродуктивними.



Рис.1.1. Централізована архітектура

Переваги централізованої архітектури:

- Централізоване адміністрування системи;
- Спрощення виконання технічних робіт з обслуговування;

- Доступне серверне приміщення відповідно норм, що в свою чергу реалізовує ефективне взаємне розташування обладнання безпеки, системи охолодження, пристроїв резервування живлення;
- Зниження навантаження на моніторинг працездатності апаратних компонентів;
- Зменшення витрат та використання професійного апаратних та програмних компонентів;
- Зручність масштабування системи;
- Зменшення витрат на обслуговування.

Недоліки централізованої архітектури:

- Залежність роботи повної системи від центрального вузла;
- За відсутності резервування формується єдина точка відмови;
- Підвищені вимоги до мережевої інфраструктури, пропускної здатності, обладнання та побудови серверного приміщення;
- Підвищені вимоги до навантаження на сервери та електроживлення;
- Відсутність гнучкості в умовах географічно віддалених об'єктів;
- Зростання витрат на модернізацію, ремонт та обслуговування апаратного забезпечення.

1.2.2. Розподілена архітектура

Розподілена архітектура не має центрального вузла, натомість конфігурація системи зберігається на всіх вузлах одночасно. Для таких систем вводиться поняття “Мастер-база”, що визначає сервер на якому зберігається дані конфігурації в першу чергу, проходять верифікацію і після запису в СУБД запускається процес розповсюдження підтверджених змін в конфігурацію на інші сервери системи (рис.1.2). Даний підхід має високий рівень гнучкості та відмовостійкості, оскільки кожен вузол

здатний функціонувати автономним чином, що в свою чергу дозволяє будувати та об'єднувати системи, які географічно розподілені на велику відстань між собою.

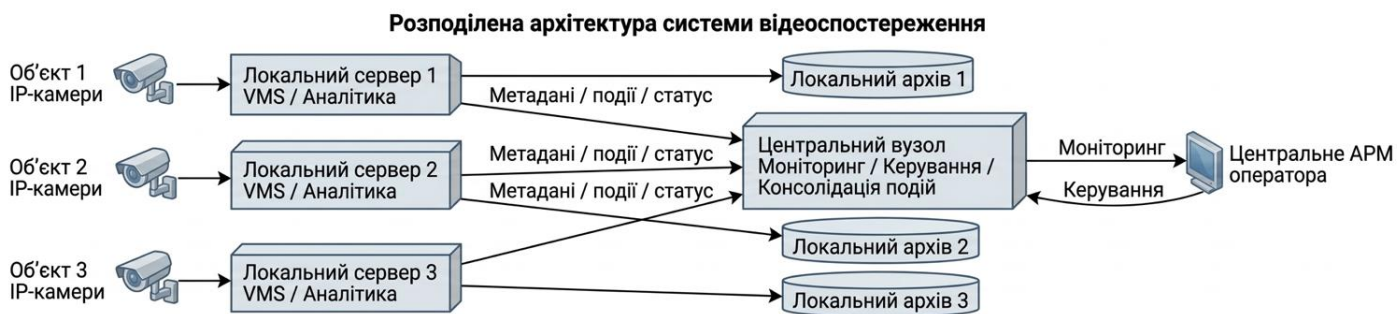


Рис.1.2. Розподілена архітектура

Переваги розподіленої архітектури:

- Високий рівень масштабованості та гнучкості;
- Високий рівень відмовостійкості та автономності;
- Особливо доцільно для використання у географічно розподілених об'єктах;
- Зменшення затримок в передачі даних;
- Зменшені витрати на локальне обладнання;
- Можливість поетапного впровадження системи.

Недоліки розподіленої архітектури:

- Збільшення витрат на побудову цілісної системи;
- Складність адміністрування обладнання та системи;
- Підвищені вимоги до персоналу або необхідність залучення сторонніх обслуговуючих компаній;
- Неможливість централізованого збереження даних;

1.2.3. Гібридна архітектура

Для об'єктів, які є особливо складними та містять багато різнопланових локацій, що можуть бути також географічно віддалені один від одного але потребують

централізованого підходу, існує можливість побудови гібридної архітектури(рис1.3). Такий підхід об'єднує всі переваги централізованого та розподіленого підходу, що вирішує більшість задач, які пов'язані з технічними аспектами впровадження щодо якості роботи системи, але є найбільш дорого вартісним та складним в адмініструванні.

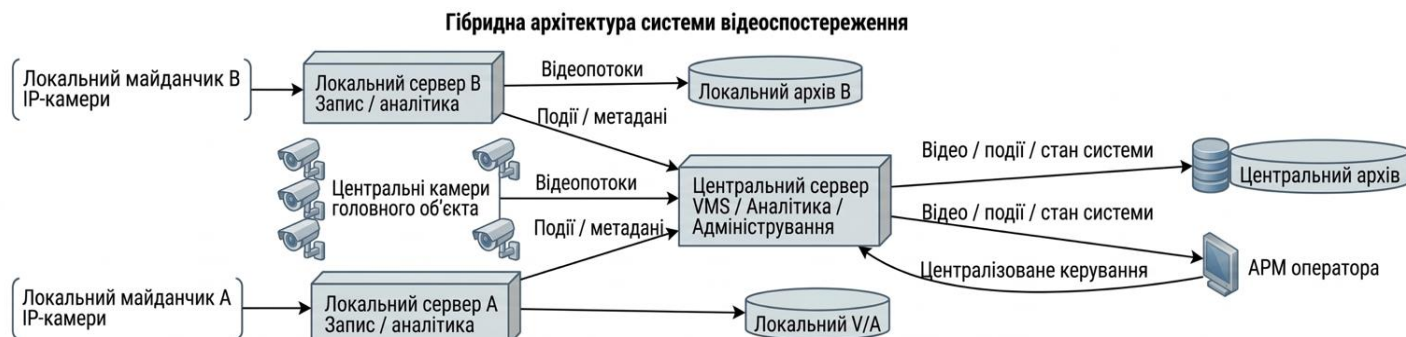


Рис.1.3. Гібридна архітектура

Переваги гібридної архітектури:

- Підхід містить всі переваги розподіленої та централізованої архітектури;
- Висока відмовостійкість системи.

Недоліки гібридної архітектури:

- Високий рівень фінансових затрат на початку та під час експлуатації;
- Складність в адмініструванні

1.2.4. EDGE-рішення та NVR

Для окремих випадків існує можливість використання відеокамер з вбудованою аналітикою, що є особливо актуальним для рішень з низькою пропускнуою здатністю мережі(рис.1.4). Під час роботи камера може не транслювати постійно зображення на сервер для обробки, а виконує всі аналітичні операції на власному процесорі, що дозволяє передавати лише результати обробки відеопотоку – метадані, з мінімальним використанням трафіку. Не зважаючи на таку перевагу існує також ряд критичних

недоліків, а саме вартість відеокамер в декілька разів перевищує ціну на звичайні, відсутність інтеграції з сторонніми системами, які не підтримуються виробником, а також неможливість заміни аналітики на іншу, оскільки вона прив'язана виробником камери. Для цих задач можна використовувати NVR аналогічного бренду, що розширить область застосування та функціональні можливості, але такий підхід раціонально використовувати у невеликих об'єктах або ж окремих складних специфічних випадках.

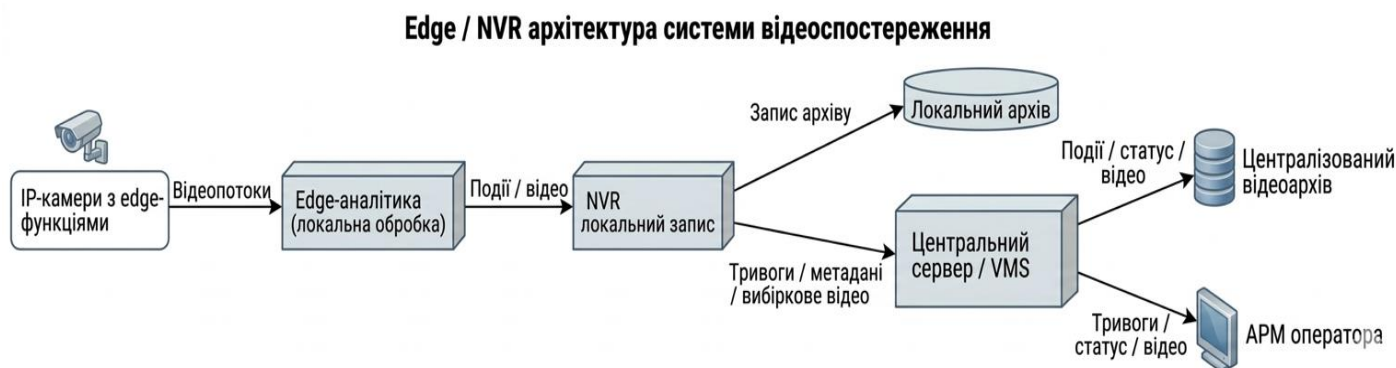


Рис.1.4. EDGE/NVR архітектура

Переваги EDGE/NVR архітектури:

- Економія трафіку та низькі вимоги до мережевої інфраструктури;
- Швидке налаштування та впровадження;
- Просте масштабування.

Недоліки EDGE/NVR архітектури:

- Обмежені обчислювальні ресурси обладнання;
- Монобрендовість і відсутність інтеграцій з іншими системами;
- Висока ціна за одиницю обладнання;
- Наявність єдиної точки відмови;
- Майбутні складнощі в адмініструванні та обслуговуванні

1.2.5. Хмарна архітектура та vSaaS

Найсучасніший підхід для побудови системи - це оренда серверної обчислювальної потужності, а також дискового простору(рис.1.5). Така архітектура дозволяє особливо швидко та ефективно впровадити систему, так як не вимагає закупівлі обладнання, його монтажу, налаштування, а лише оренда на визначений договором час. Масштабувати таку систему легко, обслуговування апаратних компонентів повноцінно відповідальність орендодавця, але система критично залежна від якості мережевого підключення її стабільності та об'ємів даних. На початку впровадження не вимагає великих фінансових інвестицій, але при використанні даної топології в перспективі декількох років операційні витрати стають суттєві та можуть дорівнювати вартості розгортання власної інфраструктури.

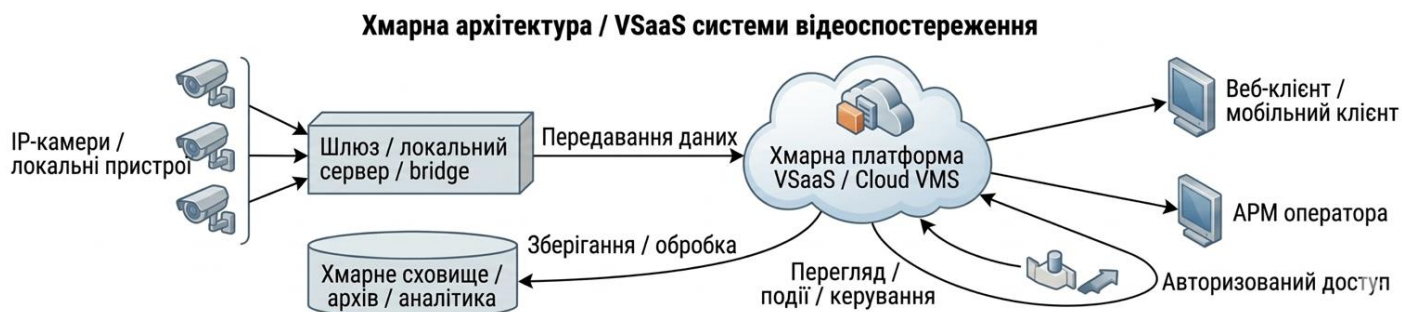


Рис.1.5. Хмарна архітектура

Переваги хмарної архітектури:

- Відсутнє адміністрування обладнання;
- Швидке впровадження системи;
- Невеликі початкові витрати;
- Найбільш просте масштабування;
- Зменшення вимог до кількості персоналу та його кваліфікації;
- Централізоване керування.

Недоліки хмарної архітектури:

- Високі вимоги до стабільної мережевої інфраструктури;
- Наявність єдиної точки відмови;
- Зростання експлуатаційних витрат.

1.2.6. Порівняння типів архітектур

Як було визначено раніше кожна архітектура має свої переваги та недоліки з точки зору впровадження так і експлуатації, що відображено в Таблиці 1.1. Зміна архітектури системи це ресурсоємний процес, який вимагає оновлення та закупівлі додаткового обладнання, заміну програмного забезпечення, якщо таке не підтримує необхідні архітектурні функції, зупинка роботи системи та її повторне повне конфігурування. На етапі проектування системи варто особливу увагу приділяти саме визначенню поточних задач, а також фіксацію подальших потенційних завдань, які має вирішувати програмно-апаратний комплекс, що дозволяє раціонально визначити необхідну топологію.

Таблиця 1.1.

Порівняння типів архітектур

Критерій	Централізована	Розподілена	Гібридна	Edge / NVR	Хмарна / VSaaS
Простота адміністрування	так	ні	ні	частково	так
Масштабованість	так	так	так	частково	так
Відмовостійкість	частково	так	так	ні	ні
Відсутність єдиної точки відмови	ні	так	так	ні	ні

Локальна автономність сегментів	ні	так	так	так	ні
Централізоване керування	так	частково	так	частково	так
Зручність інтеграції з іншими підсистемами	так	частково	так	ні	частково
Ефективність для територіально розподілених об'єктів	ні	так	так	частково	так
Низькі вимоги до магістральних каналів зв'язку	ні	частково	частково	так	ні
Можливість локальної обробки даних	ні	так	так	так	ні
Доцільність для малих об'єктів	так	частково	ні	так	так
Гнучкість побудови	частково	так	так	частково	так
Швидке розгортання	частково	ні	ні	так	так
Контроль над інфраструктурою	так	так	так	так	ні
Залежність від Інтернет-з'єднання	ні	ні	частково	частково	так

Придатність для аналітики	так	так	так	частково	так
Економічність для малих об'єктів	так	ні	ні	ні	так
Економічність для великих розподілених систем	так	так	так	ні	так

1.3. Елементи аналізу даних у системах відеоспостереження

Системи відеоспостереження складаються з різних типів відеокамер, але станом на сьогодні, в своїй більшості, використовуються IP-відеокамери, які формують картинку та передають її на сервер чи кінцеві пристрої для відображення, аналізу та зберігання. Сучасні пристрої мають велику кількість вбудованих технологій покращення зображення, що дозволяє отримувати якісну картинку в різних складних умовах. До таких технологій відноситься WDR, HLS, BWL, які здатні змінювати картинку в режимі реального часу та затемнювати або навпаки висвітлювати окремі області картини для отримання рівномірного за освітленістю та контрастом картини. Існують також різні технології стиснення відео, як H.264 та H.265 і їх модифікації, які спрямовані на зменшення бітрейту зі збереженням якості потокового відео. Крім таких технічних інструментів також окремі моделі пристроїв підтримують ряд додаткових функцій, як ІЧ-підсвітка, поворотні механізми камери, змінну фокусну відстань, антивандальний захист, тепловізор, можливість запису даних на вбудоване сховище камери і навіть використання аналітичних інструментів на борту самої відеокамери. Як висновок ми працюємо з високофункціональним пристроєм, який може бути застосований в абсолютно різних складних умовах. Вартість та загальна доступність дешевих відеокамер на сьогодні дозволяє її придбати менше за 3 000 гривень, а більшість

непоганих початкових моделей, які можна використовувати для побудови базової системи відеоспостереження в реальних умовах, коштує в діапазоні 4 500 – 10 000 грн.

Як результат у комерційних, приватних та державних об'єктах використовується велика кількість камер, через їх цінову доступність, і якщо одні об'єкти можуть налічувати до 10 відеокамер сумарно, то для інших ця цифра може складати 15 000 і більше. Такі об'єми генерують велику кількість даних, які потрібно обробляти та зберігати, де традиційний відеонагляд вже втратив актуальність через низьку ефективність обробки великих масивів даних.

Для вирішення задачі обробки даних існують різного роду аналітичні інструменти, які дозволяють спросити роботу з обробкою відеоданих і фіксувати лише необхідні відрізки відео, які мають корисну інформацію.

Інструменти для аналізу відеопотоку:

- Детектор руху;
- Інтелектуальні детектори “Трекер”;
- Аналітика розпізнавання обличь;
- Аналітика розпізнавання номерів ТЗ;
- Аналітика на основі нейронних мереж;

1.3.1. Роль відеоаналітики у сучасних системах та базові аналітичні інструменти

Задача аналітичних інструментів полягає в зменшенні обсягу інформації та її фільтрація на корисну та непотрібну. Для різних за масштабом систем такі інструменти особливо корисні, оскільки дозволяють привертати увагу до події коли вона сталась і не потребує постійного перегляду відео для фіксації інцидентів. Один з перших детекторів, що був створений має назву – детектор руху. Цей детектор і досі використовуються в системах відеонагляду та користуються широким попитом через простий принцип роботи і легкість в налаштуванні. Даний інструмент визначає зміну пікселів на

зображенні і якщо таких пікселів було більше ніж задано під час налаштування - система генерує тривожне повідомлення. За допомогою такого підходу можна відмовитись від класичного перегляду відеопотоку, а зосередити увагу на зафіксованих подіях детектором, що якісно впливає на увагу операторів, не вимагає великих ресурсів та записує лише корисні для аналізу події. Застосування аналітичних інструментів дозволило також значно зменшити використання дискової підсистеми, оскільки немає необхідності записувати відео безперервно, а лише ті ділянки відео де присутні події виявлення руху.

Не зважаючи на високу ефективність детектору руху з часом вимоги до його експлуатації ставали ще більш вимогливі і якість роботи такого детектору стала недостатня. Це зумовлено принципом, який закладений в логіку роботи детектора. Будь-яка зміна групи пікселів визначалась як тривожна подія і детектування відбувалось на всі подібні інциденти, як наприклад зміна освітлення на зображенні, зміна погодних умов, рух рослинності та інше.

Детектор “Трекер”, який в подальшому замінив роботу детектору руху, працює за схожим принципом, але містить велику кількість, вбудованих в логіку, сигнатур, а саме розміри та пропорції об’єктів (людина, автомобіль), що дозволяє визначати точніше, об’єкт, який утворює рух. Крім того даний алгоритм фіксує стан пікселя, що реалізує можливості пошуку об’єктів за кольором. Такий алгоритм створив нові додаткові детектори та пошукові інструменти, як пересічення лінії, визначення напрямку руху, пошук за кольором, рух в заданій області, перехід з області в область, визначення типу: людина, група людей, автомобіль.

Водночас відбувається розробка таких аналітичних інструментів, які дозволяють визначати та розпізнавати обличчя і номерні знаки транспортних засобів. Аналітика спростила роботу над даними, які транслює камера та зняла навантаження з операторів, зменшила кількість помилок, підвищила рівень оперативного реагування на інциденти та зменшила обсяги даних для збереження.

1.3.2. Нейромережеві методи аналізу даних

На відмінну від підходу де система очікує на передбачувану математичну зміну пікселів, аналітика на основі нейронних мереж використовує машинний зір, який не працює з сигналом або послідовністю пікселів серед кадрів, а аналізує конкретний кадр та визначає на ньому об'єкт.

Точність аналітики визначається кількістю шарів, які досягаються об'ємом матеріалу на якому навчається аналітика. Кожен шар складається з нейронів, які здатні генерувати “вагу” або коефіцієнт достовірності характеристики об'єкту. Після того, як зображення було передано на аналіз відбувається декілька ключових етапів з зображенням, щоб аналітична нейронна модель надала результат. Важливо відмітити, що нейронна модель не генерує конкретного результату, вона визначає відсоток достовірності, за яким відбувся розрахунок ймовірності, що зафіксований об'єкт дійсно є в кадрі та має вигляд дробового числа, наприклад 0.93.

Основні етапи обробки зображення та аналітичний аналіз:

1. Попередня обробка зображення (Preprocessing)

Критично важливий етап, який готує зображення до стандартів формату матриці, на якій навчалась мережа. Спершу відбувається зміна роздільної здатності до необхідної (наприклад: 640x640, 320x320 та інше), яке не просто розтягується, а використовуючи спеціальні алгоритми, як «LetterBox», який зменшує матрицю зі збереженням пропорцій, щоб не допустити спотворення об'єктів.

2. Нормалізація зображення

Обирається необхідна кольорова схема каналів BGR/RGB та перетворення значень кольору з 8-ми бітового представлення до чисел з плаваючою комою. Зазначений підхід дозволяє обраховувати обчислювальним блокам дробові числа типу float32, що є набагато простішим та стабільнішим ніж робота з цілими числами типу integer. Для цих задач CPU не є підходящим, а тому використовуються обчислювальні блоки на GPU/NPU, які краще справляються з такими задачами за рахунок іншої архітектури,

оскільки основна відмінність між ними є в кількості обчислювальних ядер, та паралельному опрацюванню інструкцій.

3. Створення тензору

Це заключний етап в підготовці зображення до передачі його для аналізу, де звичайна картинка перетворюється на математичний (алгебраїчний) обчислювальний ресурс. Фактично тензор - звичайний уніфікований контейнер в якому збережені дані відповідно до необхідної структури $[N, C, H, W]$ або $[C, H, W]$, які будуть зрозумілі для нейромережових фреймворків.

4. Визначення характерних ознак (Feature Extraction)

Виявлення об'єктів з використанням математичних алгоритмів, які фільтрують дані на корисні та непотрібні, що працює за признаками визначення горизонтальних, вертикальних ліній та кольорових переходів. Після визначення блоків з корисною інформацією вони аналізуються на предмет зрозумілих характеристик об'єкту, а вже потім стискаються.

5. Інференс (Inference)

Готовий тензор з виділеними об'єктами передається та рухається крізь шари нейронів, які аналізують його та фіксують ймовірність. Сучасні мережі дозволяють визначати такі ймовірності (рис.1.6):

- Ймовірність наявності об'єкту на зображенні;
- Ймовірність класу (людина, авто, собака/кіт);
- Координати прямокутника (межі об'єкту);

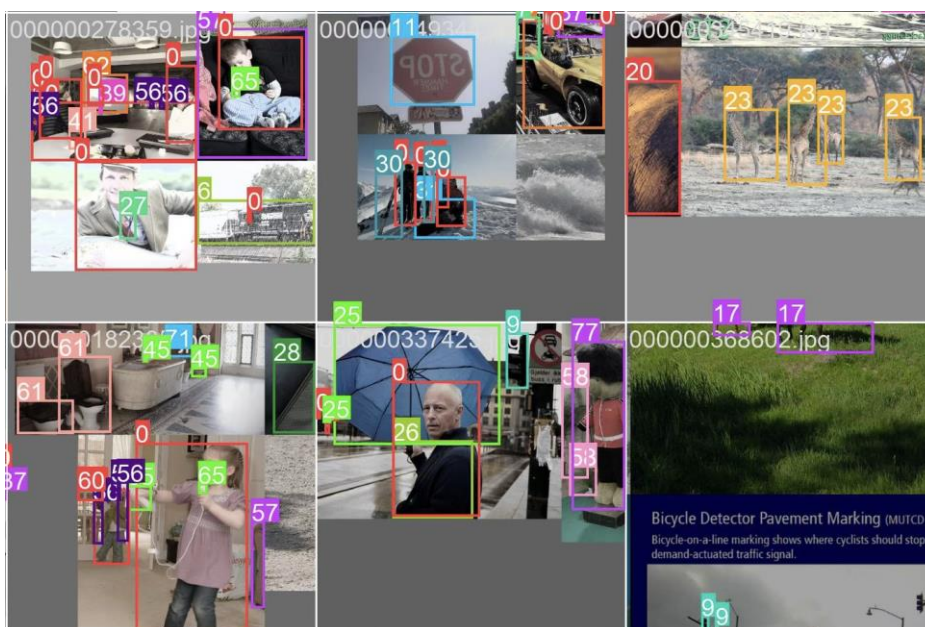


Рис. 1.6. Принцип детекції об'єктів нейромережевим детектором

Отриманий результат, за офіційними показниками виробників YOLO, сягає 53-53% точності mAP % на COCO, що є високим показником, оскільки дослідження такої нейронної мережі ведеться на переліку з будь-яких картинок інтернету і їх ракурси, положення об'єктів, спотворення пропорцій, шуми не є уніфікованими, як в системах безпеки. Тому використовуючи офіційні джерела та співпрацю з виробником систем відеонагляду «АххонSoft» було проведено ряд досліджень, в яких було визначено, що якість коректних детекцій, складає 95%-99% точності. Відмінність результату полягає в початковому зображенні, яке в будь-якій локації, країні чи території буде приблизно однакова відповідно до норм та положень монтажу відеокамер на периметральній території та всередині приміщень. Таким чином стандартизація має лише позитивний вплив на якість роботи аналітики.

1.3.3. Вплив точності аналітичних інструментів на ефективність системи

Результати роботи аналітичних інструментів, що засновані на основі машинного зору застосовуються також в таких детекторах як, розпізнавання обличчя та

розпізнавання номерів ТЗ, що значно знизило вимоги до якості зображення та місця розташування камери.

На рис. 1.7 представлено систему розпізнавання номерних знаків, яка працює за двома версіями математики. В лівій частині зображення застаріла модель, яка не має нейромережових доопрацювань, а тому такий кут встановлення є максимально допустимим для якісної роботи аналітики розпізнавання номерних знаків транспортного засобу. В лівій частині зображення показано сучасний підхід з використанням нейронних мереж, який дозволяє виконувати розпізнавання в екстремально складних умовах, навіть коли недотримуються правила монтажу та дистанція до об'єкту.



Рис. 1.7. Порівняння розпізнавання номерів з двома різними моделями аналітики

Сучасний підхід та імплементація машинного зору дозволяє змінити мінімальні вимоги до якості зображення, умови монтажу відеокамери, та дозволяє обробляти об'єкти, які мають значно складніші умови. Оскільки в реальних умовах експлуатації, людина може бути одягнена в головні убори, окуляри або медичні маски, як

представлено на рис. 1.8. де обличчя людини в більшості закрито маскою, то такі ситуації значно ускладнюють процес ідентифікації особи. Тим не менш, не зважаючи на такий рівень перекриття аналітика здатна працювати ефективно і розпізнає обличчя.

Можемо зробити висновок, що з використанням машинного зору, якісь та точність аналітики значно зросла і такий підхід в системах безпеки є набагато ефективніший ніж попередні версії алгоритмів розпізнавання. Системою генерується менша кількість хибних подій, точність зростає, вимоги до монтажу спрощується та вже існують прецеденти, де побудований окремий контур безпеки та бізнес-процесів в якому людина не приймає участі, що потенційно зменшує витрати та збільшує ефективність вцілому.

The screenshot displays a face recognition software interface. At the top, there are tabs for 'Monitor', 'Search', 'Face DB', 'Recognize file/folder', 'Analytics', and 'Statistics'. Below these, there's a section for 'Archive' with a 'Show faces' dropdown set to 'All', a 'Filters' button, a checkbox for 'Follow new faces', and a 'View' button. The main area is a table of captured faces with columns: 'Captured face', 'Original from DB', 'Full name', 'Age', 'Gender', 'Camera', and 'Date/Time'. The table lists several entries, with the fourth entry highlighted in blue, indicating a successful match.

Captured face	Original from DB	Full name	Age	Gender	Camera	Date/Time
		Kurinnyi Oleksandr AxxonSoft 99.4 %	32	Male	Camera 1	6/4/2020 2:36:02 PM
		Dyban Oleksii AxxonSoft 99.0 %	54	Male	Camera 1	6/4/2020 2:34:42 PM
		Kurinnyi Oleksandr AxxonSoft 97.3 %	27	Male	Camera 1	6/4/2020 2:34:37 PM
		Pevnev Vladislav AxxonSoft 95.7 %	40	Male	Camera 1	6/4/2020 2:34:36 PM
		Liubyi Bohdan AxxonSoft 89.3 %	49	Male	Camera 1	6/4/2020 2:33:21 PM
		Pevnev Vladislav AxxonSoft 99.1 %	26	Male	Camera 1	6/4/2020 2:33:12 PM

On the right side, there's a detailed view of the selected match. It features a red banner with the text 'Attention! Pevnev Vladislav'. Below this, there are two images: a captured face with a mask and an original face from the database. A large '96%' similarity score is displayed. At the bottom right, there's a smaller table with columns: 'Image', 'Full name', 'Similarity level', 'Comment', and 'Date/Time'. It shows a match for 'Pevnev Vladislav AxxonSoft' with a similarity level of 95.7 % and a date/time of 6/4/2020 1:53:29 PM.

Рис. 1.8. Розпізнавання обличч з перекриттям

1.4. Аналіз існуючих підходів оцінювання системи відеоспостереження

Станом на сьогодні, при проектування систем відеонагляду використовуються різні підходи, які дозволяють оцінити ефективність системи або ж її рентабельність до впровадження за різними критеріями. Кожен з підходів має свій обсяг критеріїв, формалізацію, обґрунтування та аргументи. Проведений аналіз показує, що у своїй більшості використовується лише один метод для оцінки, який не здатний охарактеризувати та показати цілісний показник якості і ефективності системи, що призводить в свою чергу до вибору частково обґрунтованої пропозиції.

Одним і найрозповсюдженіший підхід є оцінювання технічних характеристик обладнання та системи. В даному підході визначаються обсяги обладнання, його продуктивність, точність аналітичних інструментів, мережева інфраструктура, безпека даних, зберігання архіву та забезпечення відмовостійкості. Такий підхід має свої переваги оскільки він в повній мірі визначає мінімальні порогові значення по критеріям, які важливі для реалізації системи для досягнення відповідного рівня роботи. З іншого боку, аналізуючи лише технічну складову, належним чином не досліджують факт подальшого розвитку системи з точки зору організації процесів та їх автоматизації, масштабування системи в рамках декількох підрозділів одного й того ж підприємства, економічну доцільність і функціональну ефективність.

Інший підхід заснований на фінансових обмеженнях бюджету та вартості впровадження. В такому випадку технічні характеристики часто нехтуються та не враховуються при детальному аналізі. Кожен з виробників має свій ціновий сегмент, характеристики обладнання і поширена практика, коли бюджетні бренди пропонують високі параметри обладнання за нижчою ціною. Тому для таких підходів часто обираються такі бюджетні рішення, які по специфікації є підходящими. Під час експлуатації, такий метод показує, що економія на компонентах та якісному обладнанні робить систему неефективною як з технічної точки зору так і з фінансових метрик. Збільшення проблемних інцидентів, заміна комплектуючих, часті проблеми зі

стабільністю впливають на постійне обслуговування системи. Відповідальність за систему несуть співробітники, а рішення приймається фінансовим директором, який оцінює потенційні витрати тільки по вартості впровадження.

Також існують підходи, де визначається суто лише окремі функції системи відеоспостереження, як точність аналітики, наявність тих чи наших інструментів, що є найменш стратегічним способом в даному списку. Для підтвердження цифр точності часто використовують випробування, на яких вимірюється на реальному об'єкті якості роботи аналітики. В результаті такі підходи, в своїй більшості, є найбільш коштовними і мають відсутню структуру впровадження в цілісну систему безпеки та не характеризують підхід до експлуатації в цілому.

Оскільки проектуванням таких систем займаються інсталятори, то в якості критеріїв часто використовують свою експертну думку, яка є заангажованою під конкретний бренд, який є найбільш вигідним. Для підтвердження використовується маркетинговий матеріал, в якому не вказуються недоліки рішення, а фокус спрямований саме на переваги. Без прямого порівняння всіх критеріїв, з якого складається такий проект, доволі складно оцінити різницю в ефективності альтернатив, чим і зловживають комерційні організації.

Отже, проведений аналіз існуючих методів оцінки показує, що наразі не існує єдиного універсального методу, який би дозволяв якісно провести оцінку всіх необхідних параметрів проекту. Кожен з описаних підходів фрагментований та описує лише часткову складову проекту, якою оперуються при виборі того чи іншого рішення. В одному випадку орієнтується суто на технічні показники, без визначення чітких процесів, в іншому обмежуються бюджетом і не розраховують потенційні інциденти та їх вартість в перспективі часу, а інші приклади взагалі обирають систему по якості аналітики, або окремих критеріях, які локально дозволяють вирішити задачу, але неспроможні інтегруватись в цілісну структуру. Такий метод не може формуватися замовником, оскільки спеціаліст в конкретній (технічній, фінансовій, управлінській, маркетинговій) області не може зафіксувати ряд задач за межами своєї компетенції, а

проектанти або компанії-інсталятори, завчасно зацікавлені у використанні конкретного бренду, який для них буде найбільш вигідним і простим. Як результат оцінкою та підготовкою такого методу має займатись третя сторона, яка не є зацікавленою. Унаслідок цього є необхідність в розробці методу комплексного оцінювання та оптимізації систем відеоспостереження.

1.5. Моделювання функціональних залежностей та каскадних впливів у кіберфізичних системах відеоспостереження

Інтеграція технологій мобільного зв'язку п'ятого покоління (5G), периферійний обчислень (MEC) та промислового інтернету речей (IoT) в інтегровані системи відеонагляду (ICV) сприяє такому еволюційному переходу до індустрії 4.0, але водночас такий підхід створює нові виклики для проектування та оцінювання подібних систем. Традиційні ICV, як і традиційні промислові системи керування (ICS/OT), протягом тривалого часу при проектуванні не враховувались і були як ізольовані, закриті середовища, де стабільність роботи такої системи досягалась фізичними обмеженнями доступу та незмінною топологією мережі.

Сучасні гібридні архітектури ICV мають високий рівень взаємозалежності компонентів, динамікою в зміні топології роботи а також використання віртуалізації мережевих функцій (VNF). У випадку, коли такі системи використовують технологію 5G для підтримки високої щільності підключень (mMTC), високої надійності зв'язок з низькою затримкою (URLLC) та розширена пропускна здатність мережі (eMBB), можна свідчити, що відеокамери, сервери та компоненти системи більше не функціонують як ізольовані елементи.

Ця конвергенція значно розширює поверхню вразливості систем до так званих каскадних відмов (cascading failures). Каскадна відмова формується в таких випадках, коли відбувається деградація або перевантаження одного вузла (наприклад, віртуальної функції маршрутизації UPF або ж вузла аналітики MEC) та лавиноподібно поширюється

по всіх структурі системи, що в свою чергу призводить до повної втрати функціональності навіть таких компонентів, які не були фізично пошкоджені або ж перевантажені.

1.5.1. Критичний аналіз існуючих аналітичних підходів

Більшість сучасних рішень для оцінки ефективності та ризиків зосереджені на аналізі індивідуальних активів і не враховують функціональні залежності між компонентами ICB та між мережевими елементами інфраструктури. Зазначений підхід не дозволяє повноцінно виявити шляхи поширення загроз та вірно оцінювати ризики, які виникають у наслідок побудови складної структури взаємозалежності систем.

Аналіз поточних існуючих підходів у сфері ICS/OT та 5G-середовищ показує, що вони мають значний рівень фрагментованості:

1. Підходи на базі метрик індивідуальних активів: Використання балових систем (наприклад, CVSS) що дозволяє оцінити критичність окремих вузлів, але водночас ігнорує структурний контекст їх взаємодії в реальному часі.

2. Статичні графи атак та дерева відмов: Дозволяють описати сценарії компрометації з точки зору зловмисника або апаратного збою. Недоліком таких моделей є їх статичність, а через це відсутній розрахунок в розрізі часу та не враховують логіку динамічних віртуалізованих функцій (наприклад, network slicing).

3. Методи на базі штучного інтелекту (ML/AI): Такі інструменти працюють на базі графових нейронних мереж (GNN) які здатні виявляти аномалії в трафіку (наприклад, сплески в окремі часові відрізки бітрейту на камерах), але такі інструменти на надають інтерпретованої моделі функціональних залежностей між вузлами системи, що не є підходящим рішенням для використання інженерної оптимізації.

В результаті аналізу можу зазначити, що ключовий недолік існуючих підходів є обмеженість у відображенні міжвузлових залежностей у фізичному, логічному а також сервісному просторі. Крім того у вказаних підходах присутнє ігнорування часової

динаміки поширення впливів до яких відноситься затримка, тривалість, частота оновлення. Жоден з наявних методів не забезпечує повної формалізованої структури, яка була б здатна одночасно врахувати багаторівневу топологію мережі, часові атрибути алгоритмів аналізу даних та критичність активів.

1.5.2. Архітектура та класифікація вузлів у моделях функціональних залежностей

Для вирішення зазначених обмежень необхідно застосовувати підхід з використанням математичного апарату графів функціональних залежностей (Functional Dependency Graphs, FDG), який слугуватиме раціональною структурною основою для подальшого кількісного аналізу системи. Якщо розглядати контекст гібридних СІВ, то в ньому вузли такого графу будуть класифікуватись за функціональною роллю та погодженням що розділено на наступні макрорівні:

1. Компоненти ICS/ОТ (фізичний та керуючий рівень): Компоненти, які охоплюють класичні елементи технологічного процесу системи. Для ІСВ це різні пристрої як відеокамери, сенсори, датчики, програмовані контролери (PLC) сервери управління та інтерфейс оператора (HMI). Такі вузли безпосередньо приймають участь у формуванні візуальних даних і мають обмежені технічні ресурси.

2. Інфраструктура 5G/Транспортна мережа: Включає в себе як апаратні так і програмні компоненти, які необхідні для передачі даних та їх маршрутизацію. Віднести до цього рівню можна базові станції (gNB), клієнтське обладнання, функції ядра мережі (AMF, UPF, SMF), а також API-шлюзи та сервери периферійних обчислень, на яких розгортається програмне забезпечення з аналітичними інструментами. Вказані вузли відповідають за віртуалізацію мережевих функцій та гарантовану доставку об'ємного трафіку.

3. Пристрої та сервіси ІоТ/Edge (Рівень інтелектуального аналізу): Охоплює кінцеві пристрої з інтелектуальними функціями, а також хмарні технології які

використовуються для аналітики відеозображення та системи візуалізації. Вказані програми взаємодіють з мережею через ізольовані зрізи (network slices) та оброблюють відеопотік у режимі реального часу.

Така, визначена, класифікація може бути представлена у такому вигляді як на рис.1.9.

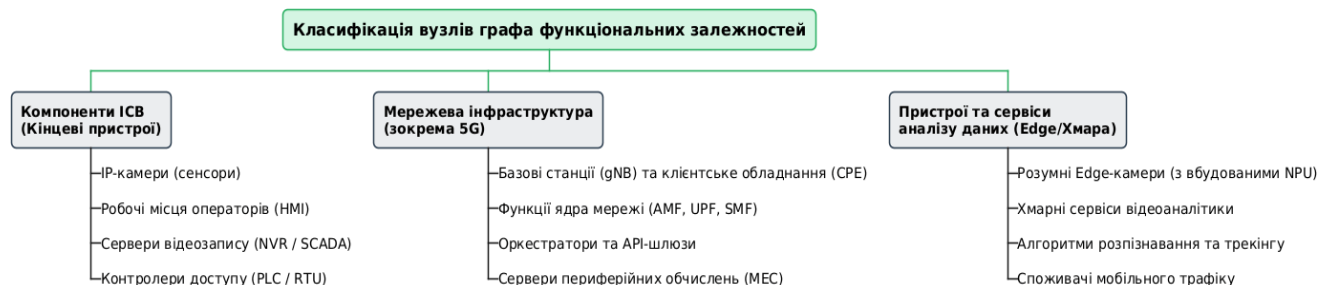


Рис.1.9. Класифікація вузлів графа функціональних залежностей в гібридних інтегрованих системах відеоспостереження.

1.5.3. Типологія функціональних взаємозв'язків (Топологія FDG)

В структурі графа FDG ребра виконують функцію формалізації міжвузлових взаємодій, які мають логічне або операційне значення. Типологія ребер визначає характер залежності між компонентами в ІСВ дозволяє показати їх динамічні властивості. Можна виділити п'ять основних типів функціональних зв'язків:

1. Інформаційна залежність: Створюється тоді, коли в цілісній системі один вузол і генерує і передає інформацію одночасно, яка є критично важлива для роботи іншого вузла.
2. Залежність керування: Утворюється тоді, коли один елемент системи визначає логіку роботи, стан та реакцію іншого компоненту.
3. Ресурсна залежність: Така залежність описує випадок, коли декілька компонентів які пов'язані між собою в одній системі мають фізичну або логічну залежність від використання спільного вузла обробки даних або ж мережевого ресурсу.

4. Часові обмеження: Відображають граничні межі затримок передачі даних.
5. Функціональний зв'язок: Виникає у випадках, коли в одному бізнес-процесі, приймає участь та формується логіка поведінки декількох вузлів.

Представлення типів таких зав'язків зображено на рис.1.10.

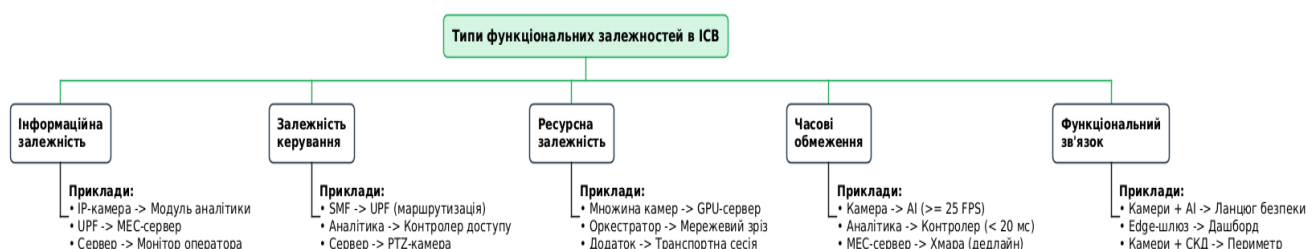


Рис.1.10. Типологія функціональних залежностей та напрямків каскадних впливів між компонентами ІСВ.

1.5.4. Семантична атрибуція як основа для кількісного оцінювання

Для того, щоб граф функціональних залежностей можна було застосовувати для методу визначення кількісного аналізу ефективності та ризиків, потрібно щоб кожне ребро та вузол мали описаний набір семантичних атрибутів. Такий підхід дозволяє ідентифікувати функціональну роль та додатково оцінити вплив на рівень критичності, вплив порушень та сформулювати чутливість до перевантаження.

- Атрибут критичності дозволяє визначити ступінь важливості конкретного вузла для того, аби забезпечити роботоздатність технологічного процесу.
- Рівень довіри, як атрибут, визначає ступінь надійності даного компоненту щодо використання та відповідно до рівня доступу (зовнішній або внутрішній).
- Затримка – характеризує пропускну здатність вузла та його продуктивність.
- Атрибут резервування визначає чи існує альтернативний маршрут або ж резервний компонент в системі, який може бути задіяний в разі виходу з ладу основного.

- Радіус впливу – дозволяє оцінити масштаб поширення каскадного ефекту в динамічному середовищі. Визначається кількість компонентів, які можуть вийти з ладу при такому сценарії.

Розробка моделей на базі графів функціональних залежностей для середовищ, які інтегрують ICS/OT компоненти з 5G та AI-інфраструктурою, потребує у врахуванні таких вимог, як гетерогенність, адаптивність до динаміки мережі та масштабованість. Сформована модель здатна відображати структуру гібридної системи відеоспостереження, яка складається з множини зав'язків, що збагачені семантикою. Таким чином формується аналітичне підґрунтя для розробки формалізованих методів оцінювання та багатокритеріальної оптимізації ICB.

1.6. Класифікація підходів оцінювання за групами критеріїв

Після проведення аналізів існуючих підходів до оцінки проекту системи відеоспостереження було визначено, що підходи існують, але вони неоднорідні, мають різні критерії оцінки та не враховують повний спектр необхідних аргументів для коректного та зваженого прийняття рішення. В подальшому, для можливості оцінки, необхідно створити та систематизувати такі підходи за типом та критеріями в цілісний метод оцінки ефективності. Такий підхід дозволить оцінити запропоновану систему з повним набором компонентів, які можуть бути важливими для прийняття рішення вибору комплексної інтегрованої системи відеоспостереження з елементами аналізу даних.

Виділяємо наступні параметри оцінки:

1. Технічні параметри;
2. Економічні параметри;
3. Експлуатаційні параметри;
4. Аналітичні параметри;
5. Функціональні параметри;

6. Ризикові параметри.

1.6.1. Технічні критерії

Технічні параметри описують ресурси та технічні засоби, які будуть впроваджені в проєкті. До переліку необхідних параметрів включається не лише перелік обладнання, а й його налаштування та конфігурація, яка прямопропорційно впливає на інші елементи підсистем безпеки. Для оцінки таких систем буде недостатнім описати кількість камер та їх характеристики, а важливо зазначити, яка буде використовуватись роздільна здатність, кількість кадрів, бітрейт та кодек стиснення. Для коректного розрахунку необхідно розрахувати корисний дисковий простір для збереження даних та завантаженість пропускної мережі для проєктуванні мережевої інфраструктури. Ці показники дозволяють запроектувати систему максимально деталізовано та, за допомогою такого підходу, використовувати прогнозування щодо можливих похибок.

Визначаємо такі параметри:

- Відеокамера (роздільна здатність, частота кадрів, тип стиснення, кут огляду, інтенсивність сцени);
- Аналітика (тип аналітики, точність виявлення, обчислювальні мінімальні вимоги, задача аналітичного інструменту)
- Сервер (продуктивність та навантаження на сру/гру, обсяги оперативної пам'яті, пропускна здатність інтерфейсів, резерв продуктивності, надійність обладнання, мережева інфраструктура, пропускна здатність, топологія мережі, розподіл трафіку, якість та надійність обладнання, резервування обладнання);
- Система зберігання даних (необхідність такого підходу, обсяг архіву, тривалість зберігання, тип дисків, масштабованість NAS, рівень RAID);
- Резервування та відмовостійкість (наявність єдиної точки відмови, роботоздатності, резервування живлення, відновлення системи після збою).

1.6.2. Економічні критерії

Економічні критерії характеризують фінансову сторону проекту, яка складається з доцільності впровадження та експлуатації системи. Врахування таких параметрів дозволить оцінити повну вартість проекту на старті, вартість володіння системою, терміни окупності у межах циклу проекту.

До таких параметрів можна віднести:

- CAPEX - початкові капітальні витрати;
- OPEX — поточні витрати на експлуатацію системи;
- Витрати на обладнання;
- Витрати на програмне забезпечення;
- Витрати на інсталяцію та введення в експлуатацію;
- Витрати на обслуговування;
- Витрати на персонал;
- Вартість володіння системою;
- Терміни окупності.

1.6.3. Експлуатаційні критерії

Експлуатаційні критерії визначають наскільки система буде зручна та корисна в експлуатації в реальних умовах. Дані критерії дозволяють виявити навантаження на експлуатацію та адміністрування, кількість необхідних ресурсів та вимоги до кваліфікації персоналу, до яких відноситься:

- Складність адміністрування;
- Технічний супровід;
- Кваліфікація персоналу;
- Гнучкість налаштування;
- Можливість поетапного впровадження;

- Зручність експлуатації.

1.6.4. Аналітичні критерії

Аналітичні критерії характеризують вимоги до аналітичних інструментів системи, їх задачі, точність роботи та оцінка реальної ефективності. Оскільки об'єм аналітики прямо впливає на необхідні потужності серверу даний пункт має бути прямо ототожнений з технічним пунктом. Розглянемо такі критерії:

- Точність роботи аналітики;
- Відсоток хибних тривог;
- Класифікація об'єктів;
- Стійкість до інформаційного шуму;
- Швидкість роботи аналітики;
- Гнучкість налаштування аналітики;
- Вимоги до обчислювальних ресурсів.

1.6.5. Функціональні критерії

Функціональні критерії відображають наскільки система відповідає вимогам конкретного об'єкту. Цей блок дозволяє не тільки провести оцінку формально, а й оцінити її практично за наступними критеріями:

- Відповідність основним задачам об'єкту;
- Покриття сценаріїв різних департаментів (служба безпеки, ІТ, маркетингологи, охорона праці);
- Адаптація до специфіка об'єкту;
- Зменшення навантаження на персонал;
- Автоматизація бізнес-процесів;
- Інтеграція з сторонніми сервісами;

- Оперативність розгляду та реакції на інциденти.

1.6.6. Ризикові критерії

Ризикові критерії визначають потенційні можливі ризики в роботі системи та наслідки, до яких вони можуть привести. Врахування таких критеріїв дозволяє заздалегідь визначити та підвищити рівень надійності тим самим зменшити даний параметр через такі критерії:

- Наявність єдиної точки відмови;
- Ризик пропуску інциденту;
- Ризик перевантаження обладнання;
- Ризик збільшення витрат на експлуатацію.

Таким чином представлені групи критеріїв можна описати в шість груп, які відображені в таблиці 1.2.

Таблиця 1.2.

Типи критеріїв

Група критеріїв	Що оцінює	Типові показники	Основне обмеження
Технічні	Ресурсну спроможність	канали, FPS, бітрейт, CPU/GPU, архів	не враховують економіку та функціональну корисність
Економічні	Вартість рішення	CAPEX, OPEX, TCO	не відображають технічну та функціональну якість

Експлуатаційні	Зручність супроводу	адміністрування, оновлення, ремонтпридатність	не дають інтегральної оцінки
Аналітичні	Якість AI/аналітики	точність, FP, FN	описують лише аналітичний компонент
Функціональні	Корисність для задач об'єкта	сценарії, автоматизація, час реакції	потребують зв'язку з технічними і економічними параметрами
Ризикові	Потенційні втрати та уразливості	відмови, втрата архіву, помилки детекції	важко оцінюються без комплексного методу

1.7. Постановка наукової проблеми та обґрунтування напряму дослідження

Аналіз існуючих підходів демонструє фрагментарність та загальну низьку ефективність щодо оцінки програмно-апаратного комплексу системи відеонагляду до впровадження та під час експлуатації. Частковий аналіз системи показує переваги та недоліки лише в рамках однієї або декількох групах критеріїв, але не розглядає їх в сукупності, що створює ризики до подальшого впровадження, експлуатації та модернізації систем, масштабування конфігурації і як наслідок економічні витрати. Крім оцінки таких параметрів варто врахувати, що забезпечення безпеки це інструмент який використовується користувачами, тобто власниками об'єктів, так і будується для забезпечення безпеки відвідувачів/працівників, а ігнорування його ефективності призводить до збільшення проблемних інцидентів.

Через відсутність можливості розрахувати інтегральний критерій ефективності неможливо акцентувати увагу на потенційні ризики та доцільність перегляду

альтернативних пропозицій або концепцій, адже відсутній єдиний уніфікований індикативний параметр. Як вже було визначено раніше, сучасна інтегрована система відеонагляду з елементами аналізу даних є складною та багаторівневою структурою в якій всі підсистеми та критерії тісно пов'язані між собою, тому оцінка такої системи має проходити за допомогою багатокритеріального підходу комплексно,

Створення нового методу оцінки ефективності обумовлено, визначеними в ході аналізу існуючих методів, факторами і має конкретну ціль - створити підхід, який дозволить врахувати всі необхідні складові та критерії під час проектування для раціонально правильного вибору та впровадження програмно-апаратного комплексу. Оцінювання комплексним підходом є найбільш ефективним та підходящим для сучасних систем, що відображено в таблиці 1.3.

Таблиця 1.3.

Методи оцінки ефективності, їх переваги та недоліки

Критерії порівняння	Метод 1: Оцінка за технічними характеристиками	Метод 2: Оцінка за економічними характеристиками	Метод 3: Оцінка за функціональними можливостями	Метод 4: Оцінка за експлуатаційними характеристиками	Метод 5: Оцінка за ризиками	Інтегральний комплексний метод
Технічна точність	±	±	-	±	-	+
Економічна доцільність	-	+	-	-	±	+
Простота в експлуатації	±	+	-	±	±	+
Точність аналітики	-	-	+	±	±	+
Функціональність	-	-	+	±	±	+
Масштабованість	±	-	±	-	-	+

Сумісність з іншими системами	-	-	±	±	±	+
Ризики відмов і надійність	±	-	±	+	+	+

Висновки до розділу 1

У першому розділі було проведено аналіз сучасних інтегрованих систем, архітектури побудови таких систем, інструментів, які використовуються для аналізу даних та існуючі підходи до оцінки конфігурацій, а також виявлено головну проблематику в існуючих методах.

Встановлено, що сучасні системи відеоспостереження є складними системами, які складаються з різних компонентів, до яких входить відеокамери, серверне обладнання, мережева інфраструктура, підсистеми безпеки, як системи контролю та управлінням доступу, охоронно пожежні сигналізації, архівні сховища та аналітичні інструменти. Як заключення не є правильним підходом розглядати сучасну систему відеонагляду і проводити оцінку її ефективності як кожен окремий технічний компонент або ж розглядати її доцільність до впровадження через призму виключно економічних бюджетних обмежень.

У результаті аналізу типів архітектур було визначено, що централізована, розподілена, гібридна архітектури та Edge/NVR і хмарні рішення мають власні переваги та обмеження до застосування для різних об'єктів за параметрами та вхідними даними. Визначено, що вибір архітектури є критично важливим етапом в проектуванні систем відеоспостереження і вона впливає на технічні показники ефективності та на економічну рентабельність під час впровадження та в процесі експлуатації. Визначено параметри, за якими можна оцінити кожну з представлених типів архітектур та виконано порівняння, для коректного визначення необхідного типу.

Дослідження елементів аналізу даних у сучасних інтегрованих системах безпеки доводить, що використання класичного підходу до експлуатації системи через візуальний аналіз не є ефективним та поступається у ефективності відносно сучасних інструментів відеоаналітики. Визначено, що існують різні інструменти, які відрізняють точністю та навантаженням на апаратні компоненти серверу. Використання аналітики на основі нейронних мереж є найбільш ефективним та точним для вирішення як простих

так і складних задач. В свою чергу це дозволяє точно ідентифікувати необхідні події, оперативно на них реагувати, зменшити витрати на персонал та економічні втрати на інциденти. Разом з тим було встановлено, що точність інструменту відеоаналітики не є єдиним важливим критерієм який впливає на загальну оцінку ефективності системи, а є лише одним з важливих індексів, який варто брати до уваги.

Проведений аналіз існуючих підходів оцінки ефективності системи показав, що існують підходи, які базуються на виявленні технологічних критеріїв, фінансових, бюджетних обмеженнях, точності аналітики або окремих компонентів системи, експертних оцінках інсталяторів та проектувальників. Було встановлено, що такі підходи не враховують всіх необхідних параметрів для повної оцінки системи і лише частково дозволяють оцінити окрему ефективність системи нехтуючи іншими складовими.

Було класифіковано та визначено основні групи за якими може відбуватися коректна оцінка ефективності системи. Визначені групи за технічними, економічними, експлуатаційними, аналітичними, функціональними та ризиковими параметрами, дозволяють визначити ефективність кожного розділу системи та показати реальний показник ефективності системи комплексно, оскільки аналіз поточних методів показав, що вони вирішують цю задачу лише частково.

Як результат було встановлено, що в сучасних підходах відсутній комплексний підхід аналізу та оцінювання системи, який би дозволив врахувати основні фактори та обґрунтовано визначити ефективність та раціональність впровадження пропонованого рішення. Це підтверджує необхідність у формуванні уніфікованого комплексного методу оцінки ефективності інтегрованої системи відеонагляду з елементами аналізу даних для її оптимізації в подальшому.

РОЗДІЛ 2

РОЗРОБКА МЕТОДУ КОМПЛЕКСНОГО ОЦІНЮВАННЯ ТА ОПТИМІЗАЦІЇ ІНТЕГРОВАНОЇ СИСТЕМИ ВІДЕОСПОСТЕРЕЖЕННЯ З ЕЛЕМЕНТАМИ АНАЛІЗУ ДАНИХ

2.1. Формування концепції ефективності інтегрованої системи відеоспостереження

Традиційний підхід до проектування систем відеоспостереження напряду залежала від оцінювання апаратних компонентів системи, тоді як програмна складова ігнорувалась та не бралась в розрахунок через сепарацію підсистем безпеки. Програмне забезпечення зазвичай виступало як додатковий інструмент для роботи з даними окремої підсистеми, як відеонагляд, система контролю та управлінням доступу, охоронно пожежна сигналізація, система звітності. Ефективними показниками було визначення роздільної здатності камери, кількості кадрів та кодек, який міг використовувати процесор, обсяг системи зберігання даних для архіву, рівень його відмовостійкості, кількість датчиків або зчитувачів, які можуть бути підключені на один контролер, швидкість обробки даних.

Такий підхід змінився, коли на основі PSIM (Physical Security Information Management) стало можливим об'єднувати підсистеми безпеки в одне інформаційне середовище та працювати з його компонентами використовуючи один спільний та уніфікований інтерфейс програмного забезпечення. Таким чином еволюція програмного забезпечення стала на один рівень з апаратними складовими та їх ігнорування зменшувало ефективність програмно-апаратного комплексу інтелектуальної системи відеоспостереження. Відтепер сучасний підхід значно збільшує зручність у експлуатації системи безпеки, потрібно менше обладнання, кваліфікації персоналу та уваги для роботи з системою. Для операторів та адміністраторів такий підхід дозволяє більше не

вивчати декілька системи, які розділені між собою напрямком безпеки, а працювати в єдиному інтерфейсі програмного забезпечення де знаходяться налаштування та вікна взаємодії з подіями. Такий підхід також знімає навантаження на персонал та збільшує концентрацію, оскільки все що бачить спеціаліст для нього стандартне та побудоване під ефективне використання.

Інтеграція аналітичних детекторів також на пряму збільшило вагу програмного забезпечення, оскільки використання аналітичних інструментів суттєво спрощувало задачі обробки даних особливо на об'єктах масштабних та з великою кількістю камер. Різні виробники пропонують свої рішення, які відрізняються за ціною, набором таких аналітичних функцій та їх точністю. Варто зазначити, що крім функціональних можливостей аналітики особливо важливу роль відіграє можливість опрацювання цих даних та маніпуляція з ними. Необхідно враховувати, які інструменти існують для роботи з метаданими, як вони реалізовані та як вони можуть функціонально застосовувати на об'єкті.

Окрема ніша, яка зазнала особливої уваги стала системи звітності та дашборди, наявність яких в програмному забезпеченні має особливо важливу роль. Якщо раніше такі дані оброблювались в окремих спеціалізованих сервісах як Power-BI, то зараз доступність вбудованої системи звітів є необхідним інструментом через велику кількість параметрів та змінних, які присутні в програмному забезпеченні. В свою чергу відсутність необхідних звітів робить систему менш зручною, оцінка обсягів метаданих виконується не тільки операторами в традиційному підході, а й через призму групування даних та їх представлення в множині через призму часу. Як заключення деякі системи мають широкий набір інструментів, але не мають методів їх представлення, а це в свою чергу створює прецедент зменшення ефективності, залучення сторонніх сервісів, збільшення витрат та збільшення термінів на повне впровадження проекту.

Незважаючи на такий об'єм функцій та вимог, які генеруються паралельно до активного технологічного розвитку існуюча нормативна та методична база для проектування ІСВ залишається на рівні одновимірних оцінок, які не включають

необхідний набір критеріїв. Крім того, існує також маркетинговий фактор, який активно використовується виробниками подібних рішень, що зміщують центр уваги на ті функції, які можуть бути теоретично корисними, але на практиці не застосовуються, а ті функції, які відсутні в системах, приховуються і навіть представлені в неефективному світлі через що оцінити реальну ефективність можна тільки емпірично. Практичний підхід також дає свої переваги при тестуванні показників точності та зручності запропонованої системи, оскільки такі показники також на практиці є завищені або ж узагальнені і не мають справжнього дослідницького підґрунтя, що в результаті впливає на загальне ефективність та очікування від роботи системи.

Однак, як показує практика таких підходів, вони також є неповноцінними і залишають невирішеними інші питання економічного вектору. Крім того існує пряма залежність від кількості інтелектуальних інструментів та збільшенні апаратних ресурсів, які також потрібно враховувати завчасно. Інакше кажучи, при проектуванні таких систем потрібно мати чітку лінійну кореляцію між вхідними даними та результатом, що дозволить математично оцінити ефективність системи.

Сучасна ICB є складної ієрархічною системою, яка агрегує та оброблює дані, а тому оцінка такої системи не може залежати від одного параметру. Отже ефективність такої системи має бути оцінена через визначені групи параметрів ефективності та представлені окремо і лише тоді вони можуть сформувати цілісний показник загальної ефективності системи. Варто зазначити, що такий інтегральний показник не має бути представлений самотійно, а має супроводжуватись переліком декількох груп, з яких це значення складається. Такий підхід обумовлений тим, що різні типи об'єкту мають свою власну специфіку та вимоги до реалізації. Можна розглянути на прикладі різні за критичністю об'єкти, для яких рівень точності або пропуску подій різний і тоді даний параметр може бути зменшений або збільшений відповідно. Натомість інші показники також будуть змінені але з урахуванням такого запиту. У випадках коли інший параметр, наприклад економічний, обмежений, то потрібно мати також можливість впливу на цей

індекс та в лінійній залежності провести оновлений розрахунок ефективності та методом перерахунку визначити найбільш раціональний підхід.

Для подолання описаного протиріччя, що викладено в дисертаційному дослідженні запропоновано нову комплексну методологію оцінювання проекту та конфігурації ІСВ. В основі цієї концепції закладено декомпозицію поняття цілісної ефективності на три базисні складові – вектори. Кожна складова описує та характеризує окремий напрямок з точки зору поняття ефективності (рис. 2.1.). Такі вектори логічним чином розділені на:

1. Технічна ефективність.
2. Економічна ефективність.
3. Функціональна (операційна) ефективність.

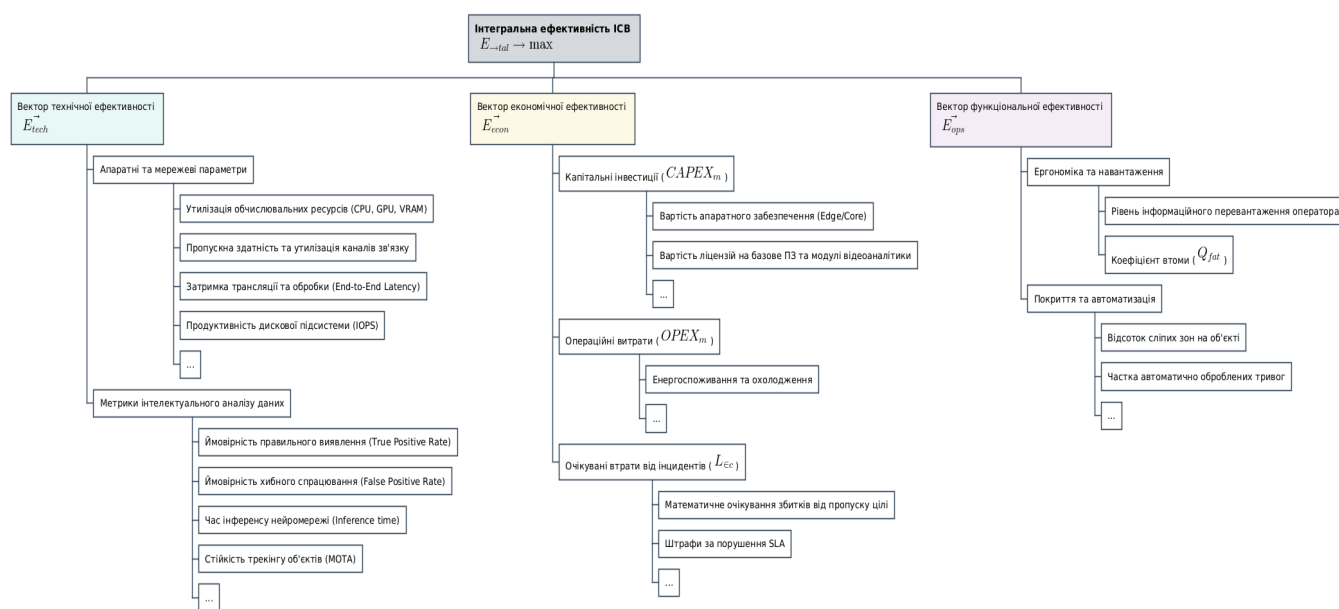


Рис.2.1. Структура комплексного методу оцінки ефективності.

Дані вектори між собою містять зв'язки, а саме тому при збільшенні кількості камери ми матимемо більший індекс ефективності але при цьому буде змінено значення економічної ефективності та функціональної. Такий підхід дозволить визначити

найбільш якісне значення, яке буде відповідати вимогам проекту з точки зору безпеки, фінансових затрат та якості управління.

2.1.1. Вектор технічної ефективності

Даний індекс ефективності описує наскільки система ефективна при використанні закладених технічних компонентів та інструментів. Визначається кількість камер, їх параметри роботи (FPS, Bitrate, Compression Codec), пропускну здатність мережі, навантаження на обчислювальні ресурси, об'єм даних для зберігання та необхідний масив дисків для їх збереження.

Для аналітичних інструментів визначаються конкретні задачі та вимоги, тип аналітики, кількість та необхідна точність роботи детекторів. Оскільки на ефективність також впливає критичність об'єкту то варто вносити коригування на вагу такої змінної, для якої параметр точності або ймовірність відсутності розпізнавання є особливо критичною.

Для якісного обрахунку такого комплексу змінних необхідно мати вхідні дані щодо об'єкту, оскільки вони впливають на вибір інструментів, які будуть застосовані в розрахунку. За відсутності таких даних, можна використовувати теоретичні показники, для визначення найбільш комфортного та допустимого рівня роботи системи. Для такого вектору призначена наступна змінна – E_{tech}

2.1.2. Вектор економічної ефективності

Економічний індекс, у контексті безпеки, є вирішальним. Як показало практичне дослідження та опитування, які проводились в рамках даної роботи більшість проєктів мають обмеженість бюджету або ж відсутнє правильне розуміння витрат. Якщо в окремих випадках бюджет не дозволяє використовувати більш якісні інструменти та обладнання і це зменшує значення технічної ефективності – це компроміс, який

приймається свідомо, але у випадках, коли проведена оцінка вартості володіння системою невірна або не включає в себе необхідні параметри – таке рішення, яке зменшує якість обладнання і відповідно загальну вартість на впровадження – хибне.

Класичні розрахунки, які проводяться також не для всіх проектів, показують, що використовується спрощена та неадаптована під даний сегмент формула сукупної вартості володіння (TCO — Total Cost of Ownership), яка в свою чергу складається з капітальних (CAPEX) та операційних витрат (OPEX). Такий підхід спрямований не на генерацію прибутку, а на мінімізацію витрат.

Відсутність врахування очікуваних втрат від інцидентів в класичному підході перетворює таку оцінку в збиткову і нераціональну. На цей параметр в більшій мірі впливає технічна ефективність і зазначені в ній індекси окремих аргументів. При використанні аналітики, яка має низьку точність і великий відсоток хибних детекцій, спостерігається зменшення фінансових затрат на рівні впровадження системи, але тим самим впливає на рівень витрат під час експлуатації, оскільки крім збільшення персоналу, для обробки даних, збільшується і кількість інцидентів, які прямо несуть збитки. Окрему увагу варто приділити репутаційними збитками, які для окремих категорій об'єктів є особливо важливими, як от наприклад банки, торгово-розважальні центри, житлові комплекси, безпечні міста. Для репутаційних випадків неможливо передбачити такий ризик математично, але враховувати таку ймовірність потрібно, так як в момент інциденту може перетворитись на кризову точку для об'єкту. Додатково, для бюджетного обладнання потрібно передбачати коротшу періодичність обслуговування та ремонт, адже на таке обладнання надається гарантія менше ніж на обладнання з вищого цінового сегменту (в середньому 1 рік на бюджетні камери та від 3х років на якісні зразки).

Крім обладнання наразі враховується вартість програмного забезпечення, яке за перевіркою комерційних специфікацій складає від 10 до 20% вартості всього проекту. Саме програмне забезпечення також впливає і на вартість обладнання в залежності від об'ємів аналітики, які планується використовувати. Вартість аналітики є високою і

вимагає також відповідно продуктивного обладнання. З огляду на практичну оцінку в проектах це і слугує точкою відмови від технічної ефективності в бік економічної, але як було зазначено – це вірно тільки на етапі впровадження. Сучасна аналітика є більш ефективною і дозволяє простіше та швидше опрацьовувати складні сцени. В окремих випадках інвестиція в такі системи дозволяє скоротити штат працівників, збільшивши при цьому ефективність. Такий зв'язок не завжди прямий, але економічно утворюється рентабельність, яка зменшує терміни окупності програмно-апаратного комплексу.

Підсумовуючи описані критерії, фінансові розрахунки ефективності можуть бути правильно передбачувані в розрізі часу. Для такого індексу ефективності призначена наступна змінна – E_{ecom} .

2.1.3. Вектор функціональної ефективності

Даний складовий елемент складається з характеристики ергономіки взаємодії з системою оператором. Навіть при роботі з самими точними аналітичними інструментами завжди існує ймовірність хибних тривоги або ж інцидентів які вимагають втручання людини. Тому варто враховувати наскільки довго зберігатиметься ефективність оператора при використанні системи, наскільки швидко відбуватиметься реагування на події, пошук необхідних інцидентів та їх обробка.

Закладається фактор втоми, який повністю ігнорується в існуючих підходах, але який прямо впливає на ефективність системи і на створення проблемних інцидентів які прямопропорційно впливають на економічні витрати. За умови оптимізації та побудови ефективної системи в окремих випадках робота оператора значно зменшується і його участь необхідна лише в критичних ситуаціях. При такій концепції зменшується витрати на персонал, його кваліфікацію і зменшується відповідальність та ризики недобросовісної поведінки працівників, які також входять до факторів економічних витрат. Звільнений фокус оператора також дозволяє швидше та ефективніше визначати реальні загрози та передбачати їх.

Високий рівень ефективності даного вектору забезпечується також рівнем автоматизації, за допомогою якої і створюються процеси спрощення та миттєвої реакції. В особливо складних сценаріях це може бути найефективнішим інструментом, оскільки ряд випадків для людини можуть супроводжуватись стресом і виконання декількох завдань одночасно стає неможливим. Тому такі задачі варто перекласти на автоматизований сценарій, який виконає це автоматично за декілька секунд. Для такої функціональної ефективності використовується змінна - E_{ops} .

2.1.4. Математична постановка концепції

Відповідно до розробленої концепції, проект інтегрованої системи відеоспостереження, який необхідно проаналізувати, правильно буде формалізувати як точку у багатовимірному просторі критеріїв. Оцінюючі традиційні підходи, які опираються на оцінку окремих параметрів, як вартість, технічна складова, ефективність роботи аналітики або кількість камер, не здатні знайти реальний показник. Саме тому загальна інтегральна ефективність E_{total} буде визначена як складна цільова функція, що об'єднує три визначені вектори: технічний, економічний та функціональний.

Математично таку просторову залежність можна подати у вигляді формули (2.1):

$$E_{total} = \Phi(\overrightarrow{E_{tech}}, \overrightarrow{E_{econ}}, \overrightarrow{E_{ops}}), \quad (2.1)$$

де Φ – оператор агрегації критеріїв у єдиний скалярний показник.

Використання такого оператора є критично необхідним, оскільки компоненти векторів ефективності мають різні одиниці виміру (відсотки, секунди, гривні, мегабайти). В такому випадку, оператор Φ включає в собі математичні процедури з нормування таких одиниць до безрозмірного стандартизованого вигляду, а також враховувати пріоритетність кожного вектору для кожного окремого об'єкту. Це обумовлено тим, що для об'єктів з високим рівнем критичності вага вектору технічної

ефективності має значний пріоритет ніж економічний вектор. У випадку з проектом типу NoReCa, такі вектори можуть бути паритетні або ж протилежні до попереднього прикладу.

Нехай $S = \{s_1, s_2, \dots, s_n\}$ – це множина можливих апаратно програмних рішень систем відеоспостереження. Така множина буде включати в себе різні варіації топології, типів аналітики, точності роботи детекторів та серверів. В такому випадку $s_i \in S$ буде характеризуватись власним унікальним набором векторів : $E_{tech}(s_i)$, $E_{econ}(s_i)$, $E_{ops}(s_i)$.

Оскільки процес розробки, впровадження та модернізації системи завжди відбувається в умовах де наявні інфраструктурні або ж фінансові обмеження, які ігнорують неприйнятні варіанти, які не відповідають запиту, то вони формують область допустимих рішень Ω .

До основних граничних умов належать:

1. Фінансові обмеження (2.2):

$$B(s_i) \leq B_{\max} , \quad (2.2)$$

де $B(s_i)$ - сукупна вартість розгортання та володіння конфігурацією s_i , яка включає в себе як капітальні так і операційні витрати, а $B_{\max}$ - це максимально допустимий бюджет.

2. Точність аналітики (2.3):

$$\text{Accuracy}(s_i) \geq \text{Accuracy}_{\min} , \quad (2.3)$$

де $\text{Accuracy}(s_i)$ – це точність роботи аналітики в конкретній конфігурації s_i , а $\text{Accuracy}_{\min}$ – мінімально допустимий рівень точної для ефективної роботи системи.

3. Інфраструктурні обмеження (2.4):

$$R_{\text{net}}(s_i) \leq R_{\max} , \quad (2.4)$$

де $R_{net}(s_i)$ – є необхідна пропускна здатність каналів зв'язку для передачі відеопотоку, а R_{max} – поточний доступний ресурс мережевої інфраструктури.

Якщо врахувати вказану концепцію підходів описаних вище, то задача багатокритеріальної оптимізації полягає в знаходженні раціональної конфігурації s_{opt} , яка забезпечує максимальну функцію інтегральної ефективності E_{total} на множині всіх допустимих рішень Ω при виконанні заданих обмежень (2.5):

$$s_{opt} = \arg \max_{s_i \in \Omega} \Phi \left(\overrightarrow{E_{tech}}(s_i), \overrightarrow{E_{econ}}(s_i), \overrightarrow{E_{ops}}(s_i) \right), \quad (2.5)$$

за таких умов (2.6):

$$B \leq B_{max}, \text{Accuracy}(s_i) \geq \text{Accuracy}_{min}, R_{net} \leq R_{max}, \quad (2.6)$$

Сформульована математична постановка задачі має необхідне теоретичний базис для вибору критеріїв та математичної формалізації системи, а також процедури нормування та згортання.

2.2. Визначення системи критеріїв оцінювання методу ефективності

Для визначення та оцінки ефективності системи відеоспостереження необхідно мати повний перелік критеріїв які впливають на це значення та дозволяє за різних значень порівняти окремі конфігурації. Такий підхід дозволяє прийняти рішення опираючись на фактичні значення, та відслідковувати їх залежність. У цьому підрозділі визначаємо необхідні критерії для побудови комплексного оцінювання що розділяється на технічні, економічні та функціональні категорії.

2.2.1. Класифікація критеріїв

Оцінювання системи буде відбуватись за критеріями, які класифікують їх відповідно до категорії:

- Технічні критерії
- Економічні критерії
- Функціональні критерії

Такий поділ обумовлений не лише логікою групування за типом, а й з ціллю визначення загального значення ефективності конкретного аспекту системи в залежності від постановки задачі. Саме тому ці критерії спершу визначаються окремо і вже потім формують єдиний інтегральний показник ефективності.

2.2.2. Опис технічних критеріїв

Технічні критерії визначають технічну ефективність системи, яка залежить від точності роботи аналітики, апаратних та програмних компонентів, що в свою чергу дозволяє зробити висновок наскільки якісним буде конкретне рішення.

Перший критерій, який буде розглянутий - точність виявлення (Detection Accuracy), який визначає відсоток правильного розпізнавання об'єктів аналітичною системою за одиницю часу. Цей показник є критично важливий для оцінки функціональності системи.

Другий критерій – частка хибних спрацювань (False Positive Rate), що визначає відсоток хибних спрацювань детектору, від загальної кількості задетектованих подій. Відповідно чим такий показник менше, тим точніше працює аналітичний інструмент.

Наступний критерій – частка пропуску подій (False Negative Rate), який визначає відсоток подій, які були пропущені системою, тобто коли в полі зору камери був шуканий об'єкт, але система його не зафіксувала.

Затримка обробки (Latency), показник, який визначає необхідний для оператора час на отримання та опрацювання інциденту.

Пропускна здатність мережі описує пропускну здатність каналу для передачі даних через канали зв'язку.

Навантаження на мережу фіксує необхідну пропуску здатність в залежності від кількості камер, які працюють одночасно та яке навантаження вони дають на мережу.

Навантаження на CPU/GPU визначає рівень завантаженості обчислювальних одиниць задачами системи (обробка відео, аналітика).

Масштабованість як параметр оцінює здатність системи до розширення, збільшення відеокамер та покладених задач на систему.

Надійність показує наскільки система використовує якісне обладнання, а також її час роботи без збоїв.

Відмовостійкість характеризує чи має об'єкт наявність резервування (живлення, мережа, архів).

2.2.3. Опис економічних критеріїв

Економічні критерії дозволяють провести оцінку вартості системи до впровадження, вартість її експлуатації, витрати на персонал та вартості інцидентів, тобто ряд збитків, які система може потенційно понести з вибором тих чи інших параметрів.

Початкові загальні витрати показують ціну системи на етапі впровадження, що включає в себе відеокамери, сервери, мережу, монтаж.

Експлуатаційні витрати характеризують витрати на обслуговування системи, технічну підтримку і ремонт.

Вартість масштабування вказує скільки потрібно фінансових ресурсів, щоб можна було розширити можливості системи відповідно до поточної конфігурації як її частка.

Вартість за один канал показує ціну за роботу одної камери в рамках повної системи, що включає в себе не тільки камеру або ліцензію, а й сервер та монтаж.

Вартість за одну корисну детекцію визначає вартість визначеного інциденту, що необхідно для оцінки ефективності системи.

Ефект зниження витрат показує наскільки система ефективна і дозволяє зменшити фінансові затрати через відсутність інцидентів.

Умовний термін окупності визначає час потрібний для покриття інвестицій.

2.2.4. Опис функціональних критеріїв

Функціональні критерії дозволяють оцінити функціональну спроможність системи, враховуючи здатності до виконання поставлених задач ефективно та зручно.

Покриття задач безпеки (Security Coverage) визначає, наскільки система здатна вирішити поставлені перед нею задачі, присутність в переліку інструментів необхідних для якісного функціонування, адміністрування та експлуатації ICB.

Релевантність аналітики задачам об'єкту (Relevance of Analytics) є показником, який визначає чи аналітика, яка була запропонована в рамках даного проекту, відповідає його вимогам та чи покриває вона необхідні задачі.

Зменшення навантаження на операторів (Operator Load Reduction) показник, який оцінює наскільки запропонована система зменшує втручання оператора в роботу системи. На цей параметр впливає аналітика, автоматизація та покриття задач безпеки.

Підвищення оперативності реагування (Response Time Improvement) - оцінка, яка визначає наскільки оперативно відбувається реакція системи та персоналу на інцидент.

Рівень автоматизації (Automation Level) – параметр, який значним чином впливає на всі попередні, оскільки характеризує можливість роботи системи працювати автономно, тобто без втручання людини, в робочі процеси. Такий рівень автоматизації та самі сценаріями мають бути зафіксовані на початку проекту, оскільки вони мають трудоемкість, яку можна виразити в бюджеті, а також оптимізує витрати на персонал.

Визначені критерії, які застосовуються для оцінювання, відображені у відповідній таблиці 2.1.

Таблиця 2.1

Перелік критеріїв та їх параметри

Тип критерію	Критерій	Одиниця вимірювання
Технічні	Точність виявлення	%
	Частка хибних спрацьовувань	%
	Частка пропусків подій	%
	Затримка обробки	мс
	Пропускна здатність	Канали/с
	Навантаження на мережу	Мбіт/с
	Навантаження на CPU/GPU	%
	Масштабованість	Канали/ресурси
	Надійність	Години
	Відмовостійкість	%
Економічні	Початкові витрати	\$
	Експлуатаційні витрати	\$/місяць
	Вартість масштабування	\$
	Вартість на один канал	\$
	Вартість на одну корисну детекцію	\$
	Ефект зниження втрат	\$
	Умовний строк окупності	роки
Функціональні	Покриття задач безпеки	%
	Релевантність аналітики задачам об'єкта	%
	Зменшення навантаження на операторів	%
	Підвищення оперативності реагування	с
	Рівень автоматизації	%

2.2.5. Визначення ваги для критеріїв

Враховуючи специфіку та вимоги кожного проекту інтегрованих систем відеоспостереження, важливо передбачити можливість встановлення пріоритетності для групи критеріїв, що визначають, який з груп параметрів буде мати найвищий показник ефективності та якості. Для такого підходу будемо використовувати визначення вагів для кожного з критеріїв. Визначаємо відповідні ваги як w_{tech} , w_{econ} , w_{ops} .

2.2.6. Структура критеріїв та їх поєднання

Отримані в процесі математичних обчислень значення потрібно звести в одну систему вимірювання для чого буде використовуватись метод нормалізації та зважування. Оскільки ми маємо параметри, які оцінюються різними величинами без такого процесу буде неможливим їх комплексний інтегральний розрахунок. Зваженого середнього або багатокритеріальної оптимізації – це саме ті методи за якими буде відбуватись приведення показників до безрозмірного вигляду. Таким чином ми зможемо комбінувати значення в одну інтегральну функцію і отримати показники ефективності. Зазначені методи є найбільш поширеними та актуальними для обчислення інтегральної оцінки ефективності. Підсумовуючи, сформована система критеріїв оцінювання є базисом для формування комплексної оцінки ефективності ІСВ. Застосування визначених критеріїв дозволяє провести математично точне та об'єктивне порівняння різних систем.

2.3. Побудова інтегрального критерію ефективності

Визначення інтегрального критерію ефективності дозволяє оцінити через конкретний показник ефективності заданої системи, яка складається з декількох груп критеріїв оцінювання. Таке значення набуває своєї ефективності при порівнянні

декількох концепцій або пропозицій, які, за допомогою даного методу, можуть бути математично розраховані і вихідні значення інтегральної ефективності порівняти між собою.

Оскільки інтегральна ефективність визначається трьома критеріями, то і визначатись він буде як функція від них (2.6).

$$E_{total} = \Phi(E_{tech}, E_{econ}, E_{ops}), \quad (2.6)$$

де:

- E_{total} - інтегральна ефективність системи;
- E_{tech} - технічна ефективність;
- E_{econ} - економічна ефективність;
- E_{ops} - функціональна ефективність.

Оскільки було визначено, що критерії оцінювання мають різні одиниці виміру, для роботи методу потрібно їх нормалізувати. Таким чином всі показники матимуть значення від 0 до 1, та відбувається за допомогою методу мінімум-максимум (2.7).

$$E_{norm} = \frac{E - E_{min}}{E_{max} - E_{min}}, \quad (2.7)$$

де:

- E - значення критерію;
- E_{min}, E_{max} - максимальні та мінімальні значення даного критерію.

Визначення вагових значень критеріїв необхідний, аби визначити, які критерії в запропонованій системі мають найбільшу цінність, тобто, іншими словами - пріоритет. За допомогою цього підходу можна визначити критичність кожного з них і відобразити її математично. В результаті вибору необхідних показників це прямопропорційно

впливає і на інтегральний показник не зважаючи на те, що окремі показники критеріїв ефективності незмінні (2.8).

$$E_{total} = w_{tech} \cdot E_{tech} + w_{econ} \cdot E_{econ} + w_{ops} \cdot E_{ops} , \quad (2.8)$$

де:

- E_{total} - інтегральний критерій ефективності;
- $E_{tech}, E_{econ}, E_{ops}$ - нормалізовані значення критеріїв.

Для забезпечення математично коректного результату на вагові коефіцієнти буде накладено умови нормування, оскільки вони є обмеженими одиницями в рамках заданого діапазону ефективності (2.9).

$$\sum_{i=1}^3 w_i = 1, w_i > 0 , \quad (2.9)$$

Варто також встановити обмеження інтегральної оцінки, оскільки такий підхід дозволяє врахувати існуючі обмеження проекту з огляду на конкретні характеристики, як точність аналітики, фінансові обмеження та інфраструктурні обмеження (2.10, 2.11, 2.12).

$$B(s_i) \leq B_{max} , \quad (2.10)$$

де:

- $B(s_i)$ — вартість конфігурації s_i ;
- B_{max} — максимально допустимий бюджет.

$$T_{accuracy}(s_i) \geq T_{accuracy-min} , \quad (2.11)$$

де:

- $T_{\text{accuracy}}(s_i)$ — точність аналітики для конфігурації s_i ;
- $T_{\text{accuracy-min}}$ — мінімально допустимий рівень точності.

$$R_{\text{net}}(s_i) \leq R_{\text{max}}, P_{\text{comp}}(s_i) \leq P_{\text{max}}, \quad (2.12)$$

де:

- $R_{\text{net}}(s_i)$ — необхідна пропускна здатність мережі для конфігурації s_i ;
- R_{max} — максимально можлива пропускна здатність мережі;
- $P_{\text{comp}}(s_i)$ — обчислювальна потужність для конфігурації s_i ;
- P_{max} — максимальний рівень обчислювальних потужностей.

Для випадків, де вибір може бути досконалим для одного критерія, але неможливим для іншого необхідно створити систему логічних фільтрів. В такому випадку якщо умова порушується, то загальний показник ефективності буде дорівнювати нулю (2.13).

$$E_{\text{total}} = \begin{cases} \sum w_i E_i, & \text{якщо } B(s_i) \leq B_{\text{max}} \\ 0, & \text{якщо } B(s_i) > B_{\text{max}} \end{cases}, \quad (2.13)$$

Таким чином, інтегральний метод оцінки ефективності за декількома критеріями набуває математичного підґрунтя. В наступному підрозділі буде розроблений метод оптимізації для даної ІСВ.

2.4. Розробка методу оптимізації

Розробка методу оптимізації інтегрованої системи відеоспостереження з елементами аналізу даних є важливим етапом, який дозволяє підібрати за допомогою математичного підходу, а саме з врахуванням критеріїв ефективності і принципу обмеження. Таким чином створюється область можливих множин, які можна дослідити

та порівняти і за результатами розрахунку обрати найбільш оптимальну, для конкретного проекту, конфігурацію систему по співвідношенню актуальних важливих параметрів. Варто зазначити, що даний метод простий в застосуванні та не потребує глибинних галузевих знань, що робить його цінними для співробітників будь-якого рівня компетенції.

Метод оптимізації містить наступну послідовність етапів, що представлені на блок-схемі (рис 2.2), для знаходження оптимуму:

1. Збір вхідних даних;
2. Фільтрація допустимих рішень;
3. Оцінка критеріїв ефективності;
4. Інтегральна оцінка ефективності;
5. Вибір оптимальної конфігурації;
6. Виведення параметрів оптимальної конфігурації.

Алгоритм вибору оптимальної конфігурації

Рис. 2.2. Блок-схема етапів роботи методу оптимізації

2.4.1. Збір даних

Перший крок, для застосування методу оптимізації є збір вхідних даних в конкретному числовому значення або ж діапазоні. Цей етап дозволяє визначити основу для подальшого оцінювання та прийняття рішення. До вхідних даних відноситься:

1. Множина альтернатив $S = \{s_1, s_2, \dots, s_n\}$ - це список всіх можливих конфігурацій системи. Обов'язковим є передача діапазону значень для допустимих конфігурацій, які коригуються відповідно до базових заданих параметрів та розрахованій технічній характеристиці, економічних і також функціональній.

2. Граничні обмеження $B_{\max}, T_{\max}, R_{\max}$ - показники, які вказують обмеження по бюджету, точності аналітичних інструментів та пропускній здатності мережі. Використовуючи ці параметри відбуватиметься фільтрація непридатних конфігурацій, які не будуть розглянуті.

3. Вагові коефіцієнти $w_{\text{tech}}, w_{\text{econ}}, w_{\text{ops}}$ - визначаються коефіцієнти, які відображають пріоритизацію кожного критерія для конкретного об'єкту.

2.4.2. Фільтрація допустимих рішень

Після завершення збору і обробки вхідних даних відбувається етап фільтрації конфігурацій. Конфігурації, які не відповідають хоча б одному з обмежень не розглядаються і не потрапляють в список допустимих рішень Ω . Для кожної конфігурації перевіряються такі умови:

- Фінансові обмеження;
- Технічні обмеження;
- Інфраструктурні обмеження.

2.4.3. Оцінка критеріїв ефективності

Для кожної допустимої конфігурації розраховуються технічні, економічні та функціональні критерії ефективності. На даному етапі важливо мати саме групові показники ефективності. Для технічних критеріїв оброблюються такі значення як:

- Точність аналітики;
- Частка хибних спрацювань;
- Затримка в обробці даних.

Економічні критерії:

- Початкові витрати на впровадження системи;
- Вартість системи з розрахунку на 1 канал відео;
- Експлуатаційні витрати на систему.

Функціональні критерії включають в себе:

- Покриття задач безпеки;
- Оперативність реагування;
- Рівень автоматизації процесів.

Для нормалізації цих параметрів використовується метод мінімальних та максимальних значень (2.14).

$$E_{\text{norm}} = \frac{E - E_{\min}}{E_{\max} - E_{\min}}, \quad (2.14)$$

2.4.4. Інтегральна оцінка ефективності

Після нормалізації кожного значення відбувається їх зважування відповідно до вагових коефіцієнтів (2.15).

$$E_{\text{total}} = w_{\text{tech}} \cdot E_{\text{tech}} + w_{\text{econ}} \cdot E_{\text{econ}} + w_{\text{ops}} \cdot E_{\text{ops}}, \quad (2.15)$$

2.4.5. Вибір оптимальної конфігурації

Після того, як було визначено інтегральні оцінки для всіх конфігурацій відбувається процес ранжування за найвищим інтегральним показником. Визначається такий критерій за наступною формулою (2.16):

$$s_{\text{opt}} = \arg \max_{s_i \in \Omega} E_{\text{total}}(s_i), \quad (2.16)$$

В результаті конфігурація, яка містить найбільший інтегральний показник, буде визначена, як найбільш оптимальна. Такий вибір обумовлений розрахунком та врахуванням локальних змінних ефективності і вагових коефіцієнтів.

2.5. Формалізація правил вибору конфігурації

Формалізація правил вибору конфігурації є важливим етапом у методі оптимізації інтегрованої системи відеоспостереження. Формування правила вибору забезпечує систематичне та структуроване прийняття рішення, яке засноване на використанні різних критеріїв та обмежень, які визначені на попередніх етапах. Даний етап дозволяє перейти від оцінки розглянутих варіантів до безпосереднього вибору найбільш оптимальної конфігурації.

2.5.1. Оцінка критеріїв ефективності

Для кожної конфігурації, які потрапляють до списку допустимих конфігурацій, виконується обчислення інтегрального критерію ефективності, що враховує технічний, економічний та функціональні критерії. Такий розрахунок виконується з використанням нормалізації значень, який відображений в наступному вигляді (2.17):

$$E_{\text{total}} = w_{\text{tech}} \cdot E_{\text{tech}} + w_{\text{econ}} \cdot E_{\text{econ}} + w_{\text{ops}} \cdot E_{\text{ops}} , \quad (2.17)$$

де:

- E_{total} — інтегральний критерій ефективності для конфігурації,
- $w_{\text{tech}}, w_{\text{econ}}, w_{\text{ops}}$ — ваги для кожного з критеріїв,
- $E_{\text{tech}}, E_{\text{econ}}, E_{\text{ops}}$ — значення технічних, економічних та функціональних критеріїв, відповідно.

2.5.2. Порівняння альтернатив

Після виконання обчислення інтегрального значення ефективності кожної конфігурації ІСВ, значення проходять взаємне порівняння. Конфігурації, які мають найбільше значення інтегрального показника пропонуються як найбільш оптимальні альтернативи.

2.5.3. Обмеження на вибір конфігурації

Даний етап є один з найважливіших етапів, оскільки в ньому відбувається оптимізація обробки даних за заданими параметрами, що дозволяє не аналізувати альтернативні конфігурації, які не задовольняють наші вхідні параметри. Цей принцип працює для визначених критеріїв і визначає наступне. Для фінансових обмежень максимальна вартість конфігурації B_{max} не має перевищувати визначену суму бюджету. Для технічних обмежень кожна конфігурація має відповідати мінімальним вимогам щодо точності роботи аналітичних інструментів T_{accuracy} , пропускну здатності мережі R_{net} , а також загальне навантаження на систему. Інфраструктурні обмеження ж вимагають, щоб конфігурація відповідала доступним обчислювальним потужностям та вимогам існуючої мережевої інфраструктури.

На цьому етапі також відбувається відкидання альтернатив за правилом, що якщо конфігурація не задовольняє вимоги хоч одного з визначеного обмеження – вона буде відкинута.

2.5.4. Розрахунок і вибір найкращої конфігурації

По завершенню процесу фільтрації та порівняння альтернативних конфігурацій, наступний етап – це вибір найкращої конфігурації з опрацьованих масивів конфігурацій. Вибір здійснюється на основі максимізації інтегрального критерію ефективності (2.18):

$$s_{opt} = \arg \max_{s_i \in \Omega} E_{total}(s_i), \quad (2.18)$$

де:

- s_{opt} — оптимальна конфігурація, яка забезпечує максимальну ефективність;
- $E_{total}(s_i)$ — інтегральний критерій ефективності для кожної конфігурації.

2.5.5. Формалізація правил пріоритетів

У випадку, якщо декілька конфігурацій мають однакове значення інтегрального критерію ефективності, тоді необхідно застосовувати правило пріоритетів на основі яких буде обрана оптимальна конфігурація. Такі пріоритети можуть бути в залежності від різних параметрів які в себе включають тип об'єкту, його критичність, важливість якості та точності аналітичних інструментів, або ж навпаки низька вартість впровадження чи високий рівень ефективності автоматизації.

Результатом вибору буде обрана конфігурація, яка містить повний набір вхідних параметрів для проекту. Це означає, що містяться дані про технічні параметри, економічні параметри та функціональні.

2.6. Наукова інтерпретація методу

В даному підрозділі ми розглядаємо наукове обґрунтування розробленого методу оптимізації інтегрованих систем відеоспостереження з елементами аналізу даних з використанням методу оцінки ефективності. Варто підкреслити, що зазначений підхід є унікальним та має значні переваги в порівнянні з існуючими підходами. Концепція цього підходу будується на складних оцінках конфігурації, для якої на першому етапі застосовується розроблений метод оцінки ефективності за множиною критеріїв, а на другому етапі відбувається процес оптимізації. Даний процес може бути циклічний до моменту поки не буде знайдено найкраще значення з перебором діапазону параметрів вхідних даних. Такий принцип методу, який запропонований в роботі, розвиває та удосконалює конфігурацію ІСВ і дозволяє скоротити небажані негативні ефекти для різних груп ефективності: технічна, економічна, функціональна. Описаний підхід важливий для створення ефективних та економічно обґрунтованих рішень.

2.6.1. Комплексний підхід до оптимізації

У порівнянні з традиційними методами оптимізації, які спрямовані на аналіз та поліпшення одного конкретного аспекту системи безпеки, як наприклад точність або бюджет впровадження, запропонований метод виконує інтеграцію груп ефективності, які наявні для всіх систем, та описує для кожного з них детальні критерії ефективності кожного параметру. Описаний метод об'єднує показники в єдиний комплексний підхід, що дозволяє створити точну картину ефективності системи.

Традиційні методи, які орієнтуються на невелику низку факторів роботи системи та її ефективності не враховують лінійний зв'язок між іншими параметрами, а саме тому вони ігноруються, що призводить до збільшення збиткових наслідків для технічної групи, економічної або ж функціональної. Описуючи дану проблематику сформованим методом оптимізації – нормований параметр ефективності, який був визначений та

розрахований неповноцінно і має не найоптимальніші показники на початку зазнає зменшення.

Підхід розроблений в рамках даної роботи враховує зв'язок між різними характеристиками і робить його універсальним інструментом для використання на різних етапах проектування системи, впровадження, експлуатації або модернізації.

2.6.2. Ваги критеріїв та гнучкість методу

Запропонований метод включає в розрахунок вагові коефіцієнти для кожного з визначених критеріїв, що дозволяє тим самим адаптувати оптимізацію під специфічні вимоги проекту. Вага кожного критерію залежна від пріоритетності якості його виконання, що дозволяє виконувати ранжування для об'єктів з різним параметром критичності, де якість виконання технічного критерію ефективності є першочерговою задачею. Інші проекти, для яких інші критерії мають більший пріоритет здатні адаптувати даний метод для визначення оптимальної конфігурації. Такий підхід дозволяє використовувати метод для різних етапів розвитку.

2.6.3. Адаптивність і універсальність методу

Інтеграція багатьох критеріїв в одну модель оцінки ефективності та оптимізації дозволяє значно підвищувати адаптивність методу у випадку змін вхідних величин проекту та зміни вектору розвитку. Врахування вагових коефіцієнтів, реагувати на зміну технічної складовою або економічної створює можливість масштабувати систему та ефективно застосовувати метод в різних за розмірами системах.

2.6.4. Математична точність і методологічна коректність

Оскільки в даному методі використовуються чіткі математичні формули, то результат їх обчислень дозволяє отримати високоточний результат та коректну оцінку ефективності. Для критеріїв використовується нормалізація за допомогою методу мінімального та максимального значення, а використання інтегрального показника ефективності забезпечує об'єктивну оцінку для вибору оптимально конфігурації ІСВ.

2.6.5. Наукове значення методу

Запропонований метод має значний науковий потенціал, оскільки він містить в своїй структурі принцип оцінки ефективності та метод оптимізації, що об'єднує ряд критично важливих критеріїв, які не були раніше інтегровані та між якими не було проведено лінійної залежності.

- Метод використовує багатокритеріальний метод оцінки ефективності та засовує оптимізацію з врахуванням специфіка інтегрованих систем безпеки;
- Метод забезпечує уніфікацію підходів для оцінки технічних, економічних та функціональних характеристик, що робить його універсальним;
- Метод є надзвичайно корисним для оптимізації конфігурацій систем, їх оцінки, підвищення ефективності та зменшенню витрат.

2.6.6. Вплив на практичне застосування

Практичне застосування методу дозволяє значно підвищити ефективність проектування та експлуатації систем відеоспостереження завдяки комплексному підходу до оцінки декількох різних критеріїв ефективності. Саме тому, такий метод, дозволяє:

- Знизити витрати;

- Підвищити ефективність;
- Оптимізувати конфігурацію.

Як висновок, запропонований метод оптимізації і оцінки ефективності є комплексним, що в свою чергу дозволяє ефективно обрати оптимальну конфігурацію системи відеоспостереження на основі вхідних даних, враховуючи всі можливі критерії та обмеження. Даний метод також володіє адаптивністю і може бути адаптований відповідно до змін, що дозволяє аргументовано приймати рішення. Універсальність методу робить його підходящим для застосування у широкому спектрі об'єктів які відрізняються за масштабом, задачами, бюджетом.

Цей метод має значний науковий потенціал і може стати основою для подальших досліджень і розробок в галузі систем безпеки та формуванні оптимальних методів.

Висновки до розділу 2

В цьому розділі було розроблено метод комплексного оцінювання ефективності та оптимізація інтегрованої системи відеоспостереження, який поєднує декілька векторів ефективності: технічну, економічну, функціональну та їх загальну інтегральну. Для кожної з цих груп було визначено критерії оцінювання, які дозволяють виконати різносторонню оцінку різних конфігурацій.

Ключові результати розробки розділу:

1. Формування концепції ефективності інтегрованої системи. Було визначено та досліджено, що ефективність сучасної системи відеонагляду має оцінюватись комплексно з використанням декількох важливих аспектів які взаємопов'язані між собою. Визначили, що це критерії технічні, економічні та функціональні, які разом утворюють інтегральний комплексний індекс ефективності.

2. Визначення системи критеріїв оцінювання. Було розроблено систему критеріїв, для оцінки альтернативних пропозицій та конфігурацій на різних етапах проектування. Основними такими критеріями для технічного вектору є точність роботи, пропускна здатність мережі, обчислювальні потужності та надійність. Визначено критерії для економічного вектору, які складаються з вартості впровадження, витрат на експлуатацію та вартість інциденту. Функціональний вектор містить критерії покриття задач безпеки та задач інших департаментів, рівень автоматизації та зменшення навантаження на персонал.

3. Сформовано інтегральний критерій ефективності. Сформовано математичну модель, яка складається з технічних, економічних та функціональних критеріїв. Даний критерій ефективності дозволяє отримати оцінку загальної ефективності запропонованої конфігурації системи.

4. Розроблено метод оптимізації. Описали етапи оптимізації, які базуються на чіткій послідовності, яка починається зі збору вхідних даних, формування

альтернативних конфігурацій, розрахунок критеріїв для визначення найкращої конфігурації.

5. Формалізація вибору конфігурації. Встановлено правила за якими відбувається порівняння альтернативних конфігурацій, враховуючи технічні, економічні та функціональні вимоги та обмеження. Вибір оптимальної конфігурації здійснюється на основі обчислених критеріїв.

6. Наукова інтерпретація методу. Підтверджується, що запропонований метод є комплексним та універсальним, так якого принцип роботи не обмежений оцінкою конкретних параметрів, а використовує всі наявні параметри для визначення ефективності системи відеоспостереження.

В результаті розробленого методу доведено, що опираючись на його результати можна оцінювати ефективність системи та оптимізувати її опираючись на ряд факторів, а саме: технічні, економічні та функціональні. Такий підхід може бути використаний при наявності вхідних даних проекту так і за їх відсутності. В наступному розділі 3 буде здійснено математичну формалізацію даного методу, а також алгоритм для практичного застосування.

РОЗДІЛ 3

МАТЕМАТИЧНА МОДЕЛЬ ТА АЛГОРИТМІЧНА РЕАЛІЗАЦІЯ МЕТОДУ ОПТИМІЗАЦІЇ ІНТЕГРОВАНОЇ СИСТЕМИ ВІДЕОСПОСТЕРЕЖЕННЯ З ЕЛЕМЕНТАМИ АНАЛІЗУ ДАНИХ

В попередньому розділі було сформовано концептуальну основу методу комплексного оцінювання ефективності ІСВ та метод її оптимізація для знаходження найбільш доцільного значення для конкретного проекту. Для реалізації цього методу було також визначені групи критеріїв за якими буде відбуватись оцінка, підхід до формування значення ефективності кожної групи ефективності, їх спільний інтегральний показник та сформовано метод оптимізації. Для переходу від теоретичної концепції до рівня математичної моделі необхідно провести формалізацію параметрів, які ми розглянули, тобто задати їх у вигляді множин математичних змінних, визначити їх типи, області допустимих значень, характер впливу на цільову функцію, а також зафіксувати систему для зовнішніх обмежень.

Метою даного розділу є перехід від теоретичних методологічних рішень до математичної моделі, яку можна застосувати на практиці, яка в своїй основі буде містити розроблений алгоритм послідовності.

3.1. Формалізація параметрів системи

В рамках даної роботи будь-яка конфігурація інтегрованої системи відеоспостереження з елементами аналізу даних ми розглядаємо як багатопараметричний об'єкт, ефективність якого визначається сукупністю характеристик технологічних, економічних та функціональних критеріїв. Таким чином зазначаємо, що кожна розглянута конфігурація не може бути оцінена в повній мірі без врахування всіх показників, тоді як метод не буде мати закладеної ефективності.

Використовуючи математичний підхід ми визначаємо, що кожна обрана конфігурація системи має бути задана як елемент багатовимірного простору параметрів, у якому окремі координати відповідають за показники різної природи, але коли вони в межах одної моделі, то набувають формального узгодженого вигляду.

Нехай $S = \{s_1, s_2, \dots, s_n\}$, це множина можливих конфігурацій ІСВ, де s_i – одна окрема конфігурація, що включає в себе конкретне поєднання відеокамер, обчислювальних ресурсів серверів, пропускної здатності мережі, аналітичних інструментів, вартості впровадження та експлуатації, а також параметрів експлуатації. В такому випадку s_i не буде одним конкретним окремим числом або параметром, а це значення інтерпретується як структурована альтернатива, що далі буде описуватись за допомогою впорядкованої сукупності формальних змінних.

3.1.1. Загальна структура параметричного опису

Для того, аби в подальшому побувати математично кожену конфігурацію системи варто її представити у вигляді узагальненого математичного вектора параметрів, що включатиме в себе три підвектори: технічний, економічний, функціональний. Тоді узагальнений опис конфігурації s_i матиме наступний вигляд як на формулі (3.1).

$$X(s_i) = [X_{tech}(s_i), X_{econ}(s_i), X_{ops}(s_i)] , \quad (3.1)$$

Формування такого запису обумовлено не лише зручністю представлення та групуванням параметрів, а й важливим кроком для формального визначення простору конфігурацій. Далі така структура дозволяє будувати локальні критерії ефективності, задавати для критеріїв процедури нормування показників, визначати вагові коефіцієнти для кожного з параметрів та формувати інтегральний показник оцінювання конфігурації.

Водночас необхідно також врахувати, що всі описані параметри мають різну природу та одиниці виміру. Для окремих параметрів, які належать до метричних шкал,

тобто значення задається кількісно, можуть застосовуватись арифметичні операції. Такими параметрами вважаємо точність аналітики, затримка обробки, вартість системи, обчислювальне навантаження, пропускна здатність мережі. Інші параметри формуються через порядкові або квазіметричні шкали, тобто коли значення представлено у вигляді рівня прояву властивості і до них відносяться такі змінні як релевантність аналітики або ж рівень автоматизації об'єкту. Щоб поєднати такі параметри в математичній моделі та вивести інтегральний показник їх потрібно привести до числової форми, яка буде придатна для подальшого нормування.

3.1.2. Формалізація технічних параметрів

Задані технічні параметри в ІСВ дозволяють охарактеризувати знатність обраної конфігурації забезпечити необхідний рівень виявлення подій, швидкість обробки даних, відповідність до можливостей пропускної здатності мережі, а також стійкість до збоїв та загальна надійність роботи системи. З урахуванням прикладної реалізації обчислювального інструменту та моделі різних можливих топологій технічний вектор конфігурації буде виглядати як зображено на формулі (3.2).

$$X_{tech}(s_i) = [A_i, FP_i, FN_i, L_i, BW_i, \tau_i, BW_{max,i}^{top}, C_i, R_i, F_i], \quad (3.2)$$

де:

- A_i - точність аналітичного модуля для конфігурації s_i ;
- FP_i - частка хибних спрацювань;
- FN_i - частка пропусків подій;
- L_i - затримка обробки відеоданих;
- BW_i - фактично необхідна пропускна здатність каналів зв'язку;
- τ_i – тип архітектурної топології;

- $BW_{\max,i}^{top}$ - граничний мережевий ресурс, допустимий для обраного типу архітектурної організації системи;
- C_i - обчислювальне навантаження на платформу обробки;
- R_i - надійність функціонування конфігурації;
- F_i - відмовостійкість.

Варто відмітити включення параметру $BW_{\max,i}^{top}$, який є принципово важливим для узгодення представленої математичної моделі з практичним інструментом оцінки. На відмінну від спрощеного підходу межах якого будуть розглядатись топології, які відносяться до такого типу як: розподілена, централізована, гібридна та хмарна. Додатково буде враховано показник, який застосовується для даного методу, коли мережева складова оцінюється та описується лише через показник сумарного навантаження, то у такій моделі враховується і межа допустимого значення мережевого ресурсу яка залежить від обраної топології.

Тип архітектурної побудови буде завдаватись як номінальна дискретна змінна що відображена в залежності (3.3).

$$\tau_i \in \{\tau^{dist}, \tau^{cent}, \tau^{hyb}, \tau^{cloud}\}, \quad (3.3)$$

де:

- τ^{dist} - розподілена архітектура;
- τ^{cent} – централізована архітектура;
- τ^{hyb} – гібридна архітектура;
- τ^{hyb} - хмарна архітектура.

Виходячи з такого представлення, параметр мережевого ресурсу є функція типу архітектури (3.4).

$$BW_{\max,i}^{top} = g(\tau_i), \quad (3.4)$$

Даний опис (3.4) означає, що параметр максимально пропускної здатності мережі буде визначатись не абстрактно, а з урахуванням побудови конкретного типу архітектури. Такий підхід має особливе практичне значення, оскільки різні архітектури мають різний вплив на пропускну здатність а там самим і на інші параметри, економічного типу та функціонального.

Домени значень таких параметрів можна задати так (3.5):

$$\begin{aligned} A_i &\in [0,1], & FP_i &\in [0,1], & FN_i &\in [0,1], \\ L_i &\in R_+, & BW_i &\in R_+, & BW_{max,i}^{top} &\in R_+, \\ C_i &\in R_+, & R_i &\in [0,1], & F_i &\in [0,1]. \end{aligned} \quad (3.5)$$

Оцінюючі в подальшому технічні параметри, які мають різний напрям оптимізації варто визначити такі напрями для усунення хибних трактувань. До параметрів, де чим більше значення тим кращий показник, відносяться: R_i , A_i , F_i .

Щодо параметрів FP_i , FN_i , L_i , BW_i , C_i , то ці значення цих параметрів зворотно пропорційне – якщо значення показника менше, тоді даний параметр краще. Параметр τ_i має структурний, а не оціночний зміст, тоді як $BW_{max,i}^{top}$ використовується як гранична умова допустимості, але не як прямий критерій якості. Повний перелік формалізованих параметрів представлений в таблиці 3.1.

Таблиця 3.1.

Формалізовані технічний вектор системи

Позначення параметра	Назва параметра	Тип шкали	Область значень	Одиниця вимірювання	Напрямок оптимізації

A_i	Точність аналітики	Метрична	[0;1]	частка або %	max
FP_i	Частка хибних спрацьовувань	Метрична	[0;1]	частка або %	min
FN_i	Частка пропусків подій	Метрична	[0;1]	частка або %	min
L_i	Затримка обробки	Метрична	R_+	мс, с	min
BW_i	Необхідна пропускна здатність	Метрична	R_+	Мбіт/с	min
τ_i	Тип архітектурної побудови системи	Номінальна, дискретна	$\tau^{dist}, \tau^{cent}, \tau^{hyb}, \tau^{cloud}$	—	—
$BW_{max,i}^{top}$	Допустима пропускна здатність для архітектури	Метрична	R_+	Мбіт/с	max*
C_i	Обчислювальне навантаження	Метрична	R_+	%, GFLOPS, ум. од.	min
R_i	Надійність	Метрична	[0;1]	частка або %	max

F_i	Відмовостійкість	Метрична	[0;1]	частка або %	max
-------	------------------	----------	-------	--------------	-----

3.1.3. Формалізація економічних параметрів

Опис економічної конфігурації є необхідною складовою даного методу та математичної модулі, оскільки раціональний вибір конфігурації також напряму залежить від вартості її впровадження, підтримки в процесі експлуатації та вартості наслідків інцидентів. В рамках даної роботи економічний вектор виглядає наступним чином (3.6).

$$X_{econ}(s_i) = [C_{0i}, C_{opi}, C_{sc,i}, C_{ch,i}, C_{det,i}, L_{inc,i}, P_i], \quad (3.6)$$

де:

- C_{0i} - початкові витрати на розгортання конфігурації;
- C_{opi} – експлуатаційні витрати;
- $C_{sc,i}$ – вартість масштабування системи;
- $C_{ch,i}$ – вартість одного каналу відеоспостереження;
- $C_{det,i}$ – вартість одної корисної детекції;
- $L_{inc,i}$ – математичне очікування витрат від пропуску інциденту;
- P_i – умовний термін окупності конфігурації.

Усі економічні параметри та оцінки в межах описаної математичної модулі мають бути задані в гривневій валюті для проведення необхідних точних розрахунків. Такий підхід забезпечує єдиний вигляд представлення вартості і таким чином узгоджується з прикладним інструментом оцінювання, у межах якого всі розрахунки відбуваються у національній валюті.

Особливу увагу варто звернути на параметр $L_{inc,i}$, який відображає математичне очікування збитку від пропуску інциденту. Його застосування в даній математичній моделі не обмежується лише прямими витратами на впровадження та обслуговування системи ІСВ, а враховує додатково також очікувані витрати, які можуть бути згенеровані рівнем недостатньої ефективності аналітики, яка запропонована в рамках конфігурації системи. Саме такий параметр дозволяє встановити лінійний зв'язок і пов'язати економічну складову з технічною і перейти від спрощеної оцінки вартості до більш складної, але наближеної до реальних умов.

Домени економічних параметрів задаються як (3.7):

$$C_{0i}, C_{opi}, C_{sc,i}, C_{ch,i}, C_{det,i}, L_{inc,i}, P_i \in R_+, \quad (3.7)$$

Усі економічні параметри належать до показнику типу де менше значення означає кращу ефективність параметру та представлені в таблиці 3.2.

Таблиця 3.2.

Формалізований економічний вектор конфігурації системи

Позначення параметра	Назва параметра	Тип шкали	Область значень	Од. вимірювання	Напрямок оптимізації
C_{0i}	Початкові витрати на розгортання	Метрична	R_+	грн	min
C_{opi}	Експлуатаційні витрати	Метрична	R_+	грн/рік	min
$C_{sc,i}$	Вартість масштабування	Метрична	R_+	грн	min

$C_{ch,i}$	Вартість одного каналу	Метрична	R_+	грн/канал	min
$C_{det,i}$	Вартість однієї корисної детекції	Метрична	R_+	грн/детекцію	min
$L_{inc,i}$	Математичне очікування збитку від пропуску інциденту	Метрична	R_+	грн	min
$L_{inc,i}$	Умовний строк окупності	Метрична	R_+	місяці або роки	min

3.1.4. Формалізація функціональних параметрів

Функціональні параметри відображають прикладну результативність роботи конфігурації, тобто наскільки запропонована система здатна вирішувати поставлені перед нею задачі безпеки та відеонагляду, підвищувати ефективність роботи оператора і виконувати автоматизацію бізнес-процесів. Такі параметри поєднують технічні функціональні можливості системи з її експлуатаційною корисністю. У межах даної роботи вектор доцільно задавати в наступному вигляді (3.8):

$$X_{ops}(s_i) = [S_i, Q_i, Q_{fat,i}, T_i, U_i], \quad (3.8)$$

де:

- S_i – рівень покриття задач безпеки;
- Q_i – релевантність аналітики під задачі конкретного об'єкту;
- $Q_{fat,i}$ – фактор втоми оператора;
- T_i – оперативність реагування на інцидент;

- U_i – рівень автоматизації.

В даній математичній моделі було замінено узагальнений показник втоми та навантаження на оператора на більш конкретний $Q_{fat,i}$, який є методично доцільним, оскільки дозволяє пов'язати фактори якості роботи аналітики та її точність, яка генерує відповідну кількість подій на якість роботи оператора. У прикладній формулі такий параметр залежить від кількості подій, які оброблює оператор, а також кількість подій які мають хибний характер через низьку точність аналітики або складність сцени. У загальному математичному вигляді ця залежність виглядає наступним чином (3.9):

$$Q_{fat,i} = f(N_{ev,i}, FP_i), \quad (3.9)$$

де $N_{ev,i}$ – параметр який містить кількість подій, що підлягають обробці оператором в межах заданої конфігурації s_i .

Для спрощеного лінійного опису така залежність може бути подана наступною формулою (3.10):

$$Q_{fat,i} = q_0 + k_1 N_{ev,i} + k_2 FP_i, \quad (3.10)$$

де:

- q_0 – базовий рівень навантаження на оператора;
- k_1 – коефіцієнт впливу на інтенсивність подій;
- k_2 – коефіцієнт впливу хибних спрацювань.

Уведення параметру $Q_{fat,i}$, є важливим зв'язком між реальною експлуатацією конфігурації системи та її математичною моделлю, так як вона дозволяє враховувати деградацію фокусу оператора при великій кількості подій та опрацювань хибних детекцій системи. Це створює основу для подальшого підрозділу, в якому буде

розглянуто вплив помилок роботи аналітичних інструментів на загальну ефективність системи.

Для функціональних параметрів буде представлений наступний нормований домен (3.11):

$$S_i, Q_i, Q_{fat,i}, T_i, U_i \in [0,1] , \quad (3.11)$$

Потрібно також зазначити, що необхідне проведення до цього інтервалу після первинного обчислення у заданих уніфікованих одиницях виміру (часових, бальних або відсотках).

Таблиця. 3.3.

Формалізований функціональний вектор конфігурації системи

Параметр	Назва параметра	Тип шкали	Область значень	Одиниця вимірювання	Напрямок оптимізації
S_i	Покриття задач безпеки	Метрична / квазіметрична	$([0;1])$	частка або %	max
Q_i	Релевантність аналітики задачам об'єкта	Порядкова / квазіметрична	$([0;1])$	безрозмірна величина	max
$Q_{fat,i}$	Фактор втоми оператора	Метрична / розрахункова	$([0;1])$ або R_+	безрозмірна величина	min
T_i	Оперативність реагування	Метрична / квазіметрична	$([0;1])$	безрозмірна величина	max
U_i	Рівень автоматизації	Метрична / квазіметрична	$([0;1])$	частка або %	max

3.1.5. Узагальнений параметричний простір конфігурацій

Після того, як було виведено три основні груп параметрів кожна конфігурація системи, для якої проводиться оцінка, може бути інтерпретована як точка в узагальненому багатовимірному просторі. Якщо m, p, q – кількість технічних, економічних та функціональних параметрів відповідно, тоді (3.12):

$$s_i \in R^m \times R^p \times R^q, \quad (3.12)$$

В більш широкому трактуванні це пояснюється тим, що у математичному просторі де є змішані метричні, порядкові та дискретні параметри. Після формального задання та числового кодування ці значення підлягають єдиній процедурі оцінювання. Вказане представлення дозволяє перевести задачу оцінювання ефективності конфігурації системи виходячи не з вербального або суб'єктивної оцінки, а з використанням математичного аналізу. Тобто, кожна альтернативна має значення (координати у просторі параметрів), а параметр оптимізації розглядається як задача для пошуку найкращої точки в цьому просторі за використанням заданих обмежень та правила інтегрального оцінювання ефективності.

3.1.6. Зовнішні обмеження та критерії допустимості

Формально, якщо задати великий масив можливих множин конфігурацій систем це не буде означати, що всі доступні такі конфігурації будуть розглянуті, як придатні до впровадження. Опираючись на реальні чинники, що будь яка система, а особливо система безпеки, будується з врахуванням зовнішніх обмежень, то вони і задають область допустимих значень. У межах даної роботи присутні наступні зовнішні обмеження:

- Бюджетні;

- Аналітичні;
- Мережеві;
- Обчислювальні.

Бюджетне обмеження представлене в наступному вигляді (3.13):

$$B(s_i) \leq B_{max} , \quad (3.13)$$

де:

$B(s_i)$ - повна вартість конфігурації,

B_{max} – максимально допустимий бюджет.

Обмеження за точністю аналітики представлене в наступному вигляді (3.14):

$$A_i \geq A_{min} , \quad (3.14)$$

де:

A_{min} - мінімальний допустимий рівень точності;

A_i – розрахований рівень точності поточної конфігурації.

Мережева допустимість конфігурації повинна враховувати не лише доступний ресурс, а й тип архітектури побудованої конфігурації, тому базова умова виглядає наступним чином (3.15):

$$BW_i \leq BW_{max,i}^{top} , \quad (3.15)$$

Обмеження за обчислювальними ресурсами подано в наступному вигляді (3.16):

$$C_i \leq C_{max} , \quad (3.16)$$

де C_{max} – максимальний допустимий обчислювальний ресурс платформи.

На основі виведених обмежень формуємо множину допустимих конфігурацій, які мають наступний вигляд (3.17):

$$\Omega = \{s_i \in S \mid B(s_i) \leq B_{max}, A_i \geq A_{min}, BW_i \leq BW_{max,i}^{top}, C_i \leq C_{max}\}, \quad (3.17)$$

Тобто, сама множина Ω , а не початкова множина S , є предметом для використання математичного оцінювання та виконання оптимізації. Такий підхід відразу виключає з аналізу такі альтернативи, які хоч і мають теоретичну ймовірність на існування, але не мають практичної реалізації.

3.1.7. Значення виконаної формалізації для подальшого дослідження

В даному підрозділі було виконано систематизацію характеристик системи, а також переведення цих характеристик в єдиний формальний простір, який є придатним для аналітичної обробки. Якщо в попередньому розділі увага була приділена змістовному визначенню критеріїв, то в даній частині всі описані та визначені критерії набули статусу математичних змінних з визначеними доменами, типами шкал, напрямками оптимізації та зовнішніми обмеженнями.

Особливо варто виокремити узгоджені моделі з прикладним обчислювальним інструментом, що використовується в межах даного дослідження. Таким чином до параметрального опису включено:

- $BW_{max,i}^{top}$ – як характеристику мережевого ресурсу, який є залежним від типу обраної архітектури конфігурації системи;
- $L_{inc,i}$ – як оцінку математичного очікування фінансових збитків від пропуску інциденту аналітичними інструментами;
- $Q_{fat,i}$ – як формалізований показник втоми оператора.

Саме включення та опрацювання таких параметрів дозволяє математичну модель пов'язати з практичною системою розрахунків, яка використовується для моделювання топології, витрат на ліцензію ПЗ, навантаження на оператора та вибір раціональної конфігурації.

3.2. Математична модель інтегральної ефективності

3.2.1. Загальна постановка моделі інтегральної ефективності

Після формалізації параметричного опису конфігурації інтегрованої системи відеоспостереження наступним кроком є побудова математичної моделі інтегральної ефективності.

Визначаємо, що для кожної конфігурації $s_i \in \Omega$, де Ω – це множина допустимих рішень. Задаємо узагальнений вектор параметрів, який має наступний вигляд (3.18):

$$X(s_i) = [X_{tech}(s_i), X_{econ}(s_i), X_{ops}(s_i)] , \quad (3.18)$$

В такому випадку інтегральна ефективність конфігурації s_i , буде визначена як функція від оцінок визначених технічними, економічними, функціональними параметрами:

$$E_{total}(s_i) = \Phi(E_{tech}(s_i), E_{econ}(s_i), E_{ops}(s_i)) , \quad (3.19)$$

де:

- $E_{tech}(s_i)$ – локальна технічна ефективність конфігурації;
- $E_{econ}(s_i)$ - локальна економічна ефективність конфігурації;
- $E_{ops}(s_i)$ - локальна функціональна ефективність конфігурації;

- Φ – оператор агрегування.

Формується задача побудови математичної моделі, яка зводиться до наступної конкретизації:

1. Визначення способу обчислення локальних ефективностей;
2. Формування правил нормування параметрів;
3. Формування правил зважування критеріїв;
4. Формування виду оператора агрегування.

Такий підхід має забезпечити та відповідати основним вимогам, як порівнювальність різних параметрів, стійкість до змін одиниць виміру та інтерпритованість результату. В результаті обчислень значення інтегрального показника, що буде отримане з використанням нормованих узгоджених величин, дозволить отримати значення для оцінки та порівняння.

3.2.2. Локальні функції ефективності

Для кожної групи параметрів формується окрема функція локальної ефективності, адже таким чином зберігається структура логіки та її послідовність. Технічна локальна ефективність будується на основі технічного вектору (3.20):

$$X_{tech}(s_i) = [A_i, FP_i, FN_i, L_i, BW_i, \tau_i, BW_{max,i}^{top}, C_i, R_i, F_i], \quad (3.20)$$

де:

- $\widehat{x_{ik}^{(tech)}}$ - нормоване значення k -го технічного рішення параметру конфігурації s_i ;
- α_k – ваговий коефіцієнт технічного параметру;
- m – кількість технічних критеріїв, що входять до оцінки;
- $\sum_{k=1}^m \alpha_k = 1$.

В такому підході τ_i займає роль структурного класифікатора, а параметр $BW_{\max,i}^{top}$ - відіграє роль технічної межі допустимості, яка визначає, чи може конфігурація, яка передана на аналіз, бути реалізована в межах обраної архітектури. Інакше кажучи, вказані параметри впливають на модель не через пряме збільшення технічної оцінки, а через формування коректного контексту для інших змінних в математичній моделі.

Для економічної ефективності формується на основі (3.21) вектору описаного нижче.

$$X_{econ}(s_i) = [C_{0i}, C_{opi}, C_{sc,i}, C_{ch,i}, C_{det,i}, L_{inc,i}, P_i], \quad (3.21)$$

Функція економічної ефективності може бути подана як (3.22):

$$E_{econ}(s_i) = \sum_{k=1}^p \beta_k \widehat{x_{ik}^{(econ)}}, \quad (3.22)$$

де:

- $x_{ik}^{(econ)}$ – нормоване значення k -го економічного параметру;
- β_k – ваговий коефіцієнт відповідного економічного параметру;
- p – кількість економічних критеріїв;
- $\sum_{k=1}^p \beta_k = 1$.

Важливий параметр для цієї частини є $L_{inc,i}$, що дозволить в даному підході врахувати не лише прямі витрати на систему, а й очікувані витрати від її експлуатації опираючи на технічні параметри ефективності.

Для функціональної ефективності буде наступне представлення, що визначається на основі вектору(3.23):

$$X_{ops}(s_i) = [S_i, Q_i, Q_{fat,i}, T_i, U_i], \quad (3.23)$$

Тоді локальна функція функціональної ефективності, в рамках розрахунку, матиме наступний вигляд (3.24):

$$E_{ops}(s_i) = \sum_{k=1}^q \gamma_k \widehat{x_{ik}^{(ops)}}, \quad (3.24)$$

де:

- $\widehat{x_{ik}^{(ops)}}$ – нормоване значення k -го функціонального параметру;
- γ_k – вага відповідного функціонального показника;
- q – кількість функціональних параметрів;
- $\sum_{k=1}^q \gamma_k = 1$.

Важливе значення на розрахунок ефективності має параметр $Q_{fat,i}$, який формує вплив якості та точності аналітики на людський фактор. Даний параметр враховує деградацію якості роботи персоналу через кількість потенційних інцидентів.

3.2.3. Нормування параметрів

Оскільки параметри системи мають різні одиниці виміру, перед виконанням агрегування потрібно виконати нормування таких значень. Без застосування цього методу інтегральна модель була б методично некоректна.

Доцільно використовувати процедуру нормування за схемою «min-max», яке приводить всі параметри до вигляду значень інтервалу $[0;1]$. Для параметрів, які мають трактування чим більше значення – тим краще використовуємо наступний підхід (3.25):

$$\widehat{x_{ij}} = \frac{x_{ij} - x_j^{min}}{x_j^{max} - x_j^{min}}, \quad (3.25)$$

Для параметрів зворотної оцінки, де менше значення означає кращий результат, використовуємо (3.26):

$$\widehat{x}_{ij} = \frac{x_j^{max} - x_{ij}}{x_j^{max} - x_j^{min}}, \quad (3.26)$$

де:

- x_{ij} – значення j -го параметру для конфігурації s_i ;
- x_j^{min} – мінімальне значення заданого параметру;
- x_j^{max} – максимальне значення цього параметру.

Таким чином, після нормалізації величин отримуємо єдину інтерпретацію $\widehat{x}_{ij}$, яка визначає оцінку по принципу чим більше – тим краще. Щодо інших параметрів, які не мають метричної шкали, то для них застосовується попереднє кодування або інтерпретація значень, а після - нормалізація.

3.2.4. Система вагових коефіцієнтів

Після виконання процесу нормування, наступним кроком, для реалізації методу оцінки, відбувається визначення вагових коефіцієнтів. Такий підхід необхідний, аби визначити які параметри мають більші вимоги щодо якості роботи або ж ефективності.

У межах локальних функцій, вагові коефіцієнти визначають відносну критичність для кожної групи (3.27):

$$\sum_{k=1}^m \alpha_k = 1, \quad \sum_{k=1}^p \beta_k = 1, \quad \sum_{k=1}^q \gamma_k = 1, \quad (3.27)$$

На рівні згортки локальних ефективностей вводяться ваги групових критеріїв, які представлені в наступному рівнянні (3.28):

$$w_{tech} + w_{econ} + w_{ops} = 1 , \quad (3.28)$$

де:

- w_{tech} - вага технічної складової;
- w_{econ} – вага економічної складової;
- w_{ops} – вага функціональної складової.

Таким чином загальна структура коефіцієнтів вагів представлена для окремих параметрів в середині кожної з груп та для груп і інтегральному методі.

3.2.5. Агрегування локальних критеріїв

Основною моделлю інтегральної ефективності в межах даного дослідження вважаємо адитивну модель згортки, яка представлена далі (3.29):

$$E_{total}(s_i) = w_{tech}E_{tech}(s_i) + w_{econ}E_{econ}(s_i) + w_{ops}E_{ops}(s_i) , \quad (3.29)$$

Перевагою використання такого підходу є:

- Забезпечує інтерпретованість;
- Чітко відстежує частку кожної складової;
- Може бути представлена в табличному вигляді та для програмної реалізації;
- Придатний для ранжування альтернатив з інтегральною оцінкою.

У випадках, коли така модель неможлива для використання існує можливість використання мультиплікативної форми, яка до того ж дозволяє посилити ефект від вагових коефіцієнтів (3.30):

$$E_{total}^{(mult)}(s_i) = E_{tech}(s_i)^{w_{tech}} \cdot E_{econ}(s_i)^{w_{econ}} \cdot E_{ops}(s_i)^{w_{ops}} , \quad (3.30)$$

Така форма є більш чутливою до низьких значень локальних ефективностей, проте, як для базової моделі, доцільно використовувати саме адитивну згортку.

3.2.6. Багатокритеріальна форма моделі

З формальної точки зору інтегральна ефективність може бути представлена у вигляді узагальненої багатокритеріальної моделі (3.31):

$$E_{total}(s_i) = \sum_{j=1}^r w_j \widehat{x}_{ij}, \quad (3.31)$$

де:

- r – загальна кількість критеріїв у моделі;
- w_j – вага j -го критерію;
- $\widehat{x}_{ij}$ – нормоване значення для j -го критерію для конфігурації s_i .

Такий запис значно компактніше, але водночас менш інтерпретований з погляду прикладного аналізу, саме тому доцільно використовувати раніше визначену дворівневу структуру з локальними функціями ефективності, а поточний запис розглядати як загальну математичну форму моделі.

3.2.7. Обмеження та штрафна форма інтегральної моделі

Незважаючи на те, що першочергово відбувається оцінка лише допустимих значень та варто з погляду математичної завершеності ввести також штрафну форму інтегральної моделі. Цей підхід дозволяє оцінювати конфігурацію навіть до процесу фільтрації, що тим самим знижує інтегральний результат у разі порушення такого обмеження.

Штрафна форма подана в наступному вигляді (3.32):

$$E_{total}^*(s_i) = E_{total}(s_i) - \Pi(s_i), \quad (3.32)$$

де $\Pi(s_i)$ – штрафна функція, що враховує порушення обмеження.

У межах практичної реалізації методу, така форма корисна на етапі попереднього модулювання множини альтернативних конфігурацій, але для фінального ранжування доцільніше з наукової точки зору використовувати лише множину з допустимих рішень Ω .

3.2.8. Інтерпретація інтегрального показника

Визначене значення інтегрального критерію ефективності $E_{total}(s_i)$ варто розуміти як узагальнена оцінка, яка має безрозмірну форму для конфігурації s_i , що враховує всі три критерії ефективності системи. Чим більше ми отримуємо дане значення, тим вища ефективність розглянутої конфігурації.

Важливо зазначити, що даний показник не підміняє собою окремі критерії, а узагальнює їх. Для більш детального аналізу можна використовувати індивідуальні показники ефективності для критеріїв, які відносяться для технічної, економічної та функціональної ефективностей.

3.3. Математичне врахування впливу модулів аналітики на інтегральну ефективність

У попередньому підрозділі було сформовано математична моделі інтегральної ефективності для інтегрованої системи відеоспостереження, яка складалась з трьох груп критеріїв. В рамках даного дослідження, де розглядаються системи з елементами аналізу даних, таких показників недостатньо, оскільки аналітичні інструменти не є нейтральними показниками, а це означає що результати його роботи прямо впливають на інші показники та на інтегральну ефективність вцілому. Тому для таких систем

вводимо додатковий параметри – похибку прийняття рішення, який зумовлений роботою аналітичного модуля.

Завданням даного підрозділу полягає у формалізації впливу аналітичних інструментів на підсистеми оцінювання. Визначено показники, на які може впливати аналітика, а саме:

- Підвищення або зниження ефективності технічної конфігурації;
- Збільшення або зменшення навантаження на оператора;
- Зменшення загального показника функціональної ефективності;
- Витрати як наслідок для кількості виявлених або невиявлених інцидентів.

3.3.1. Базові показники якості роботи аналітичних інструментів

Розглядаємо конфігурацію s_i , в якій працюють аналітичні детектори і у межах обраного спостереження формує такі базові результати класифікації:

- TP_i - кількість істинно позитивних рішень;
- TN_i - кількість істинно негативних рішень;
- FP_i - кількість хибно позитивних рішень;
- FN_i - кількість хибно негативних рішень.

В такому випадку основі показники, які визначають якість роботи аналітики будуть відображені стандартним співвідношенням представленого нижче (3.33):

$$A_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}, \quad (3.33)$$

де A_i - загальна точність аналітики.

Показник точності, в рамках такого обмеженого вигляду, є недостатнім, оскільки таких підхід не дозволяє окремо оцінити вплив хибних спрацювань і пропусків. Для

реалізації такого розрахунку використовуємо додаткові показники з формули (3.34), (3.35), (3.36):

$$Pr_i = \frac{TP_i}{TP_i + FP_i}, \quad (3.34)$$

де Pr_i – precision, тобто точність позитивного спрацювання.

$$Re_i = \frac{TP_i}{TP_i + FN_i}, \quad (3.35)$$

де Re_i – recall, тобто повнота виявлення цільових подій,

$$F1_i = \frac{2Pr_iRe_i}{Pr_i + Re_i}, \quad (3.35)$$

де $F1_i$ – гармонійне середнє між показниками precision та recall.

В рамках побудованої інтегральної математичної моделі A_i , яка вже входить в розрахунки технічного вектору конфігурації, але, як було визначено, такий показник ми розглядаємо не лише як первинний критерій якості роботи аналітики, а і як джерело вторинного впливу на інші складові інших критеріїв ефективності. Тобто хибні або правильно спрацювання будуть не лише впливати локально на технічну ефективність, а і впливають на економічну та функціональну.

3.3.2. Стохастичний характер помилок аналітики

Якість роботи аналітики, в реальних експлуатаційних умовах, мають значний фактор випадковості, оскільки на точність аналітики впливає складність сцени, щільність подій, умови освітлення, ракурс встановленої відеокамери, рівень завад і якість навчальної вибірки. Саме тому існують мінімальні вимоги від виробників

програмного забезпечення, для експлуатації, щоб зазначений рівень точності був досягнутий. Тому помилки аналітики розглядаємо як стохастичні величини.

Нехай:

- $\lambda_{inc,i}$ – інтенсивність реальних інцидентів для конфігурації s_i ;
- Δt – часовий інтервал аналізу;
- $P_{FP,i}$ – ймовірність хибного позитивного спрацювання;
- $P_{FN,i}$ – ймовірність пропуску інциденту.

Тоді математичне очікування кількості пропущених інцидентів матиме наступний вигляд (3.36):

$$\overline{N_{FN,i}} = \lambda_{inc,i} \cdot P_{FN,i} \cdot \Delta t , \quad (3.36)$$

Для правильно виявлених подій математичне очікування буде представлено наступним виглядом (3.37):

$$\overline{N_{TP,i}} = \lambda_{inc,i} \cdot (1 - P_{FN,i}) \cdot \Delta t , \quad (3.37)$$

Для $\lambda_{bg,i}$ – інтенсивність об'єктів або подій, які формують хибні тривоги. Для її розрахунку сформоване наступне математичне очікування кількості хибних тривожних подій (3.38):

$$\overline{N_{FP,i}} = \lambda_{bg,i} \cdot P_{FP,i} \cdot \Delta t , \quad (3.38)$$

Такі співвідношення дозволяють провести якісну оцінку, оскільки не формується твердження помилки аналітики, а розраховується його кількісний показник.

3.3.3. Вплив помилок аналітики на технічну ефективність

Для визначення технічної ефективності модуль формує окрему підсистему якості подій, а саме тому варто виділити окрему локальну оцінку ефективності (3.39). Нехай нормовані значення показників точності аналітики для конфігурації s_i означаються як:

$$\widehat{A}_i, \widehat{Pr}_i, \widehat{Re}_i, \widehat{F1}_i, \quad (3.39)$$

В такому випадку локальна технічна якість аналітичного модулю подається в такому вигляді:

$$E_{AI}^{tech}(s_i) = \rho_1 \widehat{A}_i + \rho_2 \widehat{Pr}_i + \rho_3 \widehat{Re}_i + \rho_4 \widehat{F1}_i, \quad (3.40)$$

де параметри $\rho_1, \rho_2, \rho_3, \rho_4$ – вагові коефіцієнти показників якості аналітики, які відповідають наступному правилу (3.41), що:

$$\rho_1 + \rho_2 + \rho_3 + \rho_4 = 1, \quad (3.41)$$

Враховуючи такі параметри базова технічна ефективність (3.42) може бути уточнена наступним виглядом:

$$E_{tech}^*(s_i) = \alpha_0 E_{base-tech}(s_i) + \alpha_{AI} E_{AI}^{tech}(s_i), \quad (3.42)$$

де:

- $E_{base-tech}(s_i)$ – технічна ефективність без спеціального виокремлення AI-компоненту;
- $E_{AI}^{tech}(s_i)$ – локальна оцінка технічної якості модулю;

- α_0, α_{AI} – коефіцієнти внеску базової технічного та аналітичної складової для яких діє правило $\alpha_0 + \alpha_{AI} = 1$.

Цей запис дозволяє уникнути спрощення, яке негативно впливає на показники аналітики та можуть бути неявно показані через точку зору впливу на технічно ефективність.

3.3.4. Вплив хибних спрацювань на навантаження оператора

Важливим аспектом необхідно враховувати вплив вторинних ефектів помилок аналітики, яке прямо впливає на ефективність роботи оператора. Чим більше утворюється хибних спрацювань – тим більше оператору доводиться оброблювати такі інциденти, що впливає на його фокус та ефективність.

Нехай:

- $\overline{N_{ev,l}}$ – загальна кількість подій;
- $\overline{N_{TP,l}}$ – математичне очікування коректних подій;
- $\overline{N_{FP,l}}$ – математичне очікування хибних подій.

Тоді маємо наступний вплив: $\overline{N_{ev,l}} = \overline{N_{TP,l}} + \overline{N_{FP,l}}$.

Фактор втоми оператора для конфігурації s_i , який виведений наступним параметром $Q_{fat,i}$, доцільно конкретизувати через лінійну модель (3.43):

$$Q_{fat,i} = q_0 + k_1 \overline{N_{ev,l}} + k_2 \overline{N_{FP,l}}, \quad (3.43)$$

де:

- q_0 – базовий коефіцієнт навантаження;
- k_1 - коефіцієнт впливу загального потоку подій;
- k_2 - коефіцієнт додаткового впливу хибних спрацювань.

За необхідності даний показник проходить процес нормалізації до величин $[0;1]$.

3.3.5. Вплив пропусків інциденту на функціональну та економічну ефективність

Визначено, що кількість хибних тривожних інцидентів впливає на функціональну ефективність системи, але пропуски інцидентів, через якість роботи аналітики, впливають саме на економічну ефективність та додаткові витрати.

Нехай $D_{inc,i}$ - середній збиток від пропущеного інциденту в конфігурації s_i . Тоді математичне очікування сумарного збитку за інтервал часу Δt подається наступним чином (3.44):

$$L_{inc,i} = D_{inc,i} \cdot \overline{N_{FN,i}}, \quad (3.44)$$

Якщо підставити вираз для $\overline{N_{FN,i}}$, то формула (3.44) матиме наступне представлення в (3.45):

$$L_{inc,i} = D_{inc,i} \cdot \lambda_{inc,i} \cdot P_{FN,i} \cdot \Delta t, \quad (3.45)$$

Цей параметр вже був включений до економічного вектору конфігурації, але тут він набуває функціонального змісту, що витрати зростають пропорційно кількості пропущених інцидентів.

Якщо оцінювати такі пропуски для параметру корисності системи, то вони також мають вплив і формалізовано, через скоригований показник функціонального покриття, це може виглядати так:

$$S_i^* = S_i (1 - \mu_1 P_{FN,i}), \quad (3.46)$$

де:

- S_i – базовий рівень покриття задач безпеки;
- μ_1 – коефіцієнт чутливості функціонального покриття до пропуску подій.

Скоригована релевантність аналітики має наступний вигляд як на формулі (3.47):

$$Q_i^* = Q_i - \mu_2 P_{FN,i} - \mu_3 P_{FP,i} , \quad (3.47)$$

де μ_2, μ_3 – коефіцієнти впливу пропусків і хибних спрацювань на функціональні вимоги якості аналітики на об'єкті.

3.3.6. Скориговані локальні функції ефективності

Необхідно врахувати наведені в попередніх підрозділах залежності для локальних функцій ефективності щоб вони містили явний вплив точності роботи аналітичних інструментів конфігурації.

Скоригована технічна ефективність (3.48):

$$E_{tech}^*(s_i) = \alpha_0 E_{base-tech}(s_i) + \alpha_{AI} E_{AI}^{tech}(s_i) , \quad (3.48)$$

Скоригована економічна ефективність (3.49):

$$E_{econ}^*(s_i) = \beta_0 E_{base-econ}(s_i) - \beta_1 \widehat{L_{inc,l}} - \beta_2 \widehat{C_{fp,l}} , \quad (3.49)$$

де:

- $\widehat{L_{inc,l}}$ – Нормовані збитки від пропусків інцидентів;
- $\widehat{C_{fp,l}}$ – Нормована вартість оброки хибних тривог;
- $\beta_0, \beta_1, \beta_2$ - вагові коефіцієнти.

Якщо $c_{op,i}$ – середня вартість обробки одної хибної тривоги оператором, тоді

$$C_{fp,i} = c_{op,i} \cdot \overline{N_{FP,i}}.$$

Скоригована функціональна ефективність (3.50):

$$E_{ops}^*(s_i) = \gamma_1 \widehat{S}_i^* + \gamma_2 \widehat{Q}_i^* + \gamma_3 (1 - \widehat{Q_{fat,i}}) + \gamma_4 \widehat{T}_i + \gamma_5 \widehat{U}_i, \quad (3.50)$$

де :

- $\widehat{S}_i^*$ - нормоване скориговане покриття задач;
- $\widehat{Q}_i^*$ - нормована скоригована релевантність аналітики;
- $\widehat{Q_{fat,i}}$ - нормований фактор втоми оператора;
- $\widehat{T}_i, \widehat{U}_i$ – нормовані показники оперативності реагування та автоматизації;
- $\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 + \gamma_5 = 1$.

Запис $(1 - \widehat{Q_{fat,i}})$ використовується тому, що втома оператора це деструктивний ефект на його ефективність і чим цей фактор впливу більший тим нижча функціональна ефективність системи.

3.3.7. Скоригована інтегральна ефективність з урахуванням аналітики

З урахуванням всіх скоригованих локальних функцій інтегральна ефективність системи з включеним в себе впливом аналітики буде мати наступний вигляд (3.51):

$$E_{total}^{AI}(s_i) = w_{tech} E_{tech}^*(s_i) + w_{econ} E_{econ}^*(s_i) + w_{ops} E_{ops}^*(s_i), \quad (3.51)$$

Ця формула є ключовим результатом даного підрозділу, оскільки розглядає критерій аналітики не як один з технічних параметрів впливу, а як повноцінне математичне врахування аналітик на всю систему. Тобто, даний підхід показує вплив

критерія аналітики не обмежується локальною областю застосування, а навпаки – вона впливає на всі інші параметри ефективності.

3.3.8. Аналіз чутливості інтегральної ефективності до якості аналітики

Щоб розрахувати оцінювання впливу змін у якості роботи аналітичного детектору в рамках конфігурації та його впливу на загальну ефективність системи необхідно додати показник чутливості інтегральної моделі до точності аналітики та її помилок. Для параметру точності представлені такі умови (3.52):

$$\frac{\partial E_{total}^{AI}}{\partial A_i} > 0, \quad (3.52)$$

Для параметру хибних спрацювань такі умови (3.53):

$$\frac{\partial E_{total}^{AI}}{\partial P_{FP,i}} < 0, \quad (3.53)$$

Для параметру пропусків інцидентів такі умови (3.54):

$$\frac{\partial E_{total}^{AI}}{\partial P_{FN,i}} < 0, \quad (3.54)$$

Що стосується практичних розрахунків, коли точність аналітики збільшує інтегральний показник ефективності, зручніше використання скінченних приростів, яке має наступне представлення (3.55):

$$\Delta E_{total}^{(A)} = E_{total}^{AI}(A_i + \Delta A) - E_{total}^{AI}(A_i), \quad (3.55)$$

3.4. Побудова алгоритму вибору раціональної конфігурації

Етап побудови алгоритму процедури вибору раціональної конфігурації переводить залежності визначені в попередніх розділах в послідовність формальних обчислювальних операцій, які можна буде застосувати для програмної реалізації методу розрахунку ефективності та оптимізації.

Метою даного алгоритму є вибір конфігурації s_{opt} , яка буде найбільш оптимальна з урахуванням заданих обмежень.

3.4.1. Формальна постановка задачі вибору

Нехай маємо задану множину альтернативних конфігурацій з виглядом (3.56):

$$S = \{s_1, s_2, \dots, s_n\}, \quad (3.56)$$

де кожна конфігурація s_i буде описана як узагальнений параметричний вектор виглядом представленого таким чином (3.57):

$$X(s_i) = [X_{tech}(s_i), X_{econ}(s_i), X_{ops}(s_i)], \quad (3.57)$$

Крім того, потрібно врахувати, що для системи задається множина зовнішніх обмежень представлених (3.58):

$$G = [B_{max}, A_{min}, BW_{max}, C_{max}], \quad (3.58)$$

Та враховується система вагових коефіцієнтів, що представлена наступним чином (3.59):

$$W = [\alpha_k, \beta_k, \gamma_k, w_{tech}, w_{econ}, w_{ops}] , \quad (3.59)$$

В результаті, задача вибору раціональної конфігурації має наступне визначення (3.60):

$$s_{opt} = \arg \max_{s_i \in \Omega} E_{total}^{AI}(s_i) , \quad (3.60)$$

де:

- $\Omega \subseteq S$ – множина допустимих значень конфігурацій;
- $E_{total}^{AI}(s_i)$ - скоригована інтегральна ефективність конфігурації з урахуванням впливу якості роботи аналітики.

Даний алгоритм виконує дві функції, а саме зменшує кількість всіх конфігурацій до списку лише допустимих, а також виконує процес ранжування допустимих конфігурацій за інтегральним критерієм ефективності та обирає найкращу.

3.4.2. Вхідні дані та формування множини допустимих рішень

У загальному вигляді вхідні дані алгоритму можуть бути подані як кортеж, що представлений нижче (3.61):

$$D = \langle S, M, G, W \rangle , \quad (3.61)$$

де:

- S – множина конфігурацій;
- $M = \{x_{ij}\}$ – матриця параметрів конфігурацій;
- G – множина обмежень;
- W – система вагових коефіцієнтів.

Першим етапом, даного алгоритму, буде перевірка допустимості кожної з конфігурацій. Для цього вводиться індикатор допустимості (3.62):

$$\delta_i = 1, \text{ якщо } B(s_i) \leq B_{\max}, A_i \geq A_{\min}, BW_i \leq BW_{\max,i}^{top}, C_i \leq C_{\max}, \quad (3.62)$$

$$\delta_i = 0, \quad \text{інакше}$$

Після цього множина допустимих конфігурацій формується за наступним визначенням (3.63):

$$\Omega = \{s_i \in S \mid \delta_i = 1\}, \quad (3.63)$$

Такий підхід відразу прибирає всі альтернативні конфігурації, які не можуть бути реалізовані в рамках заданих обмежень. Особливу увагу відіграє перевірка умови мережевого навантаження $BW_i \leq BW_{\max,i}^{top}$ з урахуванням вибраного архітектурного типу.

3.4.3. Розрахунок локальних критеріїв

Для кожної конфігурації $s_i \in \Omega$ виконується розрахунок локальних критеріїв ефективності. Перед цим параметри проходять процес нормалізації значень та після обчислюються локальні оцінки технічної (3.64), економічної (3.65) та функціональної ефективності (3.66):

$$E_{tech}(s_i) = \sum_{k=1}^m \alpha_k \widehat{x_{ik}^{(tech)}}, \quad (3.64)$$

$$E_{econ}(s_i) = \sum_{k=1}^p \beta_k \widehat{x_{ik}^{(econ)}}, \quad (3.65)$$

$$E_{ops}(s_i) = \sum_{k=1}^q \gamma_k \widehat{x_{ik}^{(ops)}}, \quad (3.66)$$

Як результат для кожної конфігурації формується вектор базових локальних оцінок (3.67):

$$E_{loc}(s_i) = [E_{tech}(s_i), E_{econ}(s_i), E_{ops}(s_i)] , \quad (3.67)$$

Цей етап дозволяє також провести порівняння конфігурацій за окремими типами критеріїв та без врахування впливу модулю аналітики.

3.4.4. Урахування компонента відеоаналітики

Особливістю даного алгоритму є окремий блок, який розраховує вплив та оцінку використання відеоаналітики. Для кожної конфігурації виконується розрахунок показників якості аналітики, стохастичних наслідків хибних спрацювань і пропуску подій, а також підхідних характеристик, що впливають на економічну та функціональну ефективність.

Загальним представленням можна показати цей етап у вигляді наступного перетворення (3.68):

$$\Psi_{AI}(s_i): E_{loc}(s_i) \rightarrow [E_{tech}^*(s_i), E_{econ}^*(s_i), E_{ops}^*(s_i)] , \quad (3.68)$$

де:

- $E_{tech}^*(s_i)$ – скоригована технічна ефективність з урахуванням якості аналітики;
- $E_{econ}^*(s_i)$ – скоригована економічна ефективність з урахуванням очікуваних витрат від пропущених інцидентів;
- $E_{ops}^*(s_i)$ – скоригована функціональна ефективність з урахуванням фактору втоми оператора та зміни функціонального покриття.

Цей блок дозволяє вирахувати, що помилки в роботі аналітики змінюють параметри ефективності для всіх груп критеріїв.

3.4.5. Обчислення інтегрального показника та ранжування альтернатив

Після отримання скоригованого значення локальних оцінок ефективності обчислюється інтегральний показник конфігурації (3.69):

$$E_{total}^{AI}(s_i) = w_{tech}E_{tech}^*(s_i) + w_{econ}E_{econ}^*(s_i) + w_{ops}E_{ops}^*(s_i), \quad (3.69)$$

Як результат для кожної конфігурації $s_i \in \Omega$ відповідає єдине числове значення, що показано у визначенні (3.70):

$$s_i \rightarrow E_{total}^{AI}(s_i), \quad (3.70)$$

Після цього виконується процес сортування допустимих альтернатив за спаданням інтегрального критерію (3.71):

$$E_{total}^{AI}(s_{(1)}) \geq E_{total}^{AI}(s_{(2)}) \geq \dots \geq E_{total}^{AI}(s_{(m)}), \quad (3.71)$$

де $m = |\Omega|$.

Процес ранжування дозволяє не лише визначати найкращу конфігурацію, а й отримати впорядкований список альтернатив, який може бути корисним для аналізу. За потреби може бути використаний додатковий показник ефективності, який показує співвідношення інтегральної ефективності до повної вартості конфігурації. (3.71):

$$K_i = \frac{E_{total}^{AI}(s_i)}{B(s_i)}, \quad (3.71)$$

3.4.6. Вибір оптимального рішення та обчислювальна складність алгоритму

Фінальний етап роботи алгоритму полягає у виборі оптимальної конфігурації за правилом (3.72):

$$s_{\text{opt}} = \arg \max_{s_i \in \Omega} E_{\text{total}}^{AI}(s_i), \quad (3.72)$$

У практичному розумінні результат алгоритму надає не лише саму конфігурацію системи, а й параметри, за якими була обрана найкраща, тобто причини її вибору. З точки зору обчислення нехай:

- n – кількість альтернативних конфігурацій;
- r – загальна кількість параметрів;
- $m = |\Omega|$ – кількість конфігурацій після фільтрації.

Тоді:

- Перевірка допустимості складається: $O(nr)$;
- Обчислення локальних та інтегральних критеріїв: $O(nr)$;
- Ранжування допустимих конфігурацій: $O(m \log m)$.

Отже, загальна обчислювальна формула алгоритму має наступний вигляд (3.73):

$$O(nr + m \log m), \quad (3.73)$$

З урахуванням того, що кількість параметрів r є відносно сталим показником, то запропонований метод залишається придатним для використання та аналізу великої кількості альтернатив. Такий висновок підтверджує, що даний метод є доцільний для реалізації його в програмному середовищі та подальшого його застосування.

Висновки до розділу 3

В третьому розділі було вперше розроблено формалізацію методу, його математичне обґрунтування, створено алгоритм та послідовність виконання методу оцінки ефективності та оптимізації інтегрованих систем відеоспостереження з елементами аналізу даних.

Конфігурація, яка передається на оцінку методу ефективності вважається параметричним об'єктом, для якого задано три вектори (технічний, економічний та функціональний), також для них було визначено домени значень, типи шкал вимірювання, а також застосування системи зовнішніх обмежень. Результатом стала формальна основа, яка використовується для побудови обчислювальної моделі методу.

Вперше було розроблено математичну модель інтегральної ефективності системи, яка дозволяє оцінити ефективність конфігурації на основі низки критеріїв різного типу. Для інтегральної ефективності визначено локальні значення ефективності для технічної, економічної та функціональної групи, які проходять через процедуру нормування значень до єдиного вигляду та агрегування результатів в єдиний інтегральний показник. Також розроблено багатокритеріальну форму моделі та штрафну форму моделі.

Вперше розроблено математичне врахування впливу точності відеоаналітичних інструментів на загальну ефективність системи. Виконано та формалізовано стохастичних характер помилок спричинених аналітичними детекторами та введено залежності, які дозволять оцінити рівень втоми та навантаження на оператора, а також рівень математичного очікування збитків через пропущені інциденти. Було продемонстровано, як такий вплив аналітики змінюють технічну, економічну та функціональну групу, а також їх прямий вплив на інтегральну оцінку конфігурації.

Вперше, для визначеної моделі оцінки ефективності, було розроблено алгоритм оптимізації, яка дозволяє обрати найбільш раціональну конфігурацію, яка включає в себе визначення та ввід вхідних даних, які задають обмеження для показників та критеріїв, перевірку допустимості альтернатив, розрахунок локальних показників

ефективності, урахування впливу аналітики та детекторів, визначення інтегрального показника для кожної допустимої конфігурації, їх ранжування та вибір найбільш оптимального рішення.

РОЗДІЛ 4

ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА ВАЛІДАЦІЯ ЗАПРОПОНОВАНОГО МЕТОДУ ОЦІНКИ ІНТЕГРАЛЬНОЇ ЕФЕКТИВНОСТІ ТА ОПТИІЗАЦІЇ КОНФІГУРАЦІЇ

4.1. Методика та організація експериментального дослідження

Після побудови математичної моделі інтегральної ефективності системи відеоспостереження та методу оптимізації виникає необхідність у його практичній реалізації та емпіричному дослідження. Ціллю даного розділу є оцінка працездатності розробленого методу, перевірка його прикладної придатності та апробація цих результатів і порівнянні з іншими, раніше розглянутими, методами.

Результати даного розділу мають показати, що побудований алгоритм та методологія є не лише правильною для застосування з теоретичної точки зору, а й такою, що здатна забезпечувати детальний, ітераційний та інженерно корисний результат який може застосовуватись для роботи над проектуванням системи відеоспостереження. Цей підхід забезпечує:

- Отримання кількісних показників, що є обґрунтованими та доведеними;
- Виявляти слабкі, нераціональні критерії відповідно до вимог системи;
- Виконувати аналіз та визначати найбільш раціональну конфігурацію в межах заданих вхідних даних;
- Наочно презентувати результати та їх порівняння.

В рамках, даного практичного дослідження процес перевірки відбуватиметься за принципом етапності – від початкових розрахунків до розрахунку інтегрального показника ефективності відповідно до блок-схеми на рис.4.1.

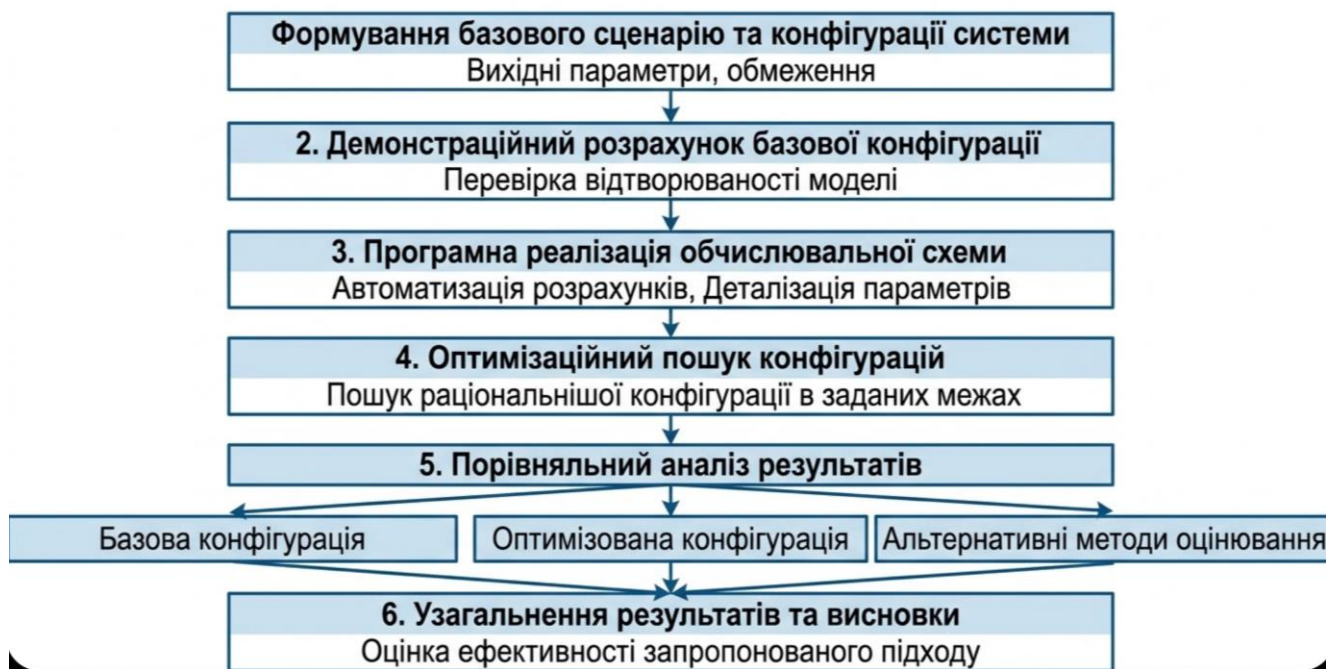


Рис. 4.1. Схема проведення експериментального дослідження.

На першому етапі відбувається збір даних та фіксація вхідної інформації про проект для подальших розрахунків. Другий етап відведений для демонстраційних математичних розрахунків заданої конфігурації з метою прозорості роботи методу. На третьому етапі використовується розроблене програмне забезпечення, яке дозволяє працювати з алгоритмом оптимізації перебираючи доступні альтернативні конфігурації в рамках заданих діапазонів обмеження. Фінальний, четвертий, етап порівнює базову модель конфігурації зі знайденою оптимізованою моделлю, що дозволяє оцінити її рентабельність та раціональність до впровадження.

Важливо зазначити, що для дослідженні використовується лише один об'єкт з конкретними вимогами для чесного та математичного порівняння результатів оцінки його відносно інших методів. Такий підхід зумовлений тим, що навіть за наявності схожих показників вхідних даних, кожен проект має свою специфіку та свої вимоги щодо побудови, як наприклад архітектура, точність аналітики, критичність та інші параметри, які дозволяють наділити об'єкт необхідними характеристиками за допомогою вагових критеріїв.

Умовно, відбувається поділ на 4 групи даних, за якими відбуватимуться математичні операції. Перша група – це параметри, які характеризують об’єкт. Сюди відноситься кількість відеокамер, роздільна здатність, частота кадрів, інтенсивність інцидентів, навантаження на оператора.

Друга група – це параметри конфігурації, які можуть бути змінені під час дослідження. До таких змінних належить тип архітектури, точність роботи аналітичних інструментів, затримка обробки даних, кількість хибних спрацювань, фінансові затрати на реалізацію та експлуатацію системи, рівень автоматизації та інше. Дана група параметрів приймає участь в пошуку альтернативних конфігурацій за допомогою методу оптимізації.

Третя група – це показники зовнішніх обмежень, які задають кордони як мінімально так і максимально допустимі для альтернативних конфігурацій. Сюди входять бюджетні обмеження, мінімально допустимий рівень аналітики, максимальне значення пропускної здатності мережі та інші.

Четверта група – це вихідні показники, які отримуються в результаті виконання методу оцінки ефективності та методу оптимізації ІСВ. До таких параметрів ми відносимо інтегральна ефективність конфігурації, а також локальні ефективності за групами. Такий підхід є ефективним, оскільки він демонструє не тільки порядок ефективностей, а й можливість інтерпретації причин та параметрів, по яким така оптимальна конфігурація визначена кращою.

Відповідний перелік вхідних параметрів, які будуть предметом розрахунку представлені в таблиці 4.1.

Таблиця.4.1.

Перелік вхідних параметрів

Група параметрів	Назва параметра	Позначення в моделі	Одиниця вимірювання	Роль у розрахунку
------------------	-----------------	---------------------	---------------------	-------------------

Сценарні	Кількість камер / каналів	N_{ch}	каналів	Визначає масштаб системи
Сценарні	Середня роздільна здатність камери	M_{cam}	Мп	Впливає на пропускну здатність і навантаження
Сценарні	Частота кадрів	FPS	кадр/с	Впливає на мережеве навантаження та обчислення
Сценарні	Коефіцієнт стиснення відео	K_{comp}	—	Коригує інтенсивність потоку даних
Сценарні	Середня кількість реальних інцидентів за годину	I_{real}	подій/год	Використовується для розрахунку TP, FN та збитків
Сценарні	Фонових подій за годину	I_{bg}	подій/год	Використовується для оцінки FP і втоми оператора
Сценарні	Подій на оператора за годину	E_{op}	подій/год	Впливає на індекс втоми оператора
Сценарні	Середній час обробки однієї події оператором	t_{op}	хв	Використовується у формулі втоми та вартості обробки
Сценарні	Вартість години роботи оператора	$C_{op,h}$	грн/год	Використовується для оцінки вартості FP
Технічні	Тип архітектури системи	$Arch_i$	—	Впливає на BW, затримку, надійність, OPEX
Технічні	Точність аналітики	A_i	частка	Основний технічний показник

Технічні	Частка хибних спрацьовувань	FP_i	частка	Впливає на precision, втому та витрати
Технічні	Частка пропусків інцидентів	FN_i	частка	Впливає на recall, покриття та збитки
Технічні	Базова затримка обробки	L_i	мс	Формує технічну ефективність
Технічні	Базове обчислювальне навантаження	C_i	%	Перевіряється на допустимість
Технічні	Базова надійність системи	R_i	частка	Формує технічну ефективність
Технічні	Базова відмовостійкість	F_i	частка	Формує технічну ефективність
Економічні	Початкові витрати без ліцензій	$CAPEX_i$	грн	Основа для сукупних початкових витрат
Економічні	Вартість ліцензії на канал	C_{lic}	грн/канал	Формує ліцензійну складову
Економічні	Експлуатаційні витрати на рік	$OPEX_i$	грн/рік	Формує економічну ефективність
Економічні	Вартість масштабування	C_{scale}	грн	Оцінка потенціалу розвитку системи
Економічні	Середній збиток від пропуску одного інциденту	L_{inc}	грн	Основа для очікуваного збитку
Економічні	Строк окупності	P_i	міс.	Один з економічних критеріїв
Функціональні	Покриття задач безпеки	S_i	частка	Відображає функціональну повноту системи

Функціональні	Релевантність аналітики задачам об'єкта	Q_i	частка	Характеризує відповідність аналітики потребам
Функціональні	Оперативність реагування	T_i	частка	Впливає на функціональну ефективність
Функціональні	Рівень автоматизації	U_i	частка	Впливає на функціональну ефективність

4.2. Формування вхідних даних і базової конфігурації системи

Після визначення загальної методики проведення дослідження необхідно сформувати вхідні дані, які будуть основою для проведення аналізу та оцінки. Доцільно розглядати не випадкові параметри, а використовувати актуальні проекти, які мають реальне застосування таких систем.

У межах даної роботи розгадається конфігурація інтегрованої системи відеоспостереження з елементами аналізу даних з фіксованим масштабом, що означає сталу кількість камер відеоспостереження та функціональних задач. Таким об'єктом визначено торгово-розважальний центр, зі значною кількістю відеокамер.

Даний проект має зовнішні обмеження, які відображені в таблиці 4.2. та будуть враховані в розрахунках, як гранично мінімальні та максимальні параметри, а також вагові коефіцієнти критичності різних локальних груп ефективності.

Таблиця 4.2.

Вхідні граничні параметри для проведення дослідження

Назва обмеження	Позначення	Значення	Одиниця	Призначення
Максимально допустимий бюджет	$B_{\max}$	22 000 000	грн	Обмеження допустимих рішень за вартістю

Мінімально допустима точність аналітики	$A_{\min}$	0.92	частка	Відсікання недостатньо ефективних рішень
Глобальний ліміт мережевого навантаження	$BW_{\max}$	12 000	Мбіт/с	Контроль допустимої пропускної здатності
Максимально допустиме обчислювальне навантаження	$C_{\max}$	90	%	Контроль ресурсоемності системи
Ваговий технічний коефіцієнт	w_{tech}	0.4	частка	Приоритетність ефективності для даної групи критеріїв
Ваговий економічний коефіцієнт	w_{econ}	0.3	частка	Приоритетність ефективності для даної групи критеріїв
Ваговий функціональний коефіцієнт	w_{ops}	0.3	частка	Приоритетність ефективності для даної групи критеріїв

З точки зору практичного застосування та прозорості, базова конфігурація системи для дослідження сформована таким чином, щоб її параметри не перебували в завчасно сприятливій області, а також не було від'ємне значення ефективності – несприятливих для реалізації. Тобто визначена базова конфігурація, яка буде розглянута в рамках дисертаційного дослідження, є цілком робочою і дієздатною до впровадження, але не така, що дозволяє залишити наміряно простір для демонстрації поліпшень ефективності системи. Дані для базової конфігурації, за якими відбуватимуться розрахунки, представлені в таблиці 4.3.

Вхідні дані конфігурації системи

Група параметрів	Назва параметра	Значення	Одиниця
Сценарні	Кількість камер / каналів	700	каналів
Сценарні	Середня роздільна здатність камери	4	Мп
Сценарні	Частота кадрів	30	кадр/с
Сценарні	Коефіцієнт стиснення відео	1	—
Сценарні	Середня кількість реальних інцидентів за годину	6	подій/год
Сценарні	Фонових подій за годину	420	подій/год
Сценарні	Подій на оператора за годину	180	подій/год
Сценарні	Середній час обробки однієї події оператором	2.5	хв
Сценарні	Вартість години роботи оператора	250	грн/год
Технічні	Тип архітектури системи	гібридна	—
Технічні	Точність аналітики	0.94	частка
Технічні	Частка хибних спрацьовувань	0.06	частка
Технічні	Частка пропусків інцидентів	0.08	частка
Технічні	Базова затримка обробки	420	мс
Технічні	Базове обчислювальне навантаження	78	%
Технічні	Базова надійність системи	0.985	частка
Технічні	Базова відмовостійкість	0.96	частка
Економічні	Початкові витрати без ліцензій	12 800 000	грн
Економічні	Вартість ліцензії на канал	7 200	грн/канал

Економічні	Експлуатаційні витрати на рік	1 550 000	грн/рік
Економічні	Вартість масштабування	2 800 000	грн
Економічні	Середній збиток від пропуску одного інциденту	450 000	грн
Економічні	Строк окупності	38	міс.
Функціональні	Покриття задач безпеки	0.9	частка
Функціональні	Релевантність аналітики задачам об'єкта	0.92	частка
Функціональні	Оперативність реагування	0.85	частка
Функціональні	Рівень автоматизації	0.78	частка

4.3. Демонстраційний розрахунок базової конфігурації

Розрахунки виконуються послідовно відповідно до математичної моделі розробленою в розділі 3. Варто врахувати, що існують додаткові параметри, які є змінювані, що застосовуються до різних параметрів, що були визначені емпірично і в разі необхідності можуть бути скориговані по відношенню до умов конкретного об'єкту.

Тип системи, що використовується – гібридна, що означає використання наступних змінних впливу вибору архітектури, де $k_{arch}^{BW} = 0.95$, $k_{arch}^{cost} = 1.10$, $k_{arch}^{opex} = 1.08$, $k_{arch}^{scale} = 1.08$, $k_{arch}^{lat} = 0.95$, $k_{arch}^{comp} = 1.05$, $\Delta R = 0.012$, $\Delta F = 0.02$, $BW_{max}^{top} = 20000$ Мбіт/с.

Розрахунок пропускної здатності мережі (4.1):

$$\begin{aligned}
 BW_I &= N_{ch} \cdot M_{cam} \cdot FPS \cdot 0,12 \cdot K_{comp} \cdot k_{arch}^{BW} \\
 BW_i &= 700 \cdot 4 \cdot 30 \cdot 0.12 \cdot 1 \cdot 0.95 = 9576
 \end{aligned}
 \tag{4.1}$$

Розрахунок сукупних початкових витрат на конфігурацію (4.2):

$$C0_i = (CAPEX + C_{lic,\Sigma}) \cdot k_{arch}^{cost}$$

$$C0_i = (12800000 + 5040000) \cdot 1.10 = 19624000 \quad (4.2)$$

Розрахунок експлуатаційних витрат (4.3):

$$C_{op_i} = 1550000 \cdot 1.08 = 167400 \quad (4.3)$$

Вартість масштабування (4.4):

$$C_{sc_i} = 2800000 \cdot 1.08 = 3024000 \quad (4.4)$$

Скоригована затримка (4.5):

$$L_i = 420 \cdot 0.95 = 399 \quad (4.5)$$

Скориговане обчислювальне навантаження (4.6):

$$C_i = 78 \cdot 1.05 = 81.9 \quad (4.6)$$

Скоригована надійність та відмовостійкість (4.7):

$$R_i = 0.985 + 0.012 = 0.997$$

$$F_i = 0.960 + 0.02 = 0.980 \quad (4.7)$$

Розрахунок аналітичного фактору правильності виявлення інциденту (4.8):

$$TP = 6 \cdot (1 - 0.88) = 5.52 \quad (4.8)$$

Розрахунок пропущеного інциденту аналітикою (4.9):

$$FN = 6 \cdot 0.08 = 0.48 \quad (4.9)$$

Хибні події згенеровані аналітикою (4.10):

$$FP = 420 \cdot 0.06 = 25.2 \quad (4.10)$$

Розрахунок параметру Precision (4.11):

$$\text{Precision} = \frac{5.52}{5.52+25.2} = 0.18 \quad (4.11)$$

Розрахунок параметру Recall (4.12):

$$\text{Recall} = \frac{5.52}{5.52+0.48} = 0.92 \quad (4.13)$$

Розрахунок параметру F1-score (4.13):

$$F1 = \frac{2 \cdot 0.180 \cdot 0.920}{0.180 + 0.920} \approx 0.301 \quad (4.13)$$

Розрахунок індексу втоми оператора (4.15) з ваговими коефіцієнтами (4.14):

$$q_0 = 0.15, k_1 = 0.006, k_2 = 1.10 \quad (4.14)$$

$$Q_{fat}^{raw} = 0.15 + 0.006 \cdot 180 \cdot 2.5 + 1.10 \cdot 0.06 = 2.916 \quad (4.15)$$

Скориговане покриття задач безпеки (4.16):

$$S_i^* = 0.90 \cdot (1 - 0.8 \cdot 0.08) = 0.8424 \quad (4.16)$$

Скоригованої релевантності аналітики (4.17):

$$Q_i^* = 0.92 - 0.5 \cdot 0.08 - 0.3 \cdot 0.06 = 0.862 \quad (4.17)$$

Розрахунок скоригованої оперативності реагування (4.18):

$$T_i^* = 0.85 - 0.25 \cdot 1.0 - 0.10 \cdot 0.08 = 0.592 \quad (4.18)$$

Розрахунок очікуваного збитку від пропущених інцидентів (4.19):

$$Linc_i = 450\,000 \cdot 0.48 = 216\,000 \quad (4.19)$$

Розрахунок вартості обробки хибних тривог (4.20):

$$Cfp_i = 250 \cdot \frac{25.2 \cdot 2.5}{60} = 262.5 \quad (4.20)$$

Розрахунок корисних детекцій на рік (4.21):

$$D_{useful} = 5.52 \cdot 24 \cdot 365 = 48\,355.2 \quad (4.21)$$

Розрахунок вартості корисної детекції (4.22):

$$Cdet_i = \frac{19\,624\,000/3 + 1\,674\,000}{48\,355.2} = 169.90 \quad (4.22)$$

Розрахунок комплексної вартості одного каналу (4.23):

$$Cch_i = \frac{19\,624\,000}{700} = 28\,034.29 \quad (4.23)$$

Нормування критеріїв (4.24):

$$\begin{aligned} A_i^{norm} &= 0.94 \\ FP_i^{norm} &= \frac{0.50 - 0.06}{0.50} = 0.88 \\ FN_i^{norm} &= \frac{0.50 - 0.08}{0.50} = 0.84 \\ L_i^{norm} &= \frac{1500 - 399}{1500 - 100} = \frac{1101}{1400} = 0.786 \\ BW_i^{norm} &= \frac{12000 - 9576}{12000} = 0.202 \\ C_i^{norm} &= \frac{90 - 81.9}{90} = 0.090 \\ R_i^{norm} &= 0.997 \\ F_i^{norm} &= 0.980 \\ CO_i^{norm} &= \frac{22\,000\,000 - 19\,624\,000}{22\,000\,000} = 0.108 \\ Cop_i^{norm} &= 0 \\ Csc_i^{norm} &= 0 \\ Cch_i^{norm} &= \frac{50\,000 - 28\,034.29}{50\,000} = 0.439 \\ Cdet_i^{norm} &= \frac{2000 - 169.90}{2000} = 0.915 \\ Linc_i^{norm} &= \frac{2\,700\,000 - 216\,000}{2\,700\,000} = 0.920 \\ Cfp_i^{norm} &= \frac{50\,000 - 262.5}{50\,000} = 0.995 \end{aligned} \quad (4.24)$$

$$\begin{aligned}
P_i^{norm} &= \frac{60 - 38}{60} = 0.367 \\
S_i^{norm} &= 0.8424 \\
Q_i^{norm} &= 0.862 \\
Qfat_i^{norm} &= 1 - Q_{fat} = 1 - 1.0 = 0 \\
T_i^{norm} &= 0.592 \\
U_i^{norm} &= 0.78
\end{aligned}$$

Розрахунок базової технічної ефективності (4.25):

$$\begin{aligned}
E_{tech}^{base} &= 0.22 \cdot 0.94 + 0.13 \cdot 0.88 + 0.18 \cdot 0.84 + 0.09 \cdot 0.786 + \\
&0.08 \cdot 0.202 + 0.10 \cdot 0.090 + 0.10 \cdot 0.997 + 0.10 \cdot 0.980 = 0.766
\end{aligned} \quad (4.25)$$

Розрахунок якості AI-компоненту (4.26):

$$E_{tech}^{AI} = 0.30 \cdot 0.94 + 0.25 \cdot 0.180 + 0.25 \cdot 0.920 + 0.20 \cdot 0.301 = 0.617 \quad (4.26)$$

Розрахунок загальної технічної ефективності (4.27):

$$E_{tech} = 0.70 \cdot 0.766 + 0.30 \cdot 0.617 = 0.721 \quad (4.27)$$

Розрахунок базової технічної ефективності (4.28) з використанням вагових коефіцієнтів (4.29):

$$\begin{aligned}
\alpha_{c0} &= 0.225, \alpha_{cop} = 0.1625, \alpha_{csc} = 0.125, \\
\alpha_{cch} &= 0.15, \alpha_{cdet} = 0.2125, \alpha_P = 0.125
\end{aligned} \quad (4.28)$$

$$E_{econ}^{base} = 0.23 \cdot 0.11 + 0.16 + 0.13 + 0.15 \cdot 0.44 + 0.21 \cdot 0.92 + 0.13 \cdot 0.37 = 0.33 \quad (4.29)$$

Розрахунок ризикової економічної складової (4.30):

$$E_{econ}^{risk} = 0.7 \cdot 0.92 + 0.3 \cdot 0.995 = 0.942 \quad (4.30)$$

Розрахунок загальної економічної ефективності (4.31):

$$E_{econ} = 0.75 \cdot 0.330 + 0.25 \cdot 0.942 = 0.483 \quad (4.31)$$

Розрахунок функціональної ефективності (4.32):

$$E_{ops} = 0.25 \cdot 0.8424 + 0.2 \cdot 0.862 + 0.2 \cdot 0 + 0.2 \cdot 0.592 + 0.15 \cdot 0.78 = 0.618 \quad (4.32)$$

Розрахунок інтегральної ефективності (4.33):

$$E_{total} = 0.40 \cdot 0.721 + 0.30 \cdot 0.483 + 0.30 \cdot 0.618 = 0.619 \quad (4.33)$$

Враховуючи, що для даної конфігурації всі обмеження були дотримані то штраф на загальну ефективність не застосовується, а тому інтегральне значення ефективності більше не зазнає змін. Тоді загальні результати матимуть наступний вигляд (4.34):

$$\begin{aligned} E_{tech} &= 0.721 \\ E_{econ} &= 0.483 \\ E_{ops} &= 0.618 \\ E_{total} &= 0.619 \end{aligned} \quad (4.34)$$

4.4. Програмна реалізація методу

Для виконання практичного завдання та розрахунків розробленого методу оцінки ефективності та оптимізації було розроблено програмне забезпечення, з використанням Python та бібліотеки Streamlit. В розробленому прототипі повністю реалізована логіка описаного методу від початкових розрахунків до репрезентації та фінальних результатів.

Програмна реалізація дає ряд переваг щодо дослідження і дозволяє швидко та без затримок в розрахунках оцінити роботу методу та значення змінних змінюючи критерії та параметри. На відміну від ручних розрахунків, які потрібно виконувати ітераційно в разі зміни базових вхідних величин або обмежень, що займає значний об'єм часу, програмна реалізація дозволяє виконати такі розрахунки з мінімальною затримкою. Крім того, розроблений зручний інтерфейс, який дозволяє проводити відразу порівняння величин, їх графічне представлення та порівняння. Окремо, для застосування методу оптимізації, такий підхід є особливо правильним, оскільки він передбачає перебір різних конфігурацій та порівняння величин, що для великих діапазонів обмежень може творити значний масив конфігурацій, для розрахунку яких потрібно буде велика кількість часу, а з використанням програмного алгоритму вони відбуваються за декілька секунд.

Структурно, такий прототип калькулятора ефективності ICB та оптимізації, складається з двох частин – Frontend та Backend, що представлено на Рис. 4.2.

Перший блок забезпечує для програмного забезпечення можливість введення даних користувачем, взаємодію з інтерфейсом та відображення результатів роботи алгоритму (значення, графіки, гістограми). Другий блок реалізований для виконання розрахунку математичних функцій та передачі результатів в блок відображення (Frontend). Backend складається з трьох модулів: агрегації векторних даних, математичне ядро системи де відбувається процес розрахунку оптимізації та згортки, генерація даних для побудови графіків та діаграм.

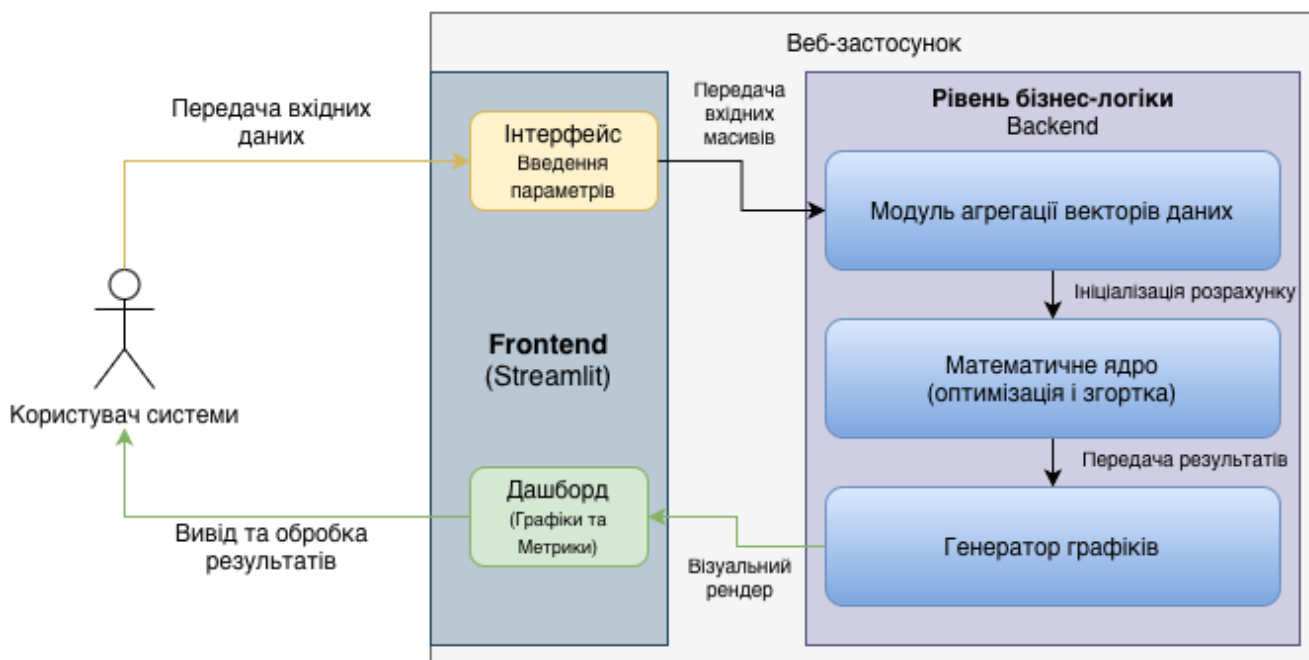


Рис. 4.2. Архітектура роботи програмного прототипу методу оцінки ефективності та оптимізації ICB

Послідовність виконання розрахунків відповідає розробленому методу оцінки ефективності та оптимізації та реалізований за наступним алгоритмом, що зображений на блок-схемі рис.4.3.

Для зручності послідовного виконання процесу було визначено 5 вікон інтерфейсу програми калькулятора:

1. Калькулятор ефективності. Містить в собі поля та повзунки для фіксації вхідних даних базової конфігурації та розрахунку її базової ефективності.
2. Деталізовані результати. Відображає всі розраховані параметри в процесі визначення ефективності та їх репрезентативне представлення через таблиці та графіки.
3. Оптимізація. Область де задається діапазон допустимих значень для знаходження оптимуму конфігурації ICB.
4. Результати оптимізації. Представлення найкращих 10 конфігурацій та конфігурації, яка є найкращою в заданому діапазоні, їх порівняння через кінцеві значення розрахунків та детальні показники еталонного.



Рис. 4.3. Структурна схема програмної реалізації методу

5. Порівняння конфігурацій. Відображення даних ефективності базової конфігурації та порівняння їх показників з оптимальною конфігурацією, які представлені в числовому, табличному та графічному вигляді.

Таким чином, даний програмний підхід дозволяє з мінімальними часовими затратами виконати процеси оцінки ефективності системи, пошук оптимуму серед допустимих конфігурацій та за необхідності внести коригування до вхідних даних і миттєво отримати оновлені значення розрахунків.

4.5. Результати оптимізації конфігурації системи

Для перевірки коректності розрахунків які проведені демонстраційно в ручному режимі а також правильності роботи програмного прототипу в даному підрозділі представлені та розглянуті результати розрахунків саме програмного веб-застосунку, так як перші представлені в попередніх підрозділах. Це означає, що виконується процес взаємної перевірки результатів.

Передбачено незначні відмінності в отриманих результатах, оскільки ручний розрахунок допускає неточне округлення величин, що є похибкою роботи з числами з плаваючою комою, тоді як програмні розрахунки є точнішими.

На першому етапі виконуємо внесення всіх вхідних даних, що були визначені в підрозділі 4.2 в програмний інтерфейс, що продемонстровано на рис. 4.4.

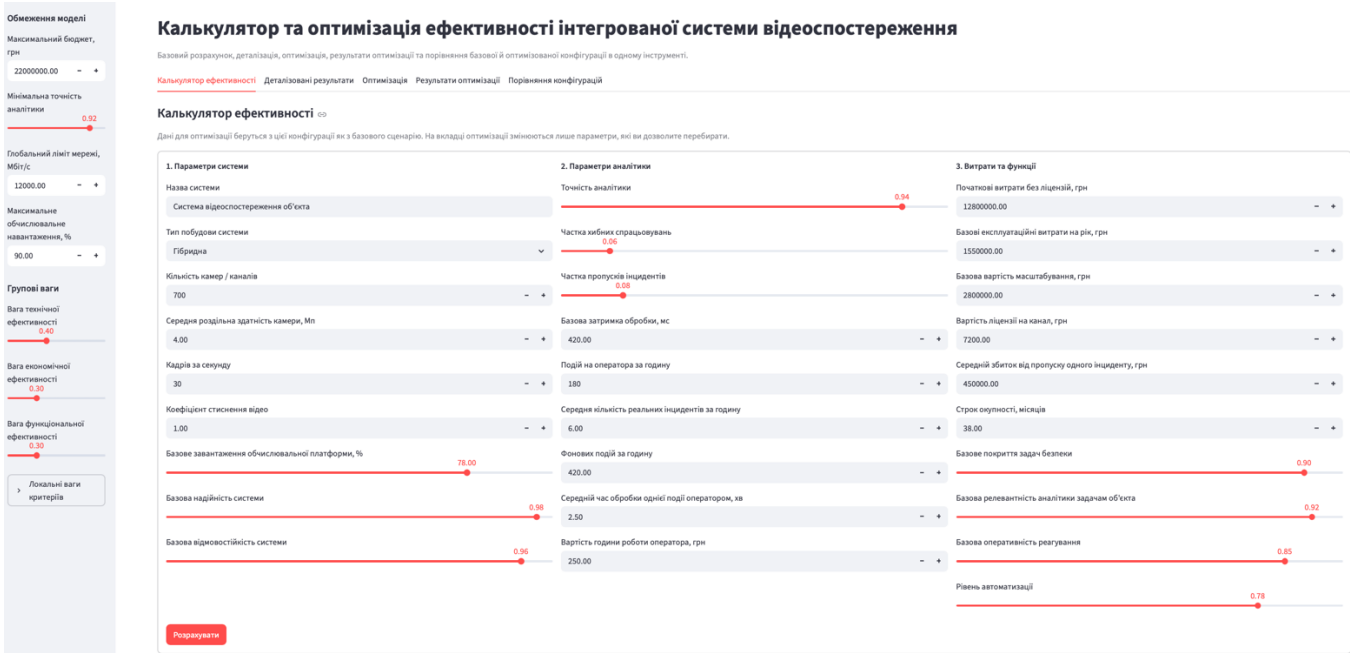


Рис. 4.4. Інтерфейс системи для введення вхідних даних

Після завершення процесу внесення даних натискаємо на клавішу «Розрахувати» та отримуємо результати, які продемонстровані на рис. 4.5.

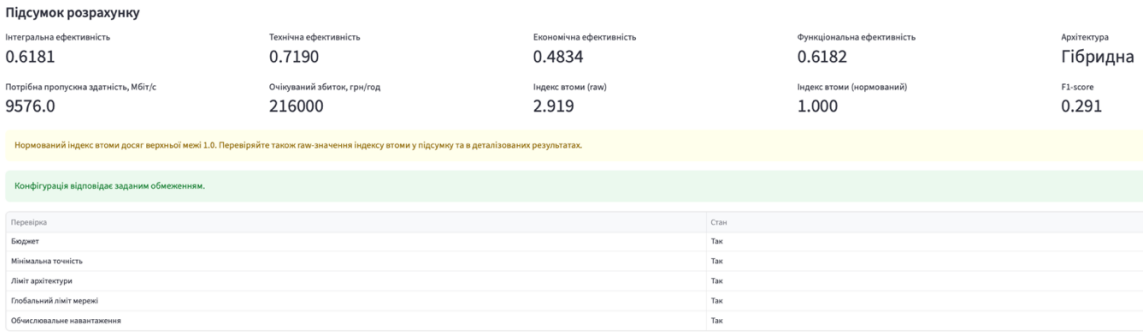


Рис. 4.5. Розрахунок базової ефективності

По завершенню роботи алгоритму ми отримуємо результати розрахунків, які включають в себе локальні ефективності заданої конфігурації, їх інтегральний показник, а також додаткові значення, які дозволяють оцінити, чи відповідають дані результати заданим вхідним обмеженням. Тобто це загальні вхідні дані, які ми можемо використовувати для порівняння з ручними розрахунками.

Окремо, для перевірки та відладки всіх процесів роботи прототипу та алгоритму, та зручності оцінки результатів, розроблено відповідне інтерфейсне вікно «Деталізовані результати», що представлено на рис. 4.6., де в графічному представленні візуально показують чисельні значення, а також таблицю, яка містить всі змінні, як базові так і розраховані математично, для проведення порівняння та знаходження помилки, якщо така буде виявлена в процесі порівняння.

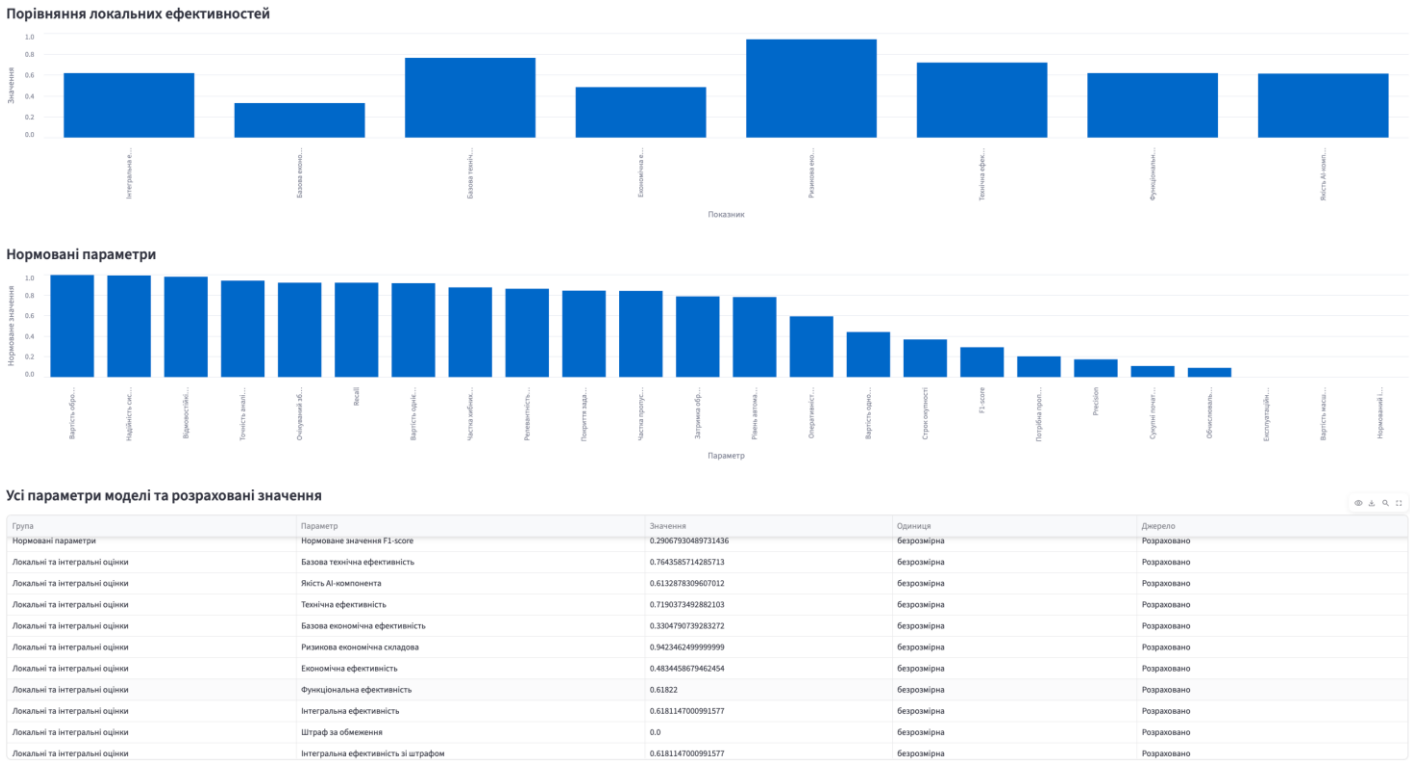


Рис. 4.6. Деталізовані результати роботи алгоритму оцінки базової ефективності

Таким чином маючи всі вхідні дані, розраховані локальні критерії та значення ефективностей відбувається перевірка результатів в таблиці 4.4.

Порівняння результатів розрахунку

Показник	Демонстраційний розрахунок	Програмний розрахунок	Висновок
Технічна ефективність	0.721	0.7190	Збігається
Економічна ефективність	0.483	0.4834	Збігається
Функціональна ефективність	0.618	0.6182	Збігається
Інтегральна ефективність	0.619	0.6181	Збігається

Після перевірки результатів та визначення збігів параметрів ефективностей зазначаємо, що метод коректно працює для розрахунків в ручному форматі та з використанням програмних засобів і пройшов взаємну перевірку.

4.6. Порівняльний аналіз базової та оптимізованої конфігурації

Визначення оптимальної конфігурації системи через розроблений метод оптимізації вимагає створення діапазону допустимих значень, які дозволяють створити простір для потенційних можливих конфігурацій, які між собою мають різні параметри ефективності. В залежності від важливості визначених критеріїв ефективності для них застосовуються також параметри вагових коефіцієнтів. Це означає, що система передає ранжований набір конфігурацій, що можна трактувати як раціональні в межах заданих параметрів.

Таким чином, здаємо відповідний діапазон допустимих значень для формування оптимізації системи представлений в таблиці 4.5.

Діапазон значень для оптимізації конфігурації

№	Параметр	Діапазон / значення	Одиниця
1	Архітектури для перебору	Всі	—
2	Кількість варіантів	1000	варіантів
3	Початкове значення генератора	1	—
4	Точність аналітики	0.90 – 1.00	частка
5	Частка хибних спрацьовувань	0.03 – 0.10	частка
6	Частка пропусків інцидентів	0.03 – 0.13	частка
7	Базова затримка обробки	250 – 650	мс
8	Базове обчислювальне навантаження	82 – 100	%
9	Початкові витрати без ліцензій	900 000 – 4 000 000	грн
10	Експлуатаційні витрати на рік	120 000 – 650 000	грн/рік
11	Вартість масштабування	180 000 – 1 110 000	грн
12	Покриття задач безпеки	0.72 – 1.00	частка
13	Релевантність аналітики задачам об'єкта	0.72 – 1.00	частка
14	Оперативність реагування	0.72 – 1.00	частка
15	Рівень автоматизації	0.60 – 1.00	частка

Після заповнення відповідних даних у розділі «Оптимізація» для виконання розрахунку натискаємо на клавішу «Запустити оптимізацію», що показано на рис.4.7. Після виконання даного процесу, ми отримуємо коротке заключення, що оптимізація завершена з визначим діапазоном кількості варіантів для генерації 1000.

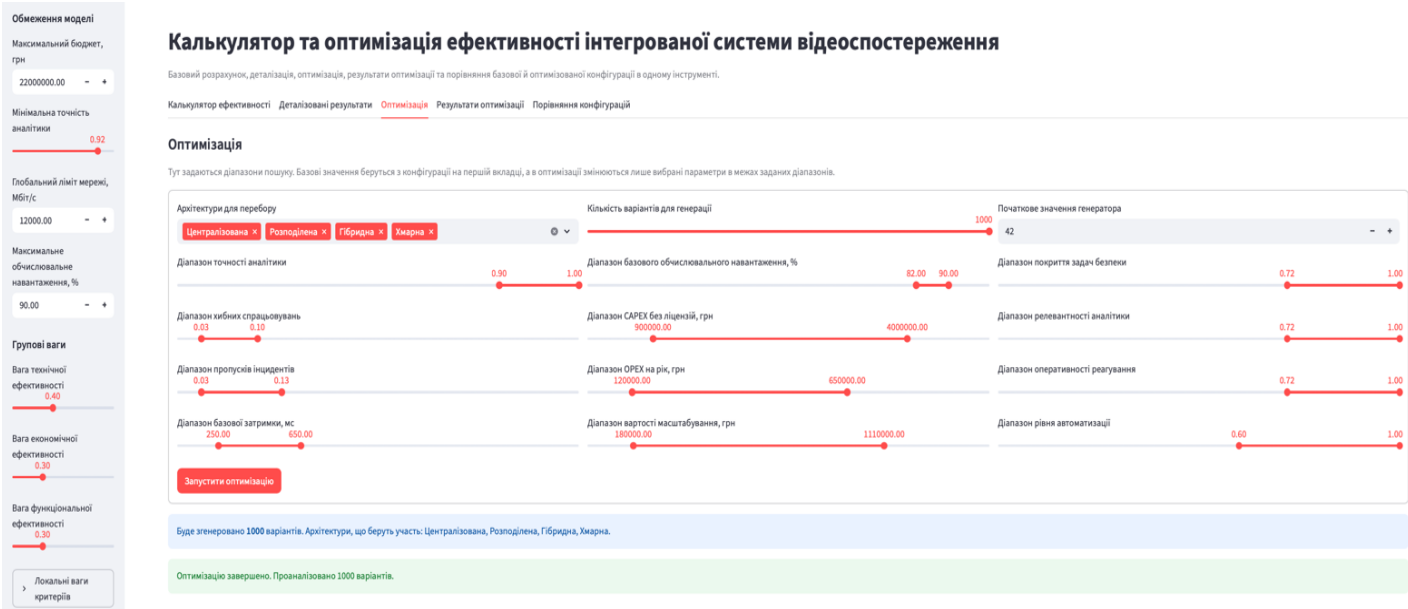


Рис. 4.7. Введення даних для запуску процесу оптимізації

Четверта вкладка програмного прототипу дозволяє оцінити виконану роботу алгоритму оптимізації та відобразити найкращу конфігурацію системи та її загальні показники ефективності. Додатково, в даному вікні, міститься інформація щодо 10 найкращих конфігурацій, які було виявлено під час процесу згортки та порівнянь. Визначені параметри відображені для порівняння в табличному представленні з числовими значеннями та відповідні графіки для наглядного представлення цих результатів, що показано на рис 4.8. та 4.9.

Група	Параметр	Значення	Одиниця	Джерело
Похідні показники аналітики	F1-score	0.213921590287388	частка	розраховано
Похідні показники аналітики	Корисні детекції за рік	50588.763452217056	детекцій/рік	Розраховано
Нормовані параметри	Точність аналітики	0.9397307477290424	безрозмірна	Розраховано
Нормовані параметри	Частка хибних спрацювань	0.8377856090837286	безрозмірна	Розраховано
Нормовані параметри	Частка пропусків інцидентів	0.9249909989428104	безрозмірна	Розраховано
Нормовані параметри	Затримка обробки	0.8773469777262692	безрозмірна	Розраховано
Нормовані параметри	Потрібна пропускна здатність	0.202	безрозмірна	Розраховано
Нормовані параметри	Обчислювальне навантаження	0.034377446263629784	безрозмірна	Розраховано
Нормовані параметри	Надійність системи	0.99	безрозмірна	Розраховано
Нормовані параметри	Відмовостійкість системи	0.978	безрозмірна	Розраховано
Нормовані параметри	Сукупні початкові витрати	0.5722700666570554	безрозмірна	Розраховано
Нормовані параметри	Експлуатаційні витрати на рік	0.7096706578209421	безрозмірна	Розраховано
Нормовані параметри	Вартість масштабування	0.0	безрозмірна	Розраховано
Нормовані параметри	Вартість одного каналу	0.7311411847558633	безрозмірна	Розраховано
Нормовані параметри	Вартість однієї корисної детекції	0.9675634407229817	безрозмірна	Розраховано
Нормовані параметри	Очікуваний збиток від пропуску інцидентів	0.9624954994714051	безрозмірна	Розраховано
Нормовані параметри	Вартість обробки хибних подій	0.9929031203974131	безрозмірна	Розраховано
Нормовані параметри	Строк окупності	0.36666666666666664	безрозмірна	Розраховано
Нормовані параметри	Покриття задач безпеки з урахуванням пропусків	0.8861027989693269	безрозмірна	Розраховано
Нормовані параметри	Релевантність аналітики задачам об'єкта	0.9498221526587951	безрозмірна	Розраховано
Нормовані параметри	Нормований індекс втоми оператора	0.0	безрозмірна	Розраховано
Нормовані параметри	Оперативність реагування з урахуванням втоми	0.6661029521550837	безрозмірна	Розраховано
Нормовані параметри	Рівень автоматизації	0.9414869820549853	безрозмірна	Розраховано
Нормовані параметри	Нормоване значення precision	0.14495415935398423	безрозмірна	Розраховано
Нормовані параметри	Нормоване значення recall	0.9624954994714052	безрозмірна	Розраховано
Нормовані параметри	Нормоване значення F1-score	0.2519621996287388	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Базова технічна ефективність	0.777510246112707	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Якість AI-компонента	0.6091740789508078	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Технічна ефективність	0.7270093959641373	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Базова економічна ефективність	0.605193989094087	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Ризикова економічна складова	0.9716177857492074	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Економічна ефективність	0.6967999382578671	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Функціональна ефективність	0.6859337680133553	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Інтегральна ефективність	0.7056238702670217	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Штраф за обмеження	0.0	безрозмірна	Розраховано
Локальні та інтегральні оцінки	Інтегральна ефективність зі штрафом	0.7056238702670217	безрозмірна	Розраховано
Вагові коефіцієнти	Вага технічної ефективності	0.4	частка	Налаштування моделі
Вагові коефіцієнти	Вага економічної ефективності	0.3	частка	Налаштування моделі
Вагові коефіцієнти	Вага функціональної ефективності	0.3	частка	Налаштування моделі

Рис. 4.8. Таблиця всіх змінних оптимізованої конфігурації

Результати оптимізації

Рекомендований варіант: №990 — Гібридна

Інтегральна ефективність		Інтегральна ефективність зі штрафом		Архітектура		Індекс втоми (raw)									
0.7056		0.7056		Гібридна		2.939									
Ранг	Варіант	Архітектура	Допустимий	Інтегральна ефективність	Інтегральна ефективність зі штрафом	Технічна ефективність	Економічна ефективність	Функціональна ефективність	Точність аналітики	Хибні спрацювання	Пропущені інциденти	Затримка обробки	Потрібна пропускна здатність	Сумарні початкові витрати	ОPEX на рік
1	990	Гібридна		0.7056	0.7056	0.727	0.6968	0.6899	0.9397	0.0811	0.0375	271.7142	9576	9410058.5335	145164.6711
2	933	Гібридна		0.7049	0.7049	0.7333	0.7513	0.6207	0.9841	0.0364	0.0876	578.9219	9576	7793665.6319	232929.3178
3	380	Гібридна		0.7038	0.7038	0.7346	0.7135	0.6529	0.9837	0.0677	0.0413	415.5173	9576	9829039.518	227804.8413
4	541	Гібридна		0.6948	0.6948	0.7331	0.7	0.6386	0.9882	0.0494	0.0864	382.0696	9576	6753754.7683	232310.3603
5	711	Гібридна		0.6926	0.6926	0.7382	0.7043	0.6201	0.9773	0.0643	0.0486	273.8713	9576	8863553.6739	294339.2677
6	366	Гібридна		0.6923	0.6923	0.7138	0.7499	0.606	0.9539	0.0445	0.1058	518.7792	9576	6678231.028	251239.1121
7	272	Гібридна		0.6921	0.6921	0.7441	0.6819	0.6328	0.9516	0.033	0.051	545.0934	9576	8974605.4796	217421.6174
8	857	Гібридна		0.6895	0.6895	0.7397	0.6753	0.6368	0.9904	0.0485	0.071	353.6565	9576	9785725.3369	191547.4548
9	569	Гібридна		0.6878	0.6878	0.7209	0.7185	0.613	0.9586	0.096	0.0403	357.231	9576	9589632.6847	280190.2603
10	803	Гібридна		0.6878	0.6878	0.7345	0.6832	0.63	0.994	0.076	0.057	267.9196	9576	6840625.7107	305944.4295

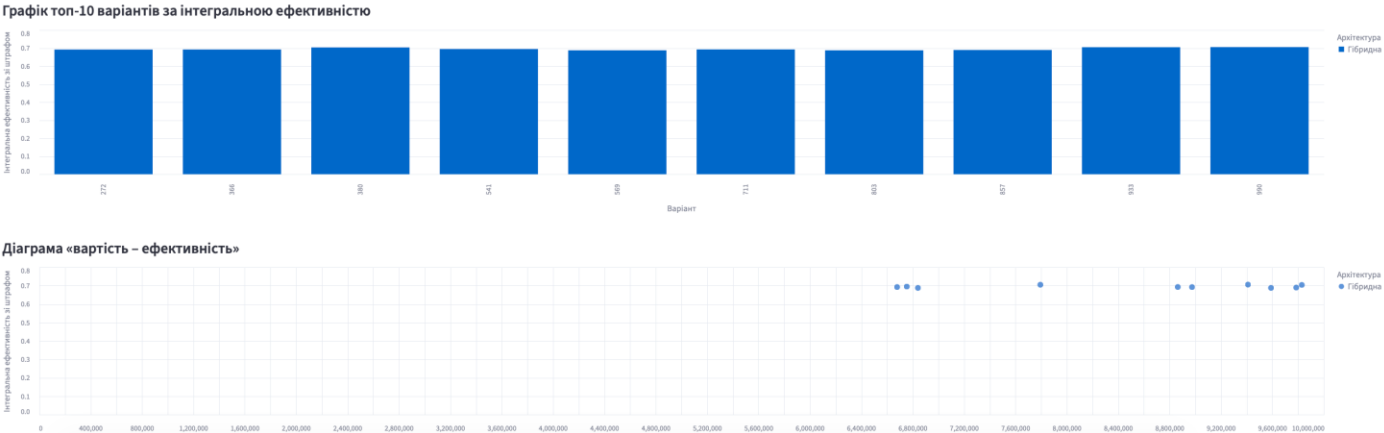


Рис.4.9. Результати оптимізації і вибір оптимуму

Остання, п’ята вкладка, відображає результати порівняння ефективності базової конфігурації та оптимізованої, яка знайдена за розробленим алгоритмом пошуку оптимуму. Результати роботи відображені на рис. 4.10.

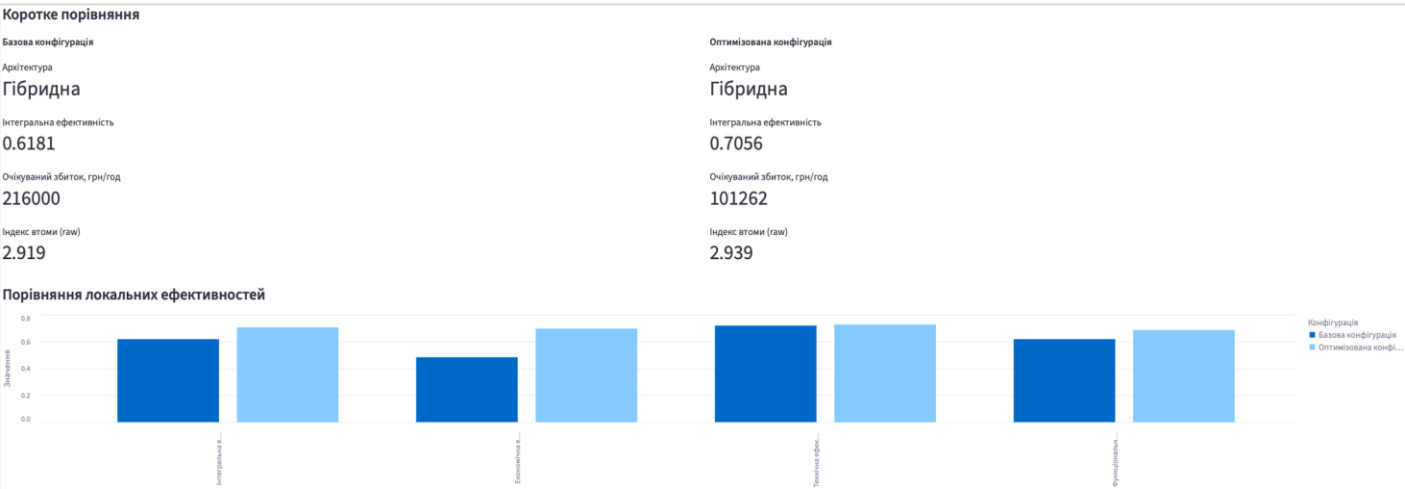


Рис. 4.10. Результати порівняння базової конфігурації та оптимальної конфігурації

За результатами проведеного дослідження маємо наступні показники, які відображають ефективність систем, що представлені в діаграмі на рис. 4.11.

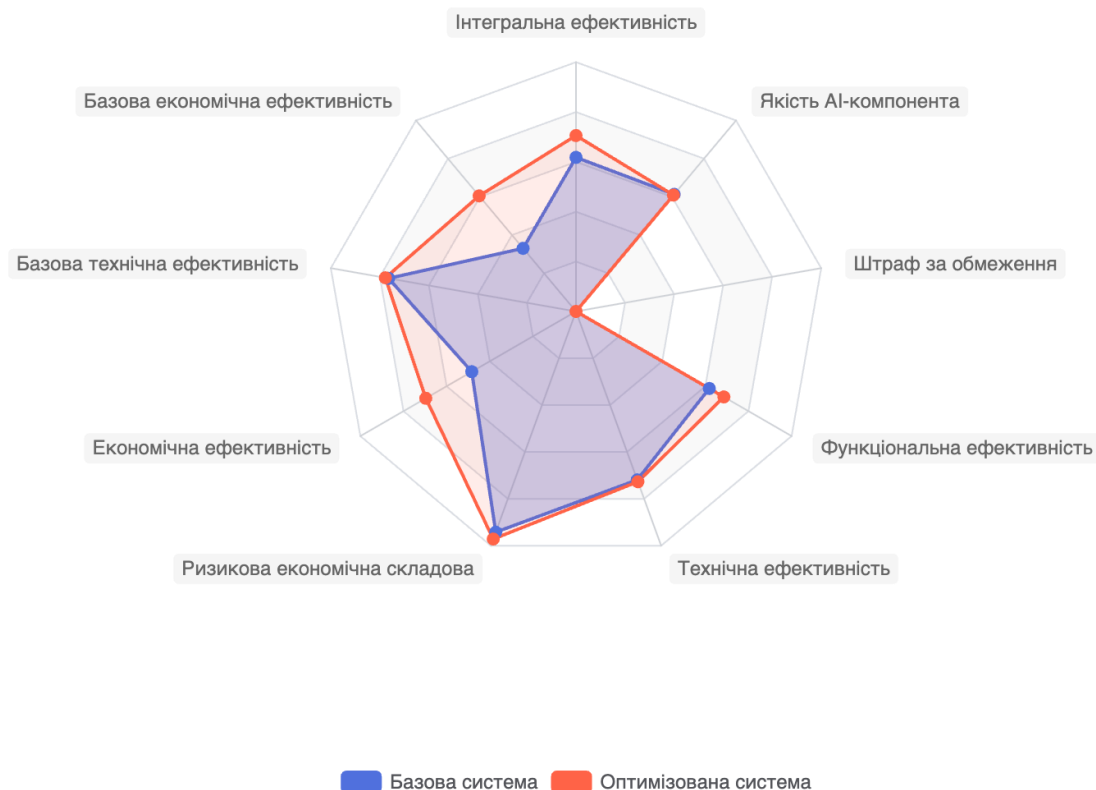


Рис. 4.11. Радарна діаграма порівняння локальних ефективностей конфігурацій

Отримана конфігурація, є значно ефективнішою за базову розглянуту в допустимому заданому діапазоні користувачем, що свідчить про придатність роботи та застосування методу в умовах багатокритеріального аналізу. Було визначено, що для такої конфігурації існує більш оптимальна та раціональна конфігурація, якщо здатна зменшити навантаження на низку критеріїв тим самим збільшити ефективність системи в цілому.

Відповідні порівняння представлені у вигляді діаграм зі значеннями розрахованих показників на рис. 4.12, 4.13.

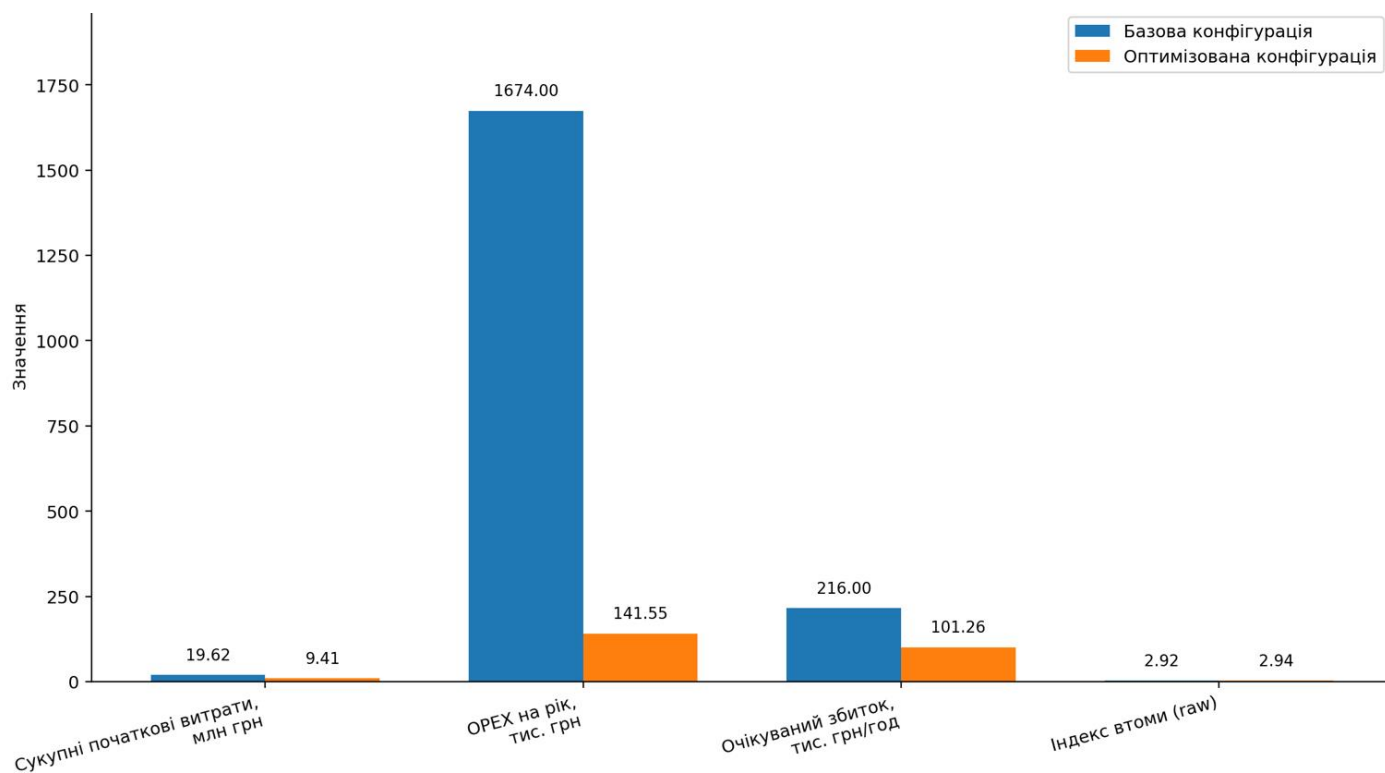


Рис. 4.12. Порівняння ключових економічних та експлуатаційних показників

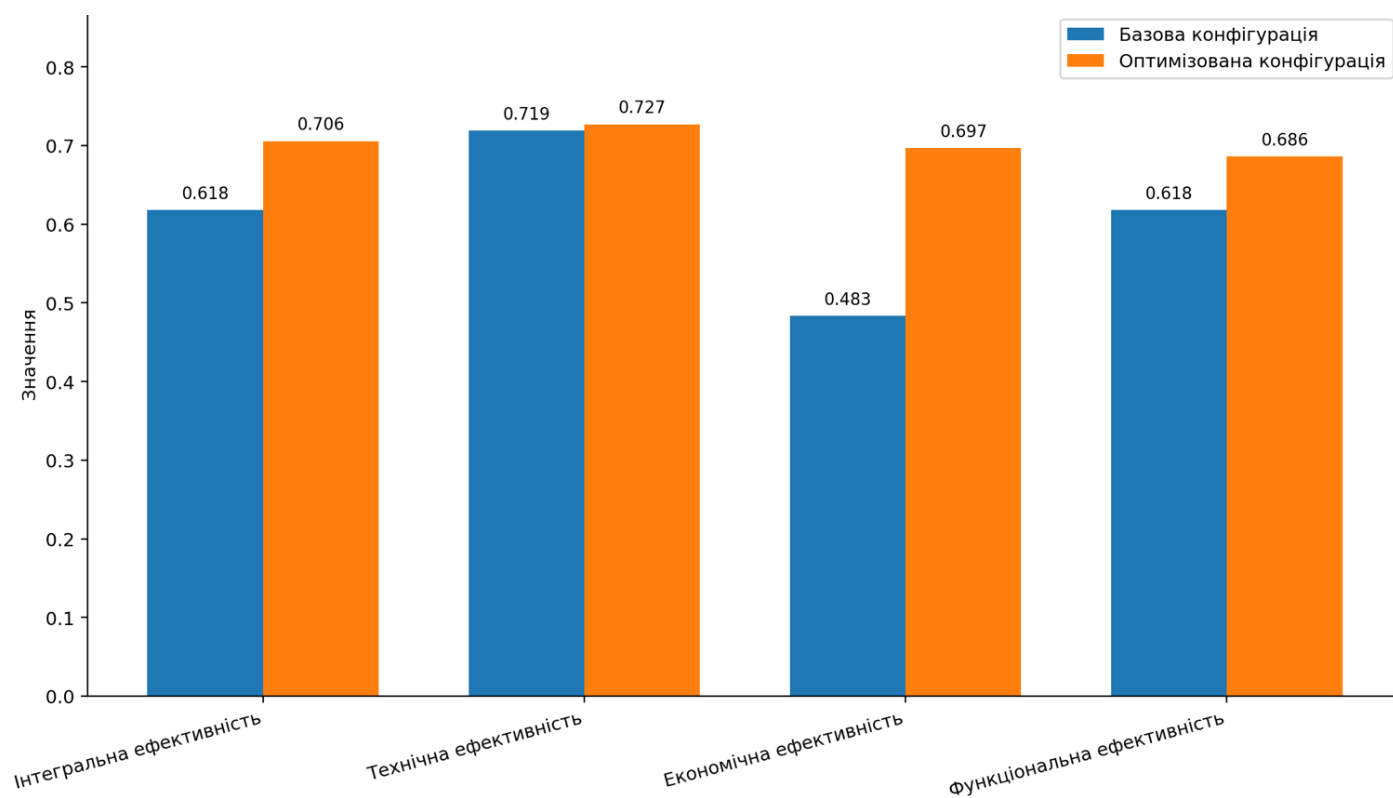


Рис. 4.13. Порівняння локальних та інтегральної ефективності

З отриманих результатів можемо оцінити, що найбільшій оптимізації була зазнана економічна ефективність. На представленому графіку рис.4.13. ми бачимо, що з від'ємно позитивних значень оптимізації є лише індекс втоми оператора, який зумовлений вхідними показниками і допустимий для даної конфігурації, по відношенню до базової конфігурації та відрізняється на показник в 0,02. Щодо інших критеріїв то вони показують значне підвищення потенційної ефективності для:

- Сукупних початкових витрат з 19,62 до 9.41, що на 52% оптимальніше;
- Операційні витрати на систему на рік з 1674.00 до 141.55, що на 91,5% оптимальніше;
- Очікуваний збиток від інцидентів за годину часу з 216.00 до 101.26 , що на 53,1% оптимальніше.

4.7. Порівняльний аналіз існуючих підходів і розробленого методу

Оскільки в попередньому підрозділі було доведено ефективність розробленого методу в рамках самого методу є необхідність в доведенні його ефективності по відношенню до інших існуючих методів, які були розглянути в першому розділі даного дисертаційного дослідження.

Серед визначених методів участь у порівнянні прийматимуть підходи технічно-інженерні, економічний та експертно-функціональний. Метод оцінки ризиків не буде розглядатись для порівняння з розробленим методом, оскільки даний метод не містить необхідних значень для порівняння, адже складається в більшості з описового характеру потенційних ризиків з використанням матриці ймовірностей і наслідків.

Дане дослідження з метою порівняння буде використовувати аналогічну конфігурацію, яка раніше призначена для оптимізації і її базові параметри, дозволять нам прозоро визначити ефективність кожного методу. Для реалізації таких розрахунків слугуватимуть вагові коефіцієнти, зміна яких дозволить правильно оцінити параметри ефективності та порівняти їх.

Варто також зазначити, перед початком дослідження, що дані методи в своїй природі не включають в себе розрахунок впливу якості та точності аналітичних інструментів виявлення та розпізнавання, через що, дані застарілі методи вже є недостатньо ефективними і таке порівняння буде з самого початку неповним для методів і від'ємно корисно ефективним в порівнянні, а саме тому ми будемо враховувати цей параметр, для правильності порівняння. Даний параметр A_i складає 0.939.

Також для даних методів відсутнє прогнозування та вплив на інші складові системи, що є їх основним недоліком і не дозволяє провести повноцінне порівняння, а тому вони будуть визначатись розробленим методом оцінки ефективності.

Для таких методів додаємо наступні вагові параметри після чого розраховуємо та фіксуємо значення ефективності:

- за технічними характеристиками: 0.8/0.1/ 0.1 для $w_{tech}, w_{econ}, w_{ops}$ відповідно;
- за економічними характеристиками: 0.1/0.8/0.1 для $w_{tech}, w_{econ}, w_{ops}$ відповідно;
- за експертно-функціональними можливостями: 0.1/0.1/0.8 для $w_{tech}, w_{econ}, w_{ops}$ відповідно.

Отримані результати розрахунку показують, що такий перерозподіл не впливає на обраховані локальні критерії ефективності, але значно впливають на інтегральну ефективність системи, що представлено на рис.4.14. Таким чином, з точки логіки самого методу, поставлена задача виконана успішно та має відповідний позитивний результат, але відсутність розрахунку впливу на інші групи ефективностей відсутній, від чого інтегральний показник зменшується.

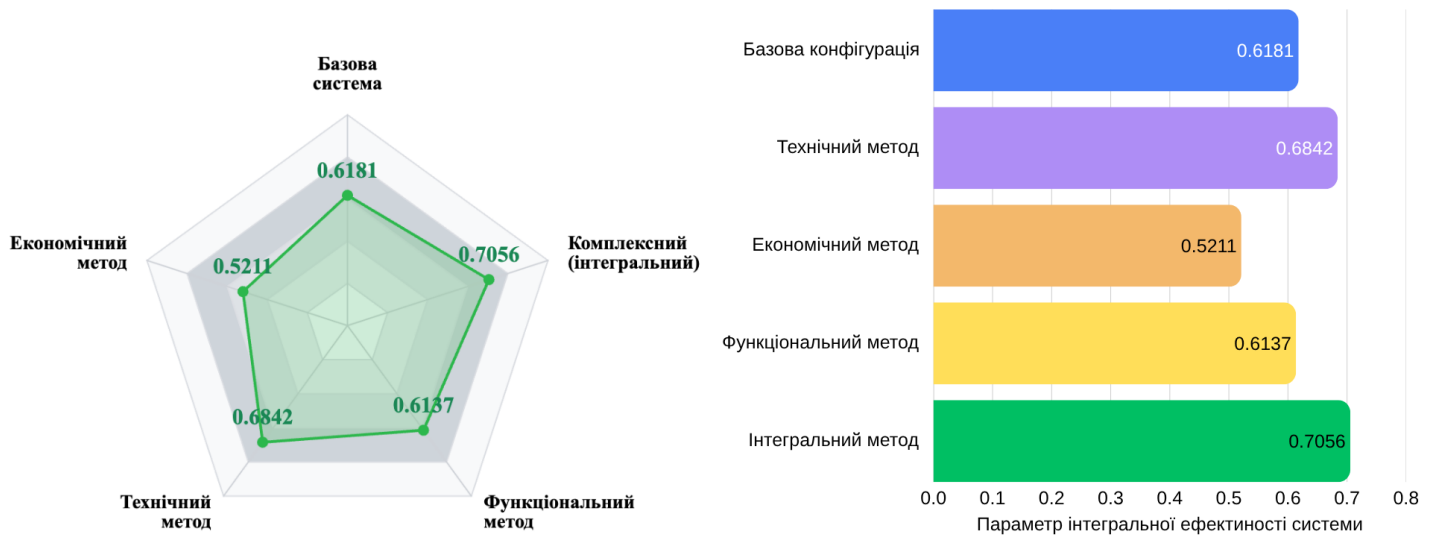


Рис. 4.14. Порівняння значень ефективності для існуючих методів та розробленого інтегрального методу

Для таких існуючих підходів визначається наступна логіка впливу на комплексну систему:

1. Технічний метод – покращує ряд технічних критеріїв і дозволяє обрати найточнішу та найпродуктивнішу систему, але абсолютно не враховує фактор економічних витрат та прогнозування.

2. Економічний метод – дозволяє скоротити витрати, що прямопропорційно впливає на якість обладнання та на його можливості, що потім вплине зворотно на сам економічний фактор, через технічний (часто виходить з ладу обладнання, потреба проводити частіше технічне обслуговування, складності в модернізації) та функціональний (збільшення кількості інцидентів, пропуски інцидентів).

3. Експертно-функціональний метод є найбільш небезпечним для проектування систем, оскільки не дозволяє коректно оцінити необхідність правильного програмно-апаратного комплексу та економічного впливу на систему.

Тому, якщо оцінювати існуючі методи без моделювання залежностей та впливу на інші групи ефективностей, то таке математичне прогнозування виглядало би відповідно до графіку представленого на рис. 4.15. Таке візуальне представлення

результатів дозволяє наочно оцінити недолік існуючих методів та зрозуміти їх непридатність до застосування для сучасних комплексних інтегрованих систем відеоспостереження з елементами аналізу даних в порівнянні з багатокритеріальним комплексним методом оцінки інтегральної ефективності.

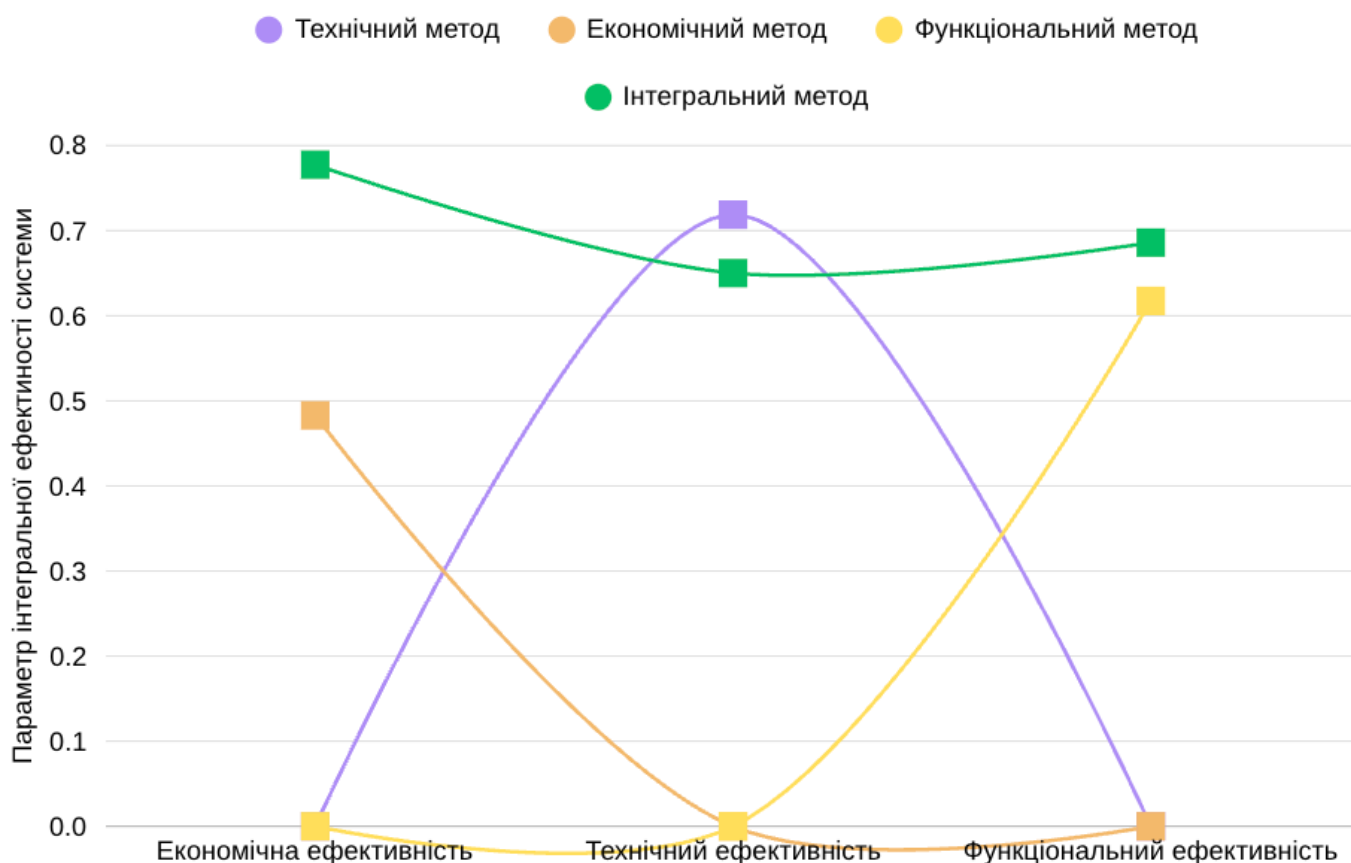


Рис. 4.15. Фактичне представлення результатів для методів без врахування прогнозування розробленим методом оцінки ефективності

Для того, аби визначити відсоткові значення покращення ефективності розраховуємо їх з використанням визначених параметрів ефективності. Використання технічного методу має значення 0.6842 та інтегрального 0.7056, що означає більшу ефективність для останнього на 3,13%. Для економічного підходу де значення ефективності складає 0.5211, що показує збільшення ефективності розробленого методу

досягає 35,41%, а для експертно-функціонально методу зі значенням 0.6137 такий приріст має показник в 14,97%.

Таким чином можемо підсумувати, що розроблений багатокритеріальний комплексний метод оцінки інтегральної ефективності має більші кількісні показники, що дозволяють більш гнучко та ефективніше проводити оцінку системи, передбачити її комплексну та локальну ефективність кожного критерію, а також виконувати оптимізацію для конфігурації, яка буде ефективніше за наявні існуючі методи до 35%.

В свою чергу якісні показники також демонструють повноту даного методу, що включає в себе багатокритеріальний підхід, вплив аналітичних детекторів та ефективність системи та локальні групи ефективностей. Розроблений прототип застосунку калькулятора дозволяє швидко та зручно застосовувати даний метод для оцінки систем та отримувати зручні для сприйняття результати для аналізу та прийняття раціонального, зваженого та аргументованого вибору.

Висновки до розділу 4

В даному розділі було проведено дослідження, метою якого було перевірка роботи методу, вірність розрахунків для даного методу та можливість інтерпретації результатів. Оцінка конфігурації обиралась з огляду на реальні запити побудови ІСВ для об'єктів в Україні.

Для перевірки роботи методу було виконано ручний демонстраційний розрахунок необхідних величин та визначено локальні критерії ефективності базової системи. За допомогою програмного прототипу було проведено аналогічні розрахунки і їх порівняння показало, що результати збігаються з допустимою похибкою. Зазначено, що даний метод повторюваний і може бути відтворений різними підходами до розрахунку, а алгоритм є послідовний та логічний.

Проведено процес оптимізації, за яким було згенеровано 1000 конфігурацій та обрано найбільш оптимальну за параметрами. Дані репрезентовано у відповідних таблицях та графіках і порівняно з основною базовою конфігурацією. Для заданого діапазону було виявлено, що оптимально конфігурація по всіх параметрах є кращою за виключенням параметру навантаження на оператора, що є допустимим, оскільки значення діапазону даного коефіцієнту є допустимим.

Таким чином можна стверджувати, що метод дозволяє покращити задану конфігурацію, а тому має практичне застосування.

За допомогою розробленого методу оцінки ефективності було перевірено та розраховано критерії ефективності для існуючих методів і визначено, що даний метод дозволяє отримати кращі кількісні та якісні показники. Кількісно такі показники для теоретичних розрахунків лежить в діапазоні від 3% до 35% по відношенню до значень інших існуючих методів. Розрахунок виконувався через зміну вагових коефіцієнтів, що дозволило провести розрахунок інтегральної ефективності.

ВИСНОВОК

У дисертаційній роботі розглянуто та вперше вирішено актуальне науково-прикладне завдання, яке поляє у розробці методу оцінювання сучасних систем відеоспостереження з елементами аналізу даних та вибору найбільш раціональної конфігурації, що враховує економічні, технічні та функціональні характеристики. Результати, отримані в ході роботи, дозволяють сформулювати наступні висновки:

1. Проведений аналіз сучасних підходів до проектування, побудови та оцінки інтегрованих систем відеоспостереження показав, що поточні існуючі методи орієнтовані на розрахунок показників окремих технічних параметрів, економічних параметрів витрат, експертність вибору проектувальника або його зацікавленість в конкретній конфігурації. Враховуючи такий підхід визначено, що вони не встановлюють зв'язок між різними параметрами як технічними, економічними та функціональними, наслідком чого є лише часткова та не повна оцінка конфігурації системи. Крім того дані підходи не враховують вплив аналітичних інструментів на вказані вище критерії та не встановлюють зв'язок залежностей між ними. Установлено, що для різних за масштабом об'єктів, особливо з високим рівнем критичності така фрагментована оцінка не дозволяє обґрунтовано та зважено порівняти конфігурації та зробити вибір.

2. Вперше сформовано та описано периметричний опис інтегрованої системи відеоспостереження, у межах якого виділено технічний, економічний та функціональний вектори параметрів, а також визначено систему зовнішніх обмежень і критеріїв допустимості. Такий підхід дозволяє перейти від загального описового рівня представлення конфігурації системи до її формалізованого вигляду який має форму подання впорядкованих математичних множин параметрів, що придатні для моделювання, аналізу та застосування ітераційного оптимізаційного пошуку.

3. Вперше розроблено математичну модель інтегральної ефективності інтегрованої системи відеоспостереження, яка поєднує технічну, економічну та

функціональну складову проекту в рамках одного критеріального простору. На відміну від розглянутих підходів, така модель враховує ряд прямих характеристик конфігурації, а також похідні показники, які залежать від якості аналітичних детекторів, очікуваними витратами від пропуску інциденту, витратами на обробку хибних детекцій та навантаженням на оператора. Дана модель забезпечує більш повне та структуроване відображення реальних умов якості роботи системи.

4. Удосконалено підхід, який дозволяє врахувати вплив аналітичних модулів на інтегральну ефективність системи. Показано, що похибка, яка спричиняється роботою аналітики, мають не локальний вплив на критерій ефективності одної групи, але й на інші групи. У результаті було встановлено, що показник відеоаналітики має бути оцінений як ключовий фактор для таких систем, як фактор загальної ефективності, а не ізольований показник.

5. Вперше запропоновано метод багатокритеріального вибору конфігурації системи, який включає етапи формування вхідних даних, перевірки допустимості рішення, розрахунок локальних показників ефективності, інтегральне оцінювання та ранжування альтернативних пропозицій. Даний метод не лише дозволяє оцінити окрему конфігурації, а й дозволяє знайти альтернативну в межах заданого сценарію та обмежень. Такі результати дозволяють наочно, кількісно та формалізовано підійти до процедури прийняття обґрунтованого рішення.

6. Виконано теоретичне обґрунтування властивостей розробленого методу. Показано та доведено, що зазначений метод є універсальним для використання з різними типами архітектури, адаптивним до різних умов застосування, придатний до масштабування та чутливий до змін ключових параметрів моделі. Встановлено, що запропонований підхід дозволяє працювати з різними класами об'єктів, що дозволяє змінювати параметричне наповнення та систему обмежень без зміни вхідних даних конфігурації.

7. Вперше реалізовано програмний прототип методу з використанням мови програмування Python та бібліотекою Streamlit, який реалізований для введення вхідних

параметрів системи, обчислення локальних та інтегральних значень ефективності, перевірку конфігурації на відповідність до граничних обмежень, а також виконання процесу оптимізації і пошуку найоптимальніших конфігурацій відповідно до заданої системи. Також розроблено інтерфейс для зручного порівняння значень базової системи та оптимізованої через основні показники, графіки, а також повну таблицю параметрів.

8. В процесі експериментальної перевірки якісно підтверджено коректність демонстраційного (ручного) розрахунку ефективності базової конфігурації використовуючи засоби розробленого програмного прототипу. Встановлено, що показники збігаються для логіки ручного розрахунку та програмного, що тим самим вказує на узгодженість математичної моделі та її коректна реалізація. Таким чином даний метод має характеристику відтворюваності результатів, а також придатність алгоритму для активної експлуатації.

9. За результатами пошуку оптимуму було встановлено, що використання даного методу оцінки ефективності та оптимізації дозволяє виконати перехід від базової конфігурації системи, яка була запропонована на вході, до оптимізованої конфігурації на виході даного методу. Представлено, що оптимізація спрямована не суто на поліпшенні інтегрального методу оптимізації, а й на локальні змінні ефективності (технічну, економічну, функціональну).

10. Проведений аналіз та порівняльний підхід існуючих методів для оцінки конфігурації системи до розробленого методу показує, що спрощені методи або не враховують частину суттєвих факторів системи, або ж не забезпечують глибини аналізу, через що відсутня точна та раціональна оцінка ефективності розглянутої конфігурації системи до впровадження. Використання запропонованого методу вирішує дану проблематику і має якісну перевагу через охоплення груп різних аспектів системи, а також кількісну за рахунок формування раціональних та ефективніших альтернатив, значення яких, по ефективності, може бути більше від 3% до 35% в порівнянні з існуючими методами.

11. Практичне значення розробленого методу полягає в його доцільності до використання для вибору або обґрунтуванню конфігурації для ІСВ різного масштабу та вимог.

Отже, мету дисертаційної роботи та дослідження було досягнуто і отримано якісні та кількісні результати, визначені поставлені задачі були в повній мірі виконані. Метод оцінки ефективності та оптимізації інтегрованих систем відеоспостереження з елементами аналізу даних є теоретично обґрунтованим, експериментально перевірений, реалізований програмно, що робить його придатним до практичного застосування.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Axxon One Video Management Software. AxxonSoft. URL: <https://www.axxonsoft.com/products/video-management-software> (дата звернення: 26.03.2026).
2. Axxon One Technical Specifications. AxxonSoft. URL: <https://www.axxonsoft.com/products/video-management-software/axxon-one-specifications> (дата звернення: 26.03.2026).
3. Editions of Axxon One Video Management Software. AxxonSoft. URL: <https://www.axxonsoft.com/products/video-management-software/editions> (дата звернення: 26.03.2026).
4. Axxon One 2.0.14 Release Notes. AxxonSoft Documentation. URL: <https://docs.axxonsoft.com/confluence/spaces/one20en/pages/378503224/Axxon+One+2.0.14+Release+Notes> (дата звернення: 26.03.2026).
5. Axxon One 2.0.10 Release Notes. AxxonSoft Documentation. URL: <https://docs.axxonsoft.com/confluence/spaces/one20en/pages/335972778/Axxon+One+2.0.10+Release+Notes> (дата звернення: 26.03.2026).
6. Axxon One 2.0.2 Release Notes. AxxonSoft Documentation. URL: <https://docs.axxonsoft.com/confluence/spaces/one20en/pages/281549812/Axxon+One+2.0.2+Release+Notes> (дата звернення: 26.03.2026).
7. Axxon One Release Notes. AxxonSoft Documentation. URL: <https://docs.axxonsoft.com/confluence/spaces/one20en/pages/314520285/Axxon+One+Release+Notes> (дата звернення: 26.03.2026).
8. DetectorPack Release Notes. AxxonSoft Documentation. URL: <https://docs.axxonsoft.com/confluence/spaces/one20en/pages/314520282/DetectorPack+Release+Notes> (дата звернення: 26.03.2026).

9. DetectorPack 3.15.2 Release Notes. AxxonSoft Documentation. URL: <https://docs.axxonsoft.com/confluence/spaces/one20en/pages/386629840/DetectorPack+3.15.2+Release+Notes> (дата звернення: 26.03.2026).
10. Smart Event Search (MomentQuest). AxxonSoft. URL: <https://www.axxonsoft.com/ai-video-analytics/ai-search/smart-event-search> (дата звернення: 26.03.2026).
11. Video Footage Management. AxxonSoft. URL: <https://www.axxonsoft.com/products/video-management-software/key-features/video-footage-management> (дата звернення: 26.03.2026).
12. Axxon PSIM Software. AxxonSoft. URL: <https://www.axxonsoft.com/products/psim> (дата звернення: 26.03.2026).
13. Axxon PSIM Specifications. AxxonSoft. URL: <https://www.axxonsoft.com/products/psim/psim-specifications> (дата звернення: 26.03.2026).
14. Axxon PSIM Key Features. AxxonSoft. URL: <https://www.axxonsoft.com/products/psim/psim-key-features> (дата звернення: 26.03.2026).
15. Axxon One Setup Checklist. AxxonSoft. URL: <https://www.axxonsoft.com/support/axxon-one-setup-checklist> (дата звернення: 26.03.2026).
16. Axxon One SDK. AxxonSoft. URL: <https://www.axxonsoft.com/support/sdk-axxon-one> (дата звернення: 26.03.2026).
17. Hardware Calculator. AxxonSoft. URL: <https://www.axxonsoft.com/support/support-hardware-calculator> (дата звернення: 26.03.2026).
18. Documentation. AxxonSoft. URL: <https://www.axxonsoft.com/support/documentations> (дата звернення: 26.03.2026).
19. ONVIF Core Specification. ONVIF. URL: <https://www.onvif.org/specs/core/ONVIF-Core-Specification.pdf> (дата звернення: 26.03.2026).

20. Network Interface Specifications. ONVIF. URL: <https://www.onvif.org/profiles/specifications/> (дата звернення: 26.03.2026).
21. ONVIF Profile S Specification. ONVIF. URL: <https://www.onvif.org/profiles/profile-s/> (дата звернення: 26.03.2026).
22. ONVIF Profile T Specification. ONVIF. URL: <https://www.onvif.org/profiles/profile-t/> (дата звернення: 26.03.2026).
23. IEC 62676-1-1:2013. Video surveillance systems for use in security applications – Part 1-1: System requirements – General. IEC Webstore. URL: <https://webstore.iec.ch/en/publication/7347> (дата звернення: 26.03.2026).
24. IEC 62676-1-2:2013. Video surveillance systems for use in security applications – Part 1-2: Performance requirements for video transmission. IEC.
25. IEC 62676-2-1:2013. Video surveillance systems for use in security applications – Part 2-1: Video transmission protocols – General requirements. IEC.
26. IEC 62676-2-11:2024. Video surveillance systems for use in security applications – Part 2-11: VMS and cloud VSaaS profiles. IEC Webstore. URL: <https://webstore.iec.ch/en/publication/66755> (дата звернення: 26.03.2026).
27. IEC 62676-2-31:2019. Video surveillance systems for use in security applications – Part 2-31: Live streaming and control based on web services. IEC Webstore. URL: <https://webstore.iec.ch/en/publication/61227> (дата звернення: 26.03.2026).
28. IEC 62676-2-32:2019. Video surveillance systems for use in security applications – Part 2-32: Recording control and replay based on web services. IEC Webstore. URL: <https://webstore.iec.ch/en/publication/61241> (дата звернення: 26.03.2026).
29. IEC 62676-2-33:2022. Video surveillance systems for use in security applications – Part 2-33: Management systems interfaces. IEC Webstore. URL: <https://webstore.iec.ch/en/publication/64698> (дата звернення: 26.03.2026).
30. IEC 62676-3:2013. Video surveillance systems for use in security applications – Part 3: Analog and digital video interfaces. IEC.

31. IEC 62676-4:2025. Video surveillance systems for use in security applications – Part 4: Application guidelines. IEC Webstore. URL: <https://webstore.iec.ch/en/publication/83425> (дата звернення: 26.03.2026).
32. IEC 62676-5:2018. Video surveillance systems for use in security applications – Part 5: Data specifications and image quality performance for camera devices. IEC.
33. IEC 62676-6:2026. Video surveillance systems for use in security applications – Part 6: Intelligent video analysis. IEC Webstore. URL: <https://webstore.iec.ch/en/publication/59704> (дата звернення: 26.03.2026).
34. ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. ISO. URL: <https://www.iso.org/standard/27001> (дата звернення: 26.03.2026).
35. ISO 31000:2018. Risk management — Guidelines. ISO. URL: <https://www.iso.org/standard/65694.html> (дата звернення: 26.03.2026).
36. Tabassi E., Allen C., Burke K. et al. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. Gaithersburg: National Institute of Standards and Technology, 2023. URL: <https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10> (дата звернення: 26.03.2026).
37. Intel Corporation. OpenVINO™ Documentation. URL: <https://docs.openvino.ai/> (дата звернення: 26.03.2026).
38. NVIDIA Corporation. NVIDIA DeepStream SDK Documentation. URL: <https://docs.nvidia.com/metropolis/deepstream/> (дата звернення: 26.03.2026).
39. Sokolova M., Lapalme G. A systematic analysis of performance measures for classification tasks // Information Processing & Management. 2009. Vol. 45, No. 4. P. 427–437.
40. Davis J., Goadrich M. The Relationship Between Precision-Recall and ROC Curves // Proceedings of the 23rd International Conference on Machine Learning. 2006. P. 233–240.

41. Powers D. M. W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation // Journal of Machine Learning Technologies. 2011. Vol. 2, No. 1. P. 37–63.
42. Fawcett T. An introduction to ROC analysis // Pattern Recognition Letters. 2006. Vol. 27, No. 8. P. 861–874.
43. Bernardin K., Stiefelhagen R. Evaluating multiple object tracking performance: the CLEAR MOT metrics // EURASIP Journal on Image and Video Processing. 2008. Vol. 2008. Article ID 246309.
44. Leal-Taixé L., Milan A., Reid I. et al. MOTChallenge 2015: Towards a benchmark for multi-target tracking // arXiv:1504.01942. 2015.
45. Dendorfer P., Osep A., Milan A. et al. MOTChallenge: A benchmark for single-camera multiple target tracking // International Journal of Computer Vision. 2021. Vol. 129. P. 845–881.
46. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2001. Vol. 1. P. I-511–I-518.
47. Dalal N., Triggs B. Histograms of oriented gradients for human detection // Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005. Vol. 1. P. 886–893.
48. Felzenszwalb P. F., Girshick R. B., McAllester D., Ramanan D. Object detection with discriminatively trained part-based models // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. Vol. 32, No. 9. P. 1627–1645.
49. Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014. P. 580–587.
50. Girshick R. Fast R-CNN // Proceedings of the IEEE International Conference on Computer Vision. 2015. P. 1440–1448.

51. Ren S., He K., Girshick R., Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. Vol. 39, No. 6. P. 1137–1149.
52. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, real-time object detection // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. P. 779–788.
53. Redmon J., Farhadi A. YOLO9000: Better, faster, stronger // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. P. 6517–6525.
54. Liu W., Anguelov D., Erhan D. et al. SSD: Single Shot MultiBox Detector // Computer Vision – ECCV 2016. Lecture Notes in Computer Science. Vol. 9905. Cham: Springer, 2016. P. 21–37.
55. Lin T.-Y., Dollár P., Girshick R. et al. Feature Pyramid Networks for object detection // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. P. 2117–2125.
56. Lin T.-Y., Goyal P., Girshick R., He K., Dollár P. Focal loss for dense object detection // Proceedings of the IEEE International Conference on Computer Vision. 2017. P. 2980–2988.
57. He K., Gkioxari G., Dollár P., Girshick R. Mask R-CNN // Proceedings of the IEEE International Conference on Computer Vision. 2017. P. 2961–2969.
58. Tan M., Pang R., Le Q. V. EfficientDet: Scalable and efficient object detection // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. P. 10781–10790.
59. Carion N., Massa F., Synnaeve G. et al. End-to-End Object Detection with Transformers // Computer Vision – ECCV 2020. Lecture Notes in Computer Science. Vol. 12346. Cham: Springer, 2020. P. 213–229.
60. Bochkovskiy A., Wang C.-Y., Liao H.-Y. M. YOLOv4: Optimal speed and accuracy of object detection // arXiv:2004.10934. 2020.

61. Ge Z., Liu S., Wang F. et al. YOLOX: Exceeding YOLO series in 2021 // arXiv:2107.08430. 2021.
62. Wang C.-Y., Bochkovskiy A., Liao H.-Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors // arXiv:2207.02696. 2022.
63. Bewley A., Ge Z., Ott L. et al. Simple online and realtime tracking // 2016 IEEE International Conference on Image Processing. 2016. P. 3464–3468.
64. Wojke N., Bewley A., Paulus D. Simple online and realtime tracking with a deep association metric // 2017 IEEE International Conference on Image Processing. 2017. P. 3645–3649.
65. Zhang Y., Sun P., Jiang Y. et al. ByteTrack: Multi-object tracking by associating every detection box // Computer Vision – ECCV 2022. Lecture Notes in Computer Science. Vol. 13682. Cham: Springer, 2022. P. 1–21.
66. Hart P. E., Nilsson N. J., Raphael B. A formal basis for the heuristic determination of minimum cost paths // IEEE Transactions on Systems Science and Cybernetics. 1968. Vol. 4, No. 2. P. 100–107.
67. Saaty T. L. The Analytic Hierarchy Process. New York: McGraw-Hill, 1980. 287 p.
68. Saaty T. L. Decision making with the analytic hierarchy process // International Journal of Services Sciences. 2008. Vol. 1, No. 1. P. 83–98.
69. Hwang C. L., Yoon K. Multiple Attribute Decision Making: Methods and Applications. Berlin: Springer, 1981. 269 p.
70. Triantaphyllou E. Multi-Criteria Decision Making Methods: A Comparative Study. Dordrecht: Kluwer Academic Publishers, 2000. 320 p.
71. Belton V., Stewart T. Multiple Criteria Decision Analysis: An Integrated Approach. Boston: Kluwer Academic Publishers, 2002. 372 p.
72. Keeney R. L., Raiffa H. Decisions with Multiple Objectives: Preferences and Value Tradeoffs. New York: John Wiley & Sons, 1976. 569 p.
73. Roy B. Multicriteria Methodology for Decision Aiding. Dordrecht: Kluwer Academic Publishers, 1996. 292 p.

74. Zadeh L. A. Fuzzy sets // Information and Control. 1965. Vol. 8, No. 3. P. 338–353.
75. Zimmermann H.-J. Fuzzy Set Theory and Its Applications. 4th ed. Boston: Kluwer Academic Publishers, 2001. 514 p.
76. Chen S.-J., Hwang C.-L. Fuzzy Multiple Attribute Decision Making: Methods and Applications. Berlin: Springer, 1992. 536 p.
77. Deb K. Multi-Objective Optimization Using Evolutionary Algorithms. Chichester: John Wiley & Sons, 2001. 497 p.
78. Boyd S., Vandenberghe L. Convex Optimization. Cambridge: Cambridge University Press, 2004. 716 p.
79. Nocedal J., Wright S. J. Numerical Optimization. 2nd ed. New York: Springer, 2006. 664 p.
80. Dantzig G. B. Linear Programming and Extensions. Princeton: Princeton University Press, 1963. 625 p.
81. Bertsekas D. P. Nonlinear Programming. 2nd ed. Belmont: Athena Scientific, 1999. 777 p.
82. Hillier F. S., Lieberman G. J. Introduction to Operations Research. 10th ed. New York: McGraw-Hill, 2015. 1008 p.
83. Taha H. A. Operations Research: An Introduction. 10th ed. Harlow: Pearson, 2017. 848 p.
84. Winston W. L. Operations Research: Applications and Algorithms. 4th ed. Belmont: Thomson Brooks/Cole, 2004. 1418 p.
85. Fishburn P. C. Utility Theory for Decision Making. New York: John Wiley & Sons, 1970. 234 p.
86. Elsayed E. A. Reliability Engineering. 2nd ed. Hoboken: John Wiley & Sons, 2012. 688 p.
87. Modarres M., Kaminskiy M., Krivtsov V. Reliability Engineering and Risk Analysis: A Practical Guide. 3rd ed. Boca Raton: CRC Press, 2016. 536 p.

88. O'Connor P. D. T., Kleyner A. Practical Reliability Engineering. 5th ed. Chichester: John Wiley & Sons, 2012. 512 p.
89. Kaplan S., Garrick B. J. On the quantitative definition of risk // Risk Analysis. 1981. Vol. 1, No. 1. P. 11–27.
90. IEC 61508-1:2010. Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements. IEC.
91. Hart S. G., Staveland L. E. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research // Advances in Psychology. 1988. Vol. 52. P. 139–183.
92. Wickens C. D., Lee J. D., Liu Y., Gordon-Becker S. E. An Introduction to Human Factors Engineering. 2nd ed. Boston: Pearson, 2013. 624 p.
93. Parasuraman R., Sheridan T. B., Wickens C. D. A model for types and levels of human interaction with automation // IEEE Transactions on Systems, Man, and Cybernetics – Part A. 2000. Vol. 30, No. 3. P. 286–297.
94. Endsley M. R. Toward a theory of situation awareness in dynamic systems // Human Factors. 1995. Vol. 37, No. 1. P. 32–64.
95. Reason J. Human Error. Cambridge: Cambridge University Press, 1990. 302 p.
96. Stanton N., Hedge A., Brookhuis K., Salas E., Hendrick H. Handbook of Human Factors and Ergonomics Methods. Boca Raton: CRC Press, 2005. 530 p.
97. ISO 9241-210:2019. Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. ISO.
98. Hu W., Tan T., Wang L., Maybank S. A survey on visual surveillance of object motion and behaviors // IEEE Transactions on Systems, Man, and Cybernetics, Part C. 2004. Vol. 34, No. 3. P. 334–352.
99. Valera M., Velastin S. A. Intelligent distributed surveillance systems: a review // IEE Proceedings – Vision, Image and Signal Processing. 2005. Vol. 152, No. 2. P. 192–204.
100. Remagnino P., Foresti G. L., Ellis T. Video-Based Surveillance Systems: Computer Vision and Distributed Processing. Boston: Kluwer Academic Publishers, 2002. 262 p.

101. Collins R., Lipton A., Kanade T. et al. A system for video surveillance and monitoring. VSAM Final Report. Pittsburgh: Carnegie Mellon University, 2000. 72 p.
102. Rinner B., Wolf W., Javed O., Ribeiro A. Distributed smart cameras for vision-based surveillance applications // Proceedings of the IEEE. 2008. Vol. 96, No. 10. P. 1565–1575.
103. Hampapur A., Brown L., Connell J. et al. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking // IEEE Signal Processing Magazine. 2005. Vol. 22, No. 2. P. 38–51.
104. Basharat A., Gritai A., Shah M. Learning object motion patterns for anomaly detection and improved object detection // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2008. P. 1–8.
105. Truong B. T., Venkatesh S. Video abstraction: A systematic review and classification // ACM Transactions on Multimedia Computing, Communications, and Applications. 2007. Vol. 3, No. 1. Article 3.
106. Shi W., Cao J., Zhang Q., Li Y., Xu L. Edge computing: Vision and challenges // IEEE Internet of Things Journal. 2016. Vol. 3, No. 5. P. 637–646.
107. Satyanarayanan M. The emergence of edge computing // Computer. 2017. Vol. 50, No. 1. P. 30–39.
108. Mao Y., You C., Zhang J., Huang K., Letaief K. B. A survey on mobile edge computing: The communication perspective // IEEE Communications Surveys & Tutorials. 2017. Vol. 19, No. 4. P. 2322–2358.
109. Armbrust M., Fox A., Griffith R. et al. A view of cloud computing // Communications of the ACM. 2010. Vol. 53, No. 4. P. 50–58.
110. Cisco Systems. Cisco Video Surveillance Storage System Design Guide. Cisco, 2014.
111. Axis Communications. Bitrate and storage calculations for video surveillance systems. Axis Communications, 2023.

112. Milestone Systems. XProtect Corporate Product Overview. Milestone Systems, 2024.
113. Genetec Inc. Security Center: Unified security platform overview. Genetec, 2024.

ДОДАТОК А

Програмна реалізація методу оцінки ефективності інтегрованих систем відеоспостереження та її оптимізація

```

import math
import random
from typing import Dict, List, Tuple

import altair as alt
import pandas as pd
import streamlit as st

st.set_page_config(
    page_title="Калькулятор ефективності ІСВ",
    page_icon="📊",
    layout="wide",
)

ARCH_TYPES = {
    "distributed": "Розподілена",
    "centralized": "Централізована",
    "hybrid": "Гібридна",
    "cloud": "Хмарна",
}

ARCH_BW_LIMITS = {
    "distributed": 1000.0,
    "centralized": 10000.0,
    "hybrid": 20000.0,
    "cloud": 1000.0,
}

ARCH_COST_MULT = {
    "distributed": 1.05,
    "centralized": 1.00,
    "hybrid": 1.10,
    "cloud": 0.95,
}

ARCH_COMPUTE_MULT = {
    "distributed": 0.90,
    "centralized": 1.00,
    "hybrid": 1.05,
    "cloud": 0.75,
}

ARCH_LATENCY_MULT = {
    "distributed": 0.92,
    "centralized": 1.00,
    "hybrid": 0.95,
    "cloud": 1.18,
}

ARCH_OPEX_MULT = {
    "distributed": 1.10,
    "centralized": 1.00,
    "hybrid": 1.08,
    "cloud": 1.25,
}

```

```

ARCH_SCALE_MULT = {
    "distributed": 1.12,
    "centralized": 1.00,
    "hybrid": 1.08,
    "cloud": 0.92,
}

ARCH_RELIABILITY_SHIFT = {
    "distributed": 0.010,
    "centralized": 0.000,
    "hybrid": 0.012,
    "cloud": -0.008,
}

ARCH_FAULT_TOLERANCE_SHIFT = {
    "distributed": 0.030,
    "centralized": 0.000,
    "hybrid": 0.020,
    "cloud": 0.010,
}

DEFAULT_LOCAL_WEIGHTS = {
    "tech": {"A_i": 0.22, "FP_i": 0.13, "FN_i": 0.18, "L_i": 0.09, "BW_i": 0.08, "C_i": 0.10, "R_i": 0.10, "F_i": 0.10},
    "econ": {"C0_i": 0.18, "Cop_i": 0.13, "Csc_i": 0.10, "Cch_i": 0.12, "Cdet_i": 0.17, "Linc_i": 0.20, "P_i": 0.10},
    "ops": {"S_i": 0.25, "Q_i": 0.20, "Qfat_i": 0.20, "T_i": 0.20, "U_i": 0.15},
}

DEFAULT_GROUP_WEIGHTS = {"tech": 0.40, "econ": 0.30, "ops": 0.30}

DEFAULT_CONSTRAINTS = {
    "B_max": 3_000_000.0,
    "A_min": 0.92,
    "BW_max": 1_500.0,
    "C_max": 90.0,
}

DEFAULT_CONFIG = {
    "system_name": "Система відеоспостереження об'єкта",
    "arch_type": "hybrid",
    "channels": 100,
    "cameras_mp": 4.0,
    "fps": 15,
    "codec_factor": 1.00,
    "ai_accuracy": 0.94,
    "false_positive_rate": 0.07,
    "false_negative_rate": 0.09,
    "latency_ms": 450.0,
    "compute_load_pct": 65.0,
    "reliability": 0.97,
    "fault_tolerance": 0.92,
    "capex_uah": 1_450_000.0,
    "opex_uah_year": 180_000.0,
    "scale_cost_uah": 320_000.0,
    "license_uah_per_channel": 6200.0,
    "incident_loss_uah": 120_000.0,
    "payback_months": 24.0,
    "security_coverage": 0.83,
    "analytics_relevance": 0.88,
    "response_score": 0.79,
    "automation_level": 0.76,
    "operator_hour_cost_uah": 180.0,
    "operator_actions_per_event_min": 2.5,
    "events_per_hour": 55,
    "incident_rate_per_hour": 2.0,
}

```

```

    "background_event_rate_per_hour": 80.0,
}

DEFAULT_OPTIMIZATION = {
    "arch_options": ["centralized", "distributed", "hybrid", "cloud"],
    "n_variants": 100,
    "random_seed": 42,
    "ai_accuracy_range": (0.90, 0.97),
    "false_positive_range": (0.03, 0.10),
    "false_negative_range": (0.03, 0.13),
    "latency_range": (250.0, 650.0),
    "compute_load_range": (45.0, 75.0),
    "capex_range": (900000.0, 1800000.0),
    "opex_range": (120000.0, 240000.0),
    "scale_range": (180000.0, 380000.0),
    "coverage_range": (0.72, 0.90),
    "relevance_range": (0.72, 0.92),
    "response_range": (0.72, 0.90),
    "automation_range": (0.60, 0.90),
}

CONFIG_LABELS = {
    "system_name": "Назва системи",
    "arch_type": "Тип побудови системи",
    "channels": "Кількість камер / каналів",
    "cameras_mp": "Середня роздільна здатність камери",
    "fps": "Кадрів за секунду",
    "codec_factor": "Коефіцієнт стиснення відео",
    "ai_accuracy": "Точність аналітики",
    "false_positive_rate": "Частка хибних спрацьовувань",
    "false_negative_rate": "Частка пропусків інцидентів",
    "latency_ms": "Базова затримка обробки",
    "compute_load_pct": "Базове обчислювальне навантаження",
    "reliability": "Базова надійність системи",
    "fault_tolerance": "Базова відмовостійкість системи",
    "capex_uah": "Початкові витрати без ліцензій",
    "opex_uah_year": "Базові експлуатаційні витрати на рік",
    "scale_cost_uah": "Базова вартість масштабування",
    "license_uah_per_channel": "Вартість ліцензії на канал",
    "incident_loss_uah": "Середній збиток від пропуску одного інциденту",
    "payback_months": "Строк окупності",
    "security_coverage": "Базове покриття задач безпеки",
    "analytics_relevance": "Базова релевантність аналітики задачам об'єкта",
    "response_score": "Базова оперативність реагування",
    "automation_level": "Рівень автоматизації",
    "operator_hour_cost_uah": "Вартість години роботи оператора",
    "operator_actions_per_event_min": "Середній час обробки однієї події оператором",
    "events_per_hour": "Подій на оператора за годину",
    "incident_rate_per_hour": "Середня кількість реальних інцидентів за годину",
    "background_event_rate_per_hour": "Фонових подій за годину",
}

TECH_LABELS = {
    "A_i": "Точність аналітики",
    "FP_i": "Частка хибних спрацьовувань",
    "FN_i": "Частка пропусків інцидентів",
    "L_i": "Затримка обробки",
    "BW_i": "Потрібна пропускна здатність",
    "BW_top_max_i": "Ліміт пропускної здатності для архітектури",
    "C_i": "Обчислювальне навантаження",
    "R_i": "Надійність системи",
    "F_i": "Відмовостійкість системи",
}

ECON_LABELS = {

```

```

"C0_i": "Сукупні початкові витрати",
"Cop_i": "Експлуатаційні витрати на рік",
"Csc_i": "Вартість масштабування",
"Cch_i": "Вартість одного каналу",
"Cdet_i": "Вартість однієї корисної детекції",
"Linc_i": "Очікуваний збиток від пропуску інцидентів",
"Cfp_i": "Вартість обробки хибних подій",
"P_i": "Строк окупності",
}

OPS_LABELS = {
"S_i": "Покриття задач безпеки з урахуванням пропусків",
"Q_i": "Релевантність аналітики задачам об'єкта",
"Qfat_i": "Нормований індекс втоми оператора",
"Qfat_raw": "Розрахунковий індекс втоми оператора до нормування",
"T_i": "Оперативність реагування з урахуванням втоми",
"U_i": "Рівень автоматизації",
}

UNITS = {
"system_name": "—",
"arch_type": "—",
"channels": "канали",
"cameras_mp": "Мп",
"fps": "fps",
"codec_factor": "—",
"ai_accuracy": "частка",
"false_positive_rate": "частка",
"false_negative_rate": "частка",
"latency_ms": "мс",
"compute_load_pct": "%",
"reliability": "частка",
"fault_tolerance": "частка",
"capex_uah": "грн",
"opex_uah_year": "грн/рік",
"scale_cost_uah": "грн",
"license_uah_per_channel": "грн/канал",
"incident_loss_uah": "грн",
"payback_months": "міс.",
"security_coverage": "частка",
"analytics_relevance": "частка",
"response_score": "частка",
"automation_level": "частка",
"operator_hour_cost_uah": "грн/год",
"operator_actions_per_event_min": "хв",
"events_per_hour": "подій/год",
"incident_rate_per_hour": "подій/год",
"background_event_rate_per_hour": "подій/год",
"A_i": "частка",
"FP_i": "частка",
"FN_i": "частка",
"L_i": "мс",
"BW_i": "Мбіт/с",
"BW_top_max_i": "Мбіт/с",
"C_i": "%",
"R_i": "частка",
"F_i": "частка",
"C0_i": "грн",
"Cop_i": "грн/рік",
"Csc_i": "грн",
"Cch_i": "грн/канал",
"Cdet_i": "грн/детекцію",
"Linc_i": "грн/год",
"Cfp_i": "грн/год",
"P_i": "міс.",

```

```

"S_i": "частка",
"Q_i": "частка",
"Qfat_i": "частка",
"Qfat_raw": "безрозмірна",
"T_i": "частка",
"U_i": "частка",
"precision": "частка",
"recall": "частка",
"f1_score": "частка",
"tp_hour": "подій/год",
"fp_hour": "подій/год",
"fn_hour": "подій/год",
"useful_detections_year": "детекцій/рік",
"technical_efficiency": "безрозмірна",
"economic_efficiency": "безрозмірна",
"functional_efficiency": "безрозмірна",
"technical_base": "безрозмірна",
"technical_ai_quality": "безрозмірна",
"economic_base": "безрозмірна",
"economic_risk_score": "безрозмірна",
"functional_base": "безрозмірна",
"integrated_efficiency": "безрозмірна",
"integrated_efficiency_penalized": "безрозмірна",
"constraint_penalty": "безрозмірна",
}

```

```
def ensure_state() -> None:
```

```

    if "constraints" not in st.session_state:
        st.session_state["constraints"] = DEFAULT_CONSTRAINTS.copy()
    if "local_weights" not in st.session_state:
        st.session_state["local_weights"] = {k: v.copy() for k, v in DEFAULT_LOCAL_WEIGHTS.items()}
    if "group_weights" not in st.session_state:
        st.session_state["group_weights"] = DEFAULT_GROUP_WEIGHTS.copy()
    if "config" not in st.session_state:
        st.session_state["config"] = DEFAULT_CONFIG.copy()
    if "last_result" not in st.session_state:
        st.session_state["last_result"] = None
    if "optimization" not in st.session_state:
        st.session_state["optimization"] = DEFAULT_OPTIMIZATION.copy()
    if "optimization_results" not in st.session_state:
        st.session_state["optimization_results"] = None
    if "optimization_details" not in st.session_state:
        st.session_state["optimization_details"] = {}

```

```
def normalize_weights(weights: Dict[str, float]) -> Dict[str, float]:
```

```

    total = sum(float(v) for v in weights.values())
    if total <= 0:
        return {k: 1.0 / len(weights) for k in weights}
    return {k: float(v) / total for k, v in weights.items()}

```

```
def clamp01(value: float) -> float:
```

```
    return max(0.0, min(1.0, float(value)))
```

```
def stratified_samples(low: float, high: float, n: int, seed: int) -> List[float]:
```

```

    if n <= 1:
        return [float(low)]
    if math.isclose(low, high):
        return [float(low)] * n
    rng = random.Random(seed)
    fractions = [((i + rng.random()) / n) for i in range(n)]
    rng.shuffle(fractions)
    return [low + frac * (high - low) for frac in fractions]

```

```
def compute_bandwidth(channels: float, cameras_mp: float, fps: float, codec_factor: float, arch_type: str) -> float:
```

```
    base = channels * cameras_mp * fps * 0.12 * codec_factor
```

```

if arch_type == "distributed":
    return base * 0.85
if arch_type == "centralized":
    return base * 1.05
if arch_type == "hybrid":
    return base * 0.95
if arch_type == "cloud":
    return base * 1.15
return base

def compute_license_total(channels: float, license_per_channel: float) -> float:
    return channels * license_per_channel

def compute_cdet(capex_uah: float, opex_uah_year: float, useful_detections_per_year: float) -> float:
    if useful_detections_per_year <= 0:
        return 1e9
    annualized_cost = capex_uah / 3.0 + opex_uah_year
    return annualized_cost / useful_detections_per_year

def compute_qfat_raw(events_per_hour: float, fp_rate: float, operator_actions_per_event_min: float) -> float:
    q0 = 0.15
    k1 = 0.006
    k2 = 1.10
    return q0 + k1 * events_per_hour * operator_actions_per_event_min + k2 * fp_rate

def normalize_by_target(value: float, best: float, worst: float, maximize: bool) -> float:
    if math.isclose(best, worst):
        return 1.0
    if maximize:
        return clamp01((value - worst) / (best - worst))
    return clamp01((worst - value) / (worst - best))

def calculate_model(config: Dict[str, float], constraints: Dict[str, float], local_weights: Dict[str, Dict[str, float]], group_weights: Dict[str, float]) -> Dict[str, object]:
    arch = config["arch_type"]
    architecture_label = ARCH_TYPES[arch]

    bw_required = compute_bandwidth(config["channels"], config["cameras_mp"], config["fps"], config["codec_factor"], arch)
    bw_limit_arch = ARCH_BW_LIMITS[arch]
    license_total = compute_license_total(config["channels"], config["license_uah_per_channel"])

    adjusted_capex = (config["capex_uah"] + license_total) * ARCH_COST_MULT[arch]
    adjusted_opex = config["opex_uah_year"] * ARCH_OPEX_MULT[arch]
    adjusted_scale_cost = config["scale_cost_uah"] * ARCH_SCALE_MULT[arch]
    adjusted_latency = config["latency_ms"] * ARCH_LATENCY_MULT[arch]
    adjusted_compute = config["compute_load_pct"] * ARCH_COMPUTE_MULT[arch]
    adjusted_reliability = clamp01(config["reliability"]) + ARCH_RELIABILITY_SHIFT[arch]
    adjusted_fault_tolerance = clamp01(config["fault_tolerance"]) + ARCH_FAULT_TOLERANCE_SHIFT[arch]

    tp_hour = config["incident_rate_per_hour"] * (1.0 - config["false_negative_rate"])
    fn_hour = config["incident_rate_per_hour"] * config["false_negative_rate"]
    fp_hour = config["background_event_rate_per_hour"] * config["false_positive_rate"]

    precision = tp_hour / max(tp_hour + fp_hour, 1e-9)
    recall = tp_hour / max(tp_hour + fn_hour, 1e-9)
    f1_score = 2 * precision * recall / max(precision + recall, 1e-9)

    useful_detections_per_hour = tp_hour
    useful_detections_per_year = max(useful_detections_per_hour * 24 * 365, 1.0)

    operator_fp_cost_per_hour = config["operator_hour_cost_uah"] * (fp_hour * config["operator_actions_per_event_min"] / 60.0)
    expected_loss_from_missed_incidents = config["incident_loss_uah"] * fn_hour
    cost_per_detection = compute_cdet(adjusted_capex, adjusted_opex, useful_detections_per_year)
    cost_per_channel = adjusted_capex / max(config["channels"], 1)

```

```

qfat_raw = compute_qfat_raw(config["events_per_hour"], config["false_positive_rate"], config["operator_actions_per_event_min"])
qfat_normalized = clamp01(qfat_raw)

corrected_security_coverage = clamp01(config["security_coverage"] * (1 - 0.8 * config["false_negative_rate"]))
corrected_analytics_relevance = clamp01(config["analytics_relevance"] - 0.5 * config["false_negative_rate"] - 0.3 *
config["false_positive_rate"])
corrected_response_score = clamp01(config["response_score"] - 0.25 * qfat_normalized - 0.10 * config["false_negative_rate"])

raw = {
    "A_i": float(config["ai_accuracy"]),
    "FP_i": float(config["false_positive_rate"]),
    "FN_i": float(config["false_negative_rate"]),
    "L_i": float(adjusted_latency),
    "BW_i": float(bw_required),
    "BW_top_max_i": float(bw_limit_arch),
    "C_i": float(adjusted_compute),
    "R_i": float(adjusted_reliability),
    "F_i": float(adjusted_fault_tolerance),
    "C0_i": float(adjusted_capex),
    "Cop_i": float(adjusted_opex),
    "Csc_i": float(adjusted_scale_cost),
    "Cch_i": float(cost_per_channel),
    "Cdet_i": float(cost_per_detection),
    "Linc_i": float(expected_loss_from_missed_incidents),
    "Cfp_i": float(operator_fp_cost_per_hour),
    "P_i": float(config["payback_months"]),
    "S_i": float(corrected_security_coverage),
    "Q_i": float(corrected_analytics_relevance),
    "Qfat_i": float(qfat_normalized),
    "Qfat_raw": float(qfat_raw),
    "T_i": float(corrected_response_score),
    "U_i": float(config["automation_level"]),
}

normalized = {
    "A_i": normalize_by_target(raw["A_i"], 1.0, 0.0, True),
    "FP_i": normalize_by_target(raw["FP_i"], 0.0, 0.50, False),
    "FN_i": normalize_by_target(raw["FN_i"], 0.0, 0.50, False),
    "L_i": normalize_by_target(raw["L_i"], 100.0, 1500.0, False),
    "BW_i": normalize_by_target(raw["BW_i"], 0.0, constraints["BW_max"], False),
    "C_i": normalize_by_target(raw["C_i"], 0.0, constraints["C_max"], False),
    "R_i": normalize_by_target(raw["R_i"], 1.0, 0.0, True),
    "F_i": normalize_by_target(raw["F_i"], 1.0, 0.0, True),
    "C0_i": normalize_by_target(raw["C0_i"], 0.0, constraints["B_max"], False),
    "Cop_i": normalize_by_target(raw["Cop_i"], 0.0, 500000.0, False),
    "Csc_i": normalize_by_target(raw["Csc_i"], 0.0, 1000000.0, False),
    "Cch_i": normalize_by_target(raw["Cch_i"], 0.0, 50000.0, False),
    "Cdet_i": normalize_by_target(raw["Cdet_i"], 0.0, 2000.0, False),
    "Linc_i": normalize_by_target(raw["Linc_i"], 0.0, config["incident_loss_uah"] * max(config["incident_rate_per_hour"], 1.0),
False),
    "Cfp_i": normalize_by_target(raw["Cfp_i"], 0.0, 50000.0, False),
    "P_i": normalize_by_target(raw["P_i"], 0.0, 60.0, False),
    "S_i": normalize_by_target(raw["S_i"], 1.0, 0.0, True),
    "Q_i": normalize_by_target(raw["Q_i"], 1.0, 0.0, True),
    "Qfat_i": normalize_by_target(raw["Qfat_i"], 0.0, 1.0, False),
    "T_i": normalize_by_target(raw["T_i"], 1.0, 0.0, True),
    "U_i": normalize_by_target(raw["U_i"], 1.0, 0.0, True),
    "precision": normalize_by_target(precision, 1.0, 0.0, True),
    "recall": normalize_by_target(recall, 1.0, 0.0, True),
    "f1_score": normalize_by_target(f1_score, 1.0, 0.0, True),
}

lw = {k: normalize_weights(v) for k, v in local_weights.items()}
gw = normalize_weights(group_weights)

```

```

technical_base = sum(normalized[k] * lw["tech"][k] for k in lw["tech"])
technical_ai_quality = 0.30 * normalized["A_i"] + 0.25 * normalized["precision"] + 0.25 * normalized["recall"] + 0.20 *
normalized["f1_score"]
technical_efficiency = 0.70 * technical_base + 0.30 * technical_ai_quality

econ_base_weights = normalize_weights({k: lw["econ"][k] for k in ["C0_i", "Cop_i", "Csc_i", "Cch_i", "Cdet_i", "P_i"]})
economic_base = sum(normalized[k] * econ_base_weights[k] for k in econ_base_weights)
economic_risk_score = 0.70 * normalized["Linc_i"] + 0.30 * normalized["Cfp_i"]
economic_efficiency = 0.75 * economic_base + 0.25 * economic_risk_score

functional_base = 0.25 * normalized["S_i"] + 0.20 * normalized["Q_i"] + 0.20 * normalized["Qfat_i"] + 0.20 * normalized["T_i"] +
0.15 * normalized["U_i"]
functional_efficiency = functional_base

integrated_efficiency = gw["tech"] * technical_efficiency + gw["econ"] * economic_efficiency + gw["ops"] * functional_efficiency

feasibility_checks = {
    "Бюджет": raw["C0_i"] <= constraints["B_max"],
    "Мінімальна точність": raw["A_i"] >= constraints["A_min"],
    "Ліміт архітектури": raw["BW_i"] <= raw["BW_top_max_i"],
    "Глобальний ліміт мережі": raw["BW_i"] <= constraints["BW_max"],
    "Обчислювальне навантаження": raw["C_i"] <= constraints["C_max"],
}
is_feasible = all(feasibility_checks.values())

constraint_penalty = 0.0
if raw["C0_i"] > constraints["B_max"]:
    constraint_penalty += 0.20
if raw["A_i"] < constraints["A_min"]:
    constraint_penalty += 0.25
if raw["BW_i"] > raw["BW_top_max_i"]:
    constraint_penalty += 0.15
if raw["BW_i"] > constraints["BW_max"]:
    constraint_penalty += 0.15
if raw["C_i"] > constraints["C_max"]:
    constraint_penalty += 0.15

integrated_efficiency_penalized = integrated_efficiency - constraint_penalty

contributions = {
    "Технічна ефективність": gw["tech"] * technical_efficiency,
    "Економічна ефективність": gw["econ"] * economic_efficiency,
    "Функціональна ефективність": gw["ops"] * functional_efficiency,
    "Штраф за обмеження": -constraint_penalty,
}

architecture_effects = {
    "Тип архітектури": architecture_label,
    "Мультиплікатор вартості": ARCH_COST_MULT[arch],
    "Мультиплікатор обчислювального навантаження": ARCH_COMPUTE_MULT[arch],
    "Мультиплікатор затримки": ARCH_LATENCY_MULT[arch],
    "Мультиплікатор OPEX": ARCH_OPEX_MULT[arch],
    "Мультиплікатор масштабування": ARCH_SCALE_MULT[arch],
    "Поправка до надійності": ARCH_RELIABILITY_SHIFT[arch],
    "Поправка до відмовостійкості": ARCH_FAULT_TOLERANCE_SHIFT[arch],
    "Ліміт пропускну здатності для архітектури": raw["BW_top_max_i"],
}

return {
    "architecture_label": architecture_label,
    "architecture_effects": architecture_effects,
    "raw": raw,
    "normalized": normalized,
    "local_weights": lw,
    "group_weights": gw,

```

```

"technical_base": technical_base,
"technical_ai_quality": technical_ai_quality,
"technical_efficiency": technical_efficiency,
"economic_base": economic_base,
"economic_risk_score": economic_risk_score,
"economic_efficiency": economic_efficiency,
"functional_base": functional_base,
"functional_efficiency": functional_efficiency,
"integrated_efficiency": integrated_efficiency,
"integrated_efficiency_penalized": integrated_efficiency_penalized,
"constraint_penalty": constraint_penalty,
"is_feasible": is_feasible,
"feasibility_checks": feasibility_checks,
"tp_hour": tp_hour,
"fp_hour": fp_hour,
"fn_hour": fn_hour,
"precision": precision,
"recall": recall,
"f1_score": f1_score,
"useful_detections_per_year": useful_detections_per_year,
"contributions": contributions,
}

```

```

def build_parameter_table(config: Dict[str, float], result: Dict[str, object], constraints: Dict[str, float]) -> pd.DataFrame:
    rows: List[Dict[str, object]] = []

```

```

    for key, label in CONFIG_LABELS.items():
        value = config[key]
        if key == "arch_type":
            value = result["architecture_label"]
        rows.append({
            "Група": "Вхідні дані",
            "Параметр": label,
            "Значення": value,
            "Одиниця": UNITS.get(key, ""),
            "Джерело": "Введено вручну",
        })

    for name, value in result["architecture_effects"].items():
        unit = ""
        if "Ліміт пропускної здатності" in name:
            unit = "Мбіт/с"
        rows.append({
            "Група": "Вплив архітектури",
            "Параметр": name,
            "Значення": value,
            "Одиниця": unit,
            "Джерело": "Статичний коефіцієнт моделі",
        })

    for key, label in TECH_LABELS.items():
        rows.append({
            "Група": "Технічні параметри",
            "Параметр": label,
            "Значення": result["raw"][key],
            "Одиниця": UNITS.get(key, ""),
            "Джерело": "Розраховано",
        })

    for key, label in ECON_LABELS.items():
        rows.append({
            "Група": "Економічні параметри",
            "Параметр": label,
            "Значення": result["raw"][key],
            "Одиниця": UNITS.get(key, ""),

```

```

        "Джерело": "Розраховано",
    })

for key, label in OPS_LABELS.items():
    rows.append({
        "Група": "Функціональні параметри",
        "Параметр": label,
        "Значення": result["raw"][key],
        "Одиниця": UNITS.get(key, ""),
        "Джерело": "Розраховано",
    })

derived_rows = [
    ("Похідні показники аналітики", "Очікувана кількість правильно виявлених інцидентів за годину", result["tp_hour"],
"подій/год"),
    ("Похідні показники аналітики", "Очікувана кількість хибних подій за годину", result["fp_hour"], "подій/год"),
    ("Похідні показники аналітики", "Очікувана кількість пропущених інцидентів за годину", result["fn_hour"], "подій/год"),
    ("Похідні показники аналітики", "Precision", result["precision"], "частка"),
    ("Похідні показники аналітики", "Recall", result["recall"], "частка"),
    ("Похідні показники аналітики", "F1-score", result["f1_score"], "частка"),
    ("Похідні показники аналітики", "Корисні детекції за рік", result["useful_detections_per_year"], "детекцій/рік"),
]
for group, name, value, unit in derived_rows:
    rows.append({"Група": group, "Параметр": name, "Значення": value, "Одиниця": unit, "Джерело": "Розраховано"})

label_map = {}
label_map.update(TECH_LABELS)
label_map.update(ECON_LABELS)
label_map.update(OPS_LABELS)
label_map.update({"precision": "Нормоване значення precision", "recall": "Нормоване значення recall", "f1_score": "Нормоване
значення F1-score"})
for key, value in result["normalized"].items():
    rows.append({
        "Група": "Нормовані параметри",
        "Параметр": label_map.get(key, key),
        "Значення": value,
        "Одиниця": "безрозмірна",
        "Джерело": "Розраховано",
    })

model_rows = [
    ("Локальні та інтегральні оцінки", "Базова технічна ефективність", result["technical_base"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Якість AI-компонента", result["technical_ai_quality"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Технічна ефективність", result["technical_efficiency"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Базова економічна ефективність", result["economic_base"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Ризикова економічна складова", result["economic_risk_score"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Економічна ефективність", result["economic_efficiency"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Функціональна ефективність", result["functional_efficiency"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Інтегральна ефективність", result["integrated_efficiency"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Штраф за обмеження", result["constraint_penalty"], "безрозмірна"),
    ("Локальні та інтегральні оцінки", "Інтегральна ефективність зі штрафом", result["integrated_efficiency_penalized"],
"безрозмірна"),
]
for group, name, value, unit in model_rows:
    rows.append({"Група": group, "Параметр": name, "Значення": value, "Одиниця": unit, "Джерело": "Розраховано"})

for key, value in result["group_weights"].items():
    group_name = {
        "tech": "Вага технічної ефективності",
        "econ": "Вага економічної ефективності",
        "ops": "Вага функціональної ефективності",
    }[key]
    rows.append({
        "Група": "Вагові коефіцієнти",
        "Параметр": group_name,
    })

```

```

        "Значення": value,
        "Одиниця": "частка",
        "Джерело": "Налаштування моделі",
    })

    for group_key, group_name in [("tech", "Локальні ваги технічних критеріїв"), ("econ", "Локальні ваги економічних критеріїв"),
    ("ops", "Локальні ваги функціональних критеріїв")]:
        for key, value in result["local_weights"][group_key].items():
            rows.append({
                "Група": "Вагові коефіцієнти",
                "Параметр": f"{{{group_name}: {label_map.get(key, key)}}",
                "Значення": value,
                "Одиниця": "частка",
                "Джерело": "Налаштування моделі",
            })

    for key, value in constraints.items():
        rows.append({
            "Група": "Обмеження моделі",
            "Параметр": {
                "B_max": "Максимальний бюджет",
                "A_min": "Мінімальна точність аналітики",
                "BW_max": "Глобальний ліміт мережі",
                "C_max": "Максимальне обчислювальне навантаження",
            }[key],
            "Значення": value,
            "Одиниця": {
                "B_max": "грн",
                "A_min": "частка",
                "BW_max": "Мбіт/с",
                "C_max": "%",
            }[key],
            "Джерело": "Налаштування моделі",
        })

    for name, state in result["feasibility_checks"].items():
        rows.append({
            "Група": "Перевірка обмежень",
            "Параметр": name,
            "Значення": "Так" if state else "Ні",
            "Одиниця": "—",
            "Джерело": "Перевірка",
        })

    return pd.DataFrame(rows)

def run_optimization(base_config: Dict[str, float], constraints: Dict[str, float], local_weights: Dict[str, Dict[str, float]], group_weights:
Dict[str, float], opt_settings: Dict[str, object]) -> Tuple[pd.DataFrame, Dict[int, Dict[str, object]]]:
    n_variants = int(opt_settings["n_variants"])
    seed = int(opt_settings["random_seed"])
    arch_options = opt_settings["arch_options"] or [base_config["arch_type"]]

    sampled = {
        "ai_accuracy": stratified_samples(opt_settings["ai_accuracy_range"][0], opt_settings["ai_accuracy_range"][1], n_variants, seed
+ 11),
        "false_positive_rate": stratified_samples(opt_settings["false_positive_range"][0], opt_settings["false_positive_range"][1],
n_variants, seed + 21),
        "false_negative_rate": stratified_samples(opt_settings["false_negative_range"][0], opt_settings["false_negative_range"][1],
n_variants, seed + 31),
        "latency_ms": stratified_samples(opt_settings["latency_range"][0], opt_settings["latency_range"][1], n_variants, seed + 41),
        "compute_load_pct": stratified_samples(opt_settings["compute_load_range"][0], opt_settings["compute_load_range"][1],
n_variants, seed + 51),
        "capex_uah": stratified_samples(opt_settings["capex_range"][0], opt_settings["capex_range"][1], n_variants, seed + 61),
        "opex_uah_year": stratified_samples(opt_settings["opex_range"][0], opt_settings["opex_range"][1], n_variants, seed + 71),
        "scale_cost_uah": stratified_samples(opt_settings["scale_range"][0], opt_settings["scale_range"][1], n_variants, seed + 81),
    }

```

```

    "security_coverage": stratified_samples(opt_settings["coverage_range"][0], opt_settings["coverage_range"][1], n_variants, seed
+ 91),
    "analytics_relevance": stratified_samples(opt_settings["relevance_range"][0], opt_settings["relevance_range"][1], n_variants,
seed + 101),
    "response_score": stratified_samples(opt_settings["response_range"][0], opt_settings["response_range"][1], n_variants, seed +
111),
    "automation_level": stratified_samples(opt_settings["automation_range"][0], opt_settings["automation_range"][1], n_variants,
seed + 121),
}

arch_sequence = list(arch_options)
repeats = math.ceil(n_variants / max(len(arch_sequence), 1))
arch_sequence = (arch_sequence * repeats)[:n_variants]
random.Random(seed + 131).shuffle(arch_sequence)

records: List[Dict[str, object]] = []
details: Dict[int, Dict[str, object]] = {}
for idx in range(n_variants):
    cfg = base_config.copy()
    cfg["system_name"] = f"БapiaHT {idx + 1}"
    cfg["arch_type"] = arch_sequence[idx]
    for key, values in sampled.items():
        cfg[key] = values[idx]

res = calculate_model(cfg, constraints, local_weights, group_weights)
details[idx + 1] = {"config": cfg.copy(), "result": res}
records.append({
    "rank": 0,
    "variant": idx + 1,
    "architecture": res["architecture_label"],
    "is_feasible": res["is_feasible"],
    "integrated_efficiency": res["integrated_efficiency"],
    "integrated_efficiency_penalized": res["integrated_efficiency_penalized"],
    "technical_efficiency": res["technical_efficiency"],
    "economic_efficiency": res["economic_efficiency"],
    "functional_efficiency": res["functional_efficiency"],
    "constraint_penalty": res["constraint_penalty"],
    "A_i": res["raw"]["A_i"],
    "FP_i": res["raw"]["FP_i"],
    "FN_i": res["raw"]["FN_i"],
    "L_i": res["raw"]["L_i"],
    "BW_i": res["raw"]["BW_i"],
    "C_i": res["raw"]["C_i"],
    "C0_i": res["raw"]["C0_i"],
    "Cop_i": res["raw"]["Cop_i"],
    "Csc_i": res["raw"]["Csc_i"],
    "Linc_i": res["raw"]["Linc_i"],
    "Qfat_raw": res["raw"]["Qfat_raw"],
    "Qfat_i": res["raw"]["Qfat_i"],
    "T_i": res["raw"]["T_i"],
    "U_i": res["raw"]["U_i"],
})

df = pd.DataFrame(records)
df = df.sort_values(by=["is_feasible", "integrated_efficiency_penalized", "integrated_efficiency"], ascending=[False, False,
False]).reset_index(drop=True)
df["rank"] = range(1, len(df) + 1)
return df, details

def comparison_chart_df(base_result: Dict[str, object], opt_result: Dict[str, object]) -> pd.DataFrame:
    return pd.DataFrame({
        "Показник": [
            "Технічна ефективність",
            "Економічна ефективність",
            "Функціональна ефективність",

```

```

        "Інтегральна ефективність",
    ],
    "Базова конфігурація": [
        base_result["technical_efficiency"],
        base_result["economic_efficiency"],
        base_result["functional_efficiency"],
        base_result["integrated_efficiency"],
    ],
    "Оптимізована конфігурація": [
        opt_result["technical_efficiency"],
        opt_result["economic_efficiency"],
        opt_result["functional_efficiency"],
        opt_result["integrated_efficiency"],
    ],
}

def constraints_sidebar() -> None:
    st.sidebar.header("Обмеження моделі")
    c = st.session_state["constraints"]
    c["B_max"] = st.sidebar.number_input("Максимальний бюджет, грн", min_value=0.0, value=float(c["B_max"]), step=100000.0)
    c["A_min"] = st.sidebar.slider("Мінімальна точність аналітики", min_value=0.50, max_value=0.999, value=float(c["A_min"]),
    step=0.005)
    c["BW_max"] = st.sidebar.number_input("Глобальний ліміт мережі, Мбіт/с", min_value=0.0, value=float(c["BW_max"]),
    step=50.0)
    c["C_max"] = st.sidebar.number_input("Максимальне обчислювальне навантаження, %", min_value=0.0, max_value=100.0,
    value=float(c["C_max"]), step=1.0)

def weights_sidebar() -> None:
    st.sidebar.header("Групові ваги")
    gw = st.session_state["group_weights"]
    gw["tech"] = st.sidebar.slider("Вага технічної ефективності", 0.0, 1.0, float(gw["tech"]), 0.01)
    gw["econ"] = st.sidebar.slider("Вага економічної ефективності", 0.0, 1.0, float(gw["econ"]), 0.01)
    gw["ops"] = st.sidebar.slider("Вага функціональної ефективності", 0.0, 1.0, float(gw["ops"]), 0.01)
    st.session_state["group_weights"] = normalize_weights(gw)

    with st.sidebar.expander("Локальні ваги критеріїв"):
        for group_key, group_name in [("tech", "Технічні"), ("econ", "Економічні"), ("ops", "Функціональні")]:
            st.markdown(f"***{group_name}**")
            current = st.session_state["local_weights"][group_key]
            updated = {}
            for k, v in current.items():
                updated[k] = st.number_input(f"{group_key}:{k}", value=float(v), min_value=0.0, step=0.01, key=f"lw_{group_key}_{k}")
            st.session_state["local_weights"][group_key] = normalize_weights(updated)
            st.caption("Σ = {:.2f}".format(sum(st.session_state["local_weights"][group_key].values())))

def calculator_tab() -> None:
    st.subheader("Калькулятор ефективності")
    st.caption("Дані для оптимізації беруться з цієї конфігурації як з базового сценарію. На вкладці оптимізації змінюються лише параметри, які ви дозволите перебирати.")
    cfg = st.session_state["config"]

    with st.form("calculator_form"):
        c1, c2, c3 = st.columns(3)
        with c1:
            st.markdown("***1. Параметри системи***")
            cfg["system_name"] = st.text_input("Назва системи", value=cfg["system_name"])
            cfg["arch_type"] = st.selectbox("Тип побудови системи", options=list(ARCH_TYPES.keys()), format_func=lambda x: ARCH_TYPES[x], index=list(ARCH_TYPES.keys()).index(cfg["arch_type"]))
            cfg["channels"] = st.number_input("Кількість камер / каналів", min_value=1, value=int(cfg["channels"]), step=1)
            cfg["cameras_mp"] = st.number_input("Середня роздільна здатність камери, Мп", min_value=0.1,
            value=float(cfg["cameras_mp"]), step=0.1)
            cfg["fps"] = st.number_input("Кадрів за секунду", min_value=1, value=int(cfg["fps"]), step=1)
            cfg["codec_factor"] = st.number_input("Коефіцієнт стиснення відео", min_value=0.2, value=float(cfg["codec_factor"]),
            step=0.01)

```

```

    cfg["compute_load_pct"] = st.slider("Базове завантаження обчислювальної платформи, %", 1.0, 100.0,
float(cfg["compute_load_pct"]), 1.0)
    cfg["reliability"] = st.slider("Базова надійність системи", 0.5, 0.999, float(cfg["reliability"]), 0.001)
    cfg["fault_tolerance"] = st.slider("Базова відмовостійкість системи", 0.5, 0.999, float(cfg["fault_tolerance"]), 0.001)
    with c2:
        st.markdown("***2. Параметри аналітики***")
        cfg["ai_accuracy"] = st.slider("Точність аналітики", 0.5, 0.999, float(cfg["ai_accuracy"]), 0.001)
        cfg["false_positive_rate"] = st.slider("Частка хибних спрацьовувань", 0.0, 0.5, float(cfg["false_positive_rate"]), 0.001)
        cfg["false_negative_rate"] = st.slider("Частка пропусків інцидентів", 0.0, 0.5, float(cfg["false_negative_rate"]), 0.001)
        cfg["latency_ms"] = st.number_input("Базова затримка обробки, мс", min_value=1.0, value=float(cfg["latency_ms"]),
step=10.0)
        cfg["events_per_hour"] = st.number_input("Подій на оператора за годину", min_value=0,
value=int(cfg["events_per_hour"]), step=1)
        cfg["incident_rate_per_hour"] = st.number_input("Середня кількість реальних інцидентів за годину", min_value=0.0,
value=float(cfg["incident_rate_per_hour"]), step=0.1)
        cfg["background_event_rate_per_hour"] = st.number_input("Фонових подій за годину", min_value=0.0,
value=float(cfg["background_event_rate_per_hour"]), step=1.0)
        cfg["operator_actions_per_event_min"] = st.number_input("Середній час обробки однієї події оператором, хв",
min_value=0.1, value=float(cfg["operator_actions_per_event_min"]), step=0.1)
        cfg["operator_hour_cost_uah"] = st.number_input("Вартість години роботи оператора, грн", min_value=0.0,
value=float(cfg["operator_hour_cost_uah"]), step=10.0)
    with c3:
        st.markdown("***3. Витрати та функції***")
        cfg["capex_uah"] = st.number_input("Початкові витрати без ліцензій, грн", min_value=0.0, value=float(cfg["capex_uah"]),
step=10000.0)
        cfg["opex_uah_year"] = st.number_input("Базові експлуатаційні витрати на рік, грн", min_value=0.0,
value=float(cfg["opex_uah_year"]), step=10000.0)
        cfg["scale_cost_uah"] = st.number_input("Базова вартість масштабування, грн", min_value=0.0,
value=float(cfg["scale_cost_uah"]), step=10000.0)
        cfg["license_uah_per_channel"] = st.number_input("Вартість ліцензії на канал, грн", min_value=0.0,
value=float(cfg["license_uah_per_channel"]), step=100.0)
        cfg["incident_loss_uah"] = st.number_input("Середній збиток від пропуску одного інциденту, грн", min_value=0.0,
value=float(cfg["incident_loss_uah"]), step=10000.0)
        cfg["payback_months"] = st.number_input("Строк окупності, місяців", min_value=0.0, value=float(cfg["payback_months"]),
step=1.0)
        cfg["security_coverage"] = st.slider("Базове покриття задач безпеки", 0.0, 1.0, float(cfg["security_coverage"]), 0.01)
        cfg["analytics_relevance"] = st.slider("Базова релевантність аналітики задачам об'єкта", 0.0, 1.0,
float(cfg["analytics_relevance"]), 0.01)
        cfg["response_score"] = st.slider("Базова оперативність реагування", 0.0, 1.0, float(cfg["response_score"]), 0.01)
        cfg["automation_level"] = st.slider("Рівень автоматизації", 0.0, 1.0, float(cfg["automation_level"]), 0.01)

    submitted = st.form_submit_button("Позрахувати", type="primary")

    st.session_state["config"] = cfg
    if submitted or st.session_state["last_result"] is None:
        st.session_state["last_result"] = calculate_model(cfg, st.session_state["constraints"], st.session_state["local_weights"],
st.session_state["group_weights"])

    result = st.session_state["last_result"]

    st.markdown("### Підсумок розрахунку")
    m1, m2, m3, m4, m5 = st.columns(5)
    m1.metric("Інтегральна ефективність", f"{result['integrated_efficiency']:.4f}")
    m2.metric("Технічна ефективність", f"{result['technical_efficiency']:.4f}")
    m3.metric("Економічна ефективність", f"{result['economic_efficiency']:.4f}")
    m4.metric("Функціональна ефективність", f"{result['functional_efficiency']:.4f}")
    m5.metric("Архітектура", result["architecture_label"])

    n1, n2, n3, n4, n5 = st.columns(5)
    n1.metric("Потрібна пропускна здатність, Мбіт/с", f"{result['raw']['BW_i']:.1f}")
    n2.metric("Очікуваний збиток, грн/год", f"{result['raw']['Linc_i']:.0f}")
    n3.metric("Індекс втоми (raw)", f"{result['raw']['Qfat_raw']:.3f}")
    n4.metric("Індекс втоми (нормований)", f"{result['raw']['Qfat_i']:.3f}")
    n5.metric("F1-score", f"{result['f1_score']:.3f}")

```

```

if result["raw"]["Qfat_i"] >= 0.999:
    st.warning("Нормований індекс втоми досяг верхньої межі 1.0. Перевіряйте також raw-значення індексу втоми у підсумку та в деталізованих результатах.")

if result["is_feasible"]:
    st.success("Конфігурація відповідає заданим обмеженням.")
else:
    st.error("Конфігурація не відповідає одному або кільком заданим обмеженням.")

checks_df = pd.DataFrame({
    "Перевірка": list(result["feasibility_checks"].keys()),
    "Стан": ["Так" if v else "Ні" for v in result["feasibility_checks"].values()],
})
st.dataframe(checks_df, use_container_width=True, hide_index=True)

def detailed_results_tab() -> None:
    st.subheader("Деталізовані результати")
    result = st.session_state.get("last_result")
    config = st.session_state.get("config")
    if result is None or config is None:
        st.info("Спочатку виконайте розрахунок на вкладці «Калькулятор ефективності».")
        return

    st.markdown("### Вплив архітектури на модель")
    arch_df = pd.DataFrame({"Параметр": list(result["architecture_effects"].keys()), "Значення":
list(result["architecture_effects"].values())})
    st.dataframe(arch_df, use_container_width=True, hide_index=True)

    st.markdown("### Внесок складових у фінальний результат")
    contrib_df = pd.DataFrame({"Компонент": list(result["contributions"].keys()), "Внесок": list(result["contributions"].values())})
    contrib_chart = alt.Chart(contrib_df).mark_bar().encode(
        x=alt.X("Компонент:N", title="Компонент"),
        y=alt.Y("Внесок:Q", title="Внесок у результат"),
        tooltip=["Компонент", "Внесок"],
    ).properties(height=320)
    st.altair_chart(contrib_chart, use_container_width=True)

    st.markdown("### Порівняння локальних ефективностей")
    local_df = pd.DataFrame({
        "Показник": ["Базова технічна ефективність", "Якість AI-компонента", "Технічна ефективність", "Базова економічна
ефективність", "Ризикова економічна складова", "Економічна ефективність", "Функціональна ефективність", "Інтегральна
ефективність"],
        "Значення": [result["technical_base"], result["technical_ai_quality"], result["technical_efficiency"], result["economic_base"],
result["economic_risk_score"], result["economic_efficiency"], result["functional_efficiency"], result["integrated_efficiency"]],
    })
    local_chart = alt.Chart(local_df).mark_bar().encode(
        x=alt.X("Показник:N", title="Показник"),
        y=alt.Y("Значення:Q", title="Значення"),
        tooltip=["Показник", "Значення"],
    ).properties(height=360)
    st.altair_chart(local_chart, use_container_width=True)

    st.markdown("### Нормовані параметри")
    label_map = {}
    label_map.update(TECH_LABELS)
    label_map.update(ECON_LABELS)
    label_map.update(OPS_LABELS)
    label_map.update({"precision": "Precision", "recall": "Recall", "f1_score": "F1-score"})
    normalized_records = [{"Параметр": label_map.get(key, key), "Нормоване значення": value} for key, value in
result["normalized"].items()]
    norm_df = pd.DataFrame(normalized_records)
    heatmap = alt.Chart(norm_df).mark_bar().encode(
        x=alt.X("Параметр:N", sort="-y", title="Параметр"),
        y=alt.Y("Нормоване значення:Q", title="Нормоване значення"),
        tooltip=["Параметр", "Нормоване значення"],
    )

```

```

).properties(height=360)
st.altair_chart(heatmap, use_container_width=True)

st.markdown("### Усі параметри моделі та розраховані значення")
parameter_table = build_parameter_table(config, result, st.session_state["constraints"])
st.dataframe(parameter_table, use_container_width=True, hide_index=True)

def optimization_tab() -> None:
    st.subheader("Оптимізація")
    st.caption("Тут задаються діапазони пошуку. Базові значення беруться з конфігурації на першій вкладці, а в оптимізації змінюються лише вибрані параметри в межах заданих діапазонів.")
    opt = st.session_state["optimization"]

    with st.form("optimization_form"):
        top1, top2, top3 = st.columns(3)
        with top1:
            opt["arch_options"] = st.multiselect(
                "Архітектури для перебору",
                options=list(ARCH_TYPES.keys()),
                default=opt["arch_options"],
                format_func=lambda x: ARCH_TYPES[x],
            )
        with top2:
            opt["n_variants"] = st.slider("Кількість варіантів для генерації", min_value=10, max_value=1000,
            value=int(opt["n_variants"]), step=10)
        with top3:
            opt["random_seed"] = st.number_input("Початкове значення генератора", min_value=0, value=int(opt["random_seed"]),
            step=1)

        c1, c2, c3 = st.columns(3)
        with c1:
            opt["ai_accuracy_range"] = st.slider("Діапазон точності аналітики", min_value=0.50, max_value=0.999,
            value=opt["ai_accuracy_range"], step=0.001)
            opt["false_positive_range"] = st.slider("Діапазон хибних спрацьовувань", min_value=0.0, max_value=0.5,
            value=opt["false_positive_range"], step=0.001)
            opt["false_negative_range"] = st.slider("Діапазон пропусків інцидентів", min_value=0.0, max_value=0.5,
            value=opt["false_negative_range"], step=0.001)
            opt["latency_range"] = st.slider("Діапазон базової затримки, мс", min_value=50.0, max_value=2000.0,
            value=opt["latency_range"], step=10.0)
        with c2:
            opt["compute_load_range"] = st.slider("Діапазон базового обчислювального навантаження, %", min_value=1.0,
            max_value=100.0, value=opt["compute_load_range"], step=1.0)
            opt["capex_range"] = st.slider("Діапазон CAPEX без ліцензій, грн", min_value=100000.0, max_value=5000000.0,
            value=opt["capex_range"], step=50000.0)
            opt["opex_range"] = st.slider("Діапазон OPEX на рік, грн", min_value=10000.0, max_value=1000000.0,
            value=opt["opex_range"], step=10000.0)
            opt["scale_range"] = st.slider("Діапазон вартості масштабування, грн", min_value=10000.0, max_value=1500000.0,
            value=opt["scale_range"], step=10000.0)
        with c3:
            opt["coverage_range"] = st.slider("Діапазон покриття задач безпеки", min_value=0.0, max_value=1.0,
            value=opt["coverage_range"], step=0.01)
            opt["relevance_range"] = st.slider("Діапазон релевантності аналітики", min_value=0.0, max_value=1.0,
            value=opt["relevance_range"], step=0.01)
            opt["response_range"] = st.slider("Діапазон оперативності реагування", min_value=0.0, max_value=1.0,
            value=opt["response_range"], step=0.01)
            opt["automation_range"] = st.slider("Діапазон рівня автоматизації", min_value=0.0, max_value=1.0,
            value=opt["automation_range"], step=0.01)

        submitted = st.form_submit_button("Занустити оптимізацію", type="primary")

    st.session_state["optimization"] = opt
    arch_list = ", ".join(ARCH_TYPES[a] for a in (opt["arch_options"] or [])) or "не вибрано"
    st.info(f"Буде згенеровано **{opt['n_variants']}** варіантів. Архітектури, що беруть участь: {arch_list}.")

    if submitted:

```

```

try:
    df, details = run_optimization(st.session_state["config"], st.session_state["constraints"], st.session_state["local_weights"],
st.session_state["group_weights"], opt)
    st.session_state["optimization_results"] = df
    st.session_state["optimization_details"] = details
    st.success(f"Оптимізацію завершено. Проаналізовано {len(df)} варіантів.")
except Exception as e:
    st.session_state["optimization_results"] = None
    st.session_state["optimization_details"] = {}
    st.error(f"Помилка під час оптимізації: {e}")

def optimization_results_tab() -> None:
    st.subheader("Результати оптимізації")
    df = st.session_state.get("optimization_results")
    details = st.session_state.get("optimization_details", {})
    if df is None or df.empty:
        st.info("Спочатку запустіть оптимізацію на попередній вкладці.")
        return

    feasible = df[df["is_feasible"]].copy()
    if feasible.empty:
        st.warning("Допустимих конфігурацій не знайдено. Нижче показані найкращі варіанти за штрафною інтегральною оцінкою.")
        top = df.head(10)
        best_row = df.iloc[0]
    else:
        best_row = feasible.iloc[0]
        st.success(f"Рекомендований варіант: №{int(best_row['variant'])} — {best_row['architecture']}")
        top = feasible.head(10)

    m1, m2, m3, m4 = st.columns(4)
    m1.metric("Інтегральна ефективність", f"{best_row['integrated_efficiency']:.4f}")
    m2.metric("Інтегральна ефективність зі штрафом", f"{best_row['integrated_efficiency_penalized']:.4f}")
    m3.metric("Архітектура", best_row["architecture"])
    m4.metric("Індекс втоми (raw)", f"{best_row['Qfat_raw']:.3f}")

    display_df = top[[
        "rank", "variant", "architecture", "is_feasible", "integrated_efficiency", "integrated_efficiency_penalized",
        "technical_efficiency", "economic_efficiency", "functional_efficiency", "A_i", "FP_i", "FN_i", "L_i", "BW_i",
        "C0_i", "Cop_i", "Csc_i", "Linc_i", "Qfat_raw", "Qfat_i", "T_i", "U_i"
    ]].rename(columns={
        "rank": "Ранг",
        "variant": "Варіант",
        "architecture": "Архітектура",
        "is_feasible": "Допустимий",
        "integrated_efficiency": "Інтегральна ефективність",
        "integrated_efficiency_penalized": "Інтегральна ефективність зі штрафом",
        "technical_efficiency": "Технічна ефективність",
        "economic_efficiency": "Економічна ефективність",
        "functional_efficiency": "Функціональна ефективність",
        "A_i": "Точність аналітики",
        "FP_i": "Хибні спрацьовування",
        "FN_i": "Пропуски інцидентів",
        "L_i": "Затримка обробки",
        "BW_i": "Потрібна пропускна здатність",
        "C0_i": "Сукупні початкові витрати",
        "Cop_i": "ОРЕХ на рік",
        "Csc_i": "Вартість масштабування",
        "Linc_i": "Очікуваний збиток",
        "Qfat_raw": "Індекс втоми (raw)",
        "Qfat_i": "Індекс втоми (нормований)",
        "T_i": "Оперативність реагування",
        "U_i": "Рівень автоматизації",
    })
    st.dataframe(display_df, use_container_width=True, hide_index=True)

```

```

st.markdown("### Графік топ-10 варіантів за інтегральною ефективністю")
bar = alt.Chart(top).mark_bar().encode(
    x=alt.X("variant:O", title="Варіант"),
    y=alt.Y("integrated_efficiency_penalized:Q", title="Інтегральна ефективність зі штрафом"),
    color=alt.Color("architecture:N", title="Архітектура"),
    tooltip=["variant", "architecture", "integrated_efficiency", "integrated_efficiency_penalized", "Qfat_raw", "Linc_i"],
).properties(height=360)
st.altair_chart(bar, use_container_width=True)

st.markdown("### Діаграма «вартість – ефективність»")
scatter = alt.Chart(top).mark_circle(size=120).encode(
    x=alt.X("CO_i:Q", title="Сукупні початкові витрати, грн"),
    y=alt.Y("integrated_efficiency_penalized:Q", title="Інтегральна ефективність зі штрафом"),
    color=alt.Color("architecture:N", title="Архітектура"),
    tooltip=["variant", "architecture", "CO_i", "integrated_efficiency_penalized", "A_i", "FP_i", "FN_i"],
).properties(height=360)
st.altair_chart(scatter, use_container_width=True)

st.markdown("### Детальний перегляд обраного варіанта")
select_options = df[["variant", "rank", "architecture", "integrated_efficiency_penalized"]].copy()
select_options["label"] = select_options.apply(lambda r: f"Варіант {int(r['variant'])} | ранг {int(r['rank'])} | {r['architecture']} | E={r['integrated_efficiency_penalized']:.4f}", axis=1)
selected_label = st.selectbox("Оберіть варіант для детального перегляду", options=select_options["label"].tolist(), index=0)
selected_variant = int(select_options.loc[selected_options["label"] == selected_label, "variant"].iloc[0])
selected_detail = details.get(selected_variant)
if selected_detail:
    parameter_table = build_parameter_table(selected_detail["config"], selected_detail["result"], st.session_state["constraints"])
    st.dataframe(parameter_table, use_container_width=True, hide_index=True)

csv = df.to_csv(index=False).encode("utf-8-sig")
st.download_button("Завантажити повні результати оптимізації (CSV)", data=csv, file_name="optimization_results.csv",
mime="text/csv")

def comparison_tab() -> None:
    st.subheader("Порівняння базової та оптимізованої конфігурації")
    base_config = st.session_state.get("config")
    base_result = st.session_state.get("last_result")
    df = st.session_state.get("optimization_results")
    details = st.session_state.get("optimization_details", {})

    if base_config is None or base_result is None:
        st.info("Спочатку виконайте базовий розрахунок на вкладці калькулятора.")
        return
    if df is None or df.empty:
        st.info("Спочатку виконайте оптимізацію на відповідній вкладці.")
        return

    feasible = df[df["is_feasible"]].copy()
    best_row = feasible.iloc[0] if not feasible.empty else df.iloc[0]
    best_variant = int(best_row["variant"])
    best_detail = details.get(best_variant)
    if not best_detail:
        st.warning("Не вдалося завантажити деталі оптимізованого варіанта.")
        return

    opt_config = best_detail["config"]
    opt_result = best_detail["result"]

    st.markdown("### Коротке порівняння")
    c1, c2 = st.columns(2)
    with c1:
        st.markdown("***Базова конфігурація***")
        st.metric("Архітектура", base_result["architecture_label"])
        st.metric("Інтегральна ефективність", f"{base_result['integrated_efficiency']:.4f}")

```

```

st.metric("Очікуваний збиток, грн/год", f"{base_result['raw']['Linc_i']:.0f}")
st.metric("Індекс втоми (raw)", f"{base_result['raw']['Qfat_raw']:.3f}")
with c2:
    st.markdown("***Оптимізована конфігурація***")
    st.metric("Архітектура", opt_result["architecture_label"])
    st.metric("Інтегральна ефективність", f"{opt_result['integrated_efficiency']:.4f}")
    st.metric("Очікуваний збиток, грн/год", f"{opt_result['raw']['Linc_i']:.0f}")
    st.metric("Індекс втоми (raw)", f"{opt_result['raw']['Qfat_raw']:.3f}")

st.markdown("### Порівняння локальних ефективностей")
comp_df = comparison_chart_df(base_result, opt_result)
melted = comp_df.melt(id_vars=["Показник", "var_name"], value_name="Значення")
grouped = alt.Chart(melted).mark_bar().encode(
    x=alt.X("Показник:N", title="Показник"),
    y=alt.Y("Значення:Q", title="Значення"),
    color=alt.Color("Конфігурація:N", title="Конфігурація"),
    xOffset="Конфігурація:N",
    tooltip=["Показник", "Конфігурація", "Значення"],
).properties(height=360)
st.altair_chart(grouped, use_container_width=True)

st.markdown("### Порівняння ключових параметрів")
param_compare = pd.DataFrame({
    "Параметр": [
        "Сукупні початкові витрати",
        "Експлуатаційні витрати на рік",
        "Потрібна пропускна здатність",
        "Очікуваний збиток",
        "Індекс втоми (raw)",
        "Оперативність реагування",
        "Рівень автоматизації",
        "F1-score",
    ],
    "Базова": [
        base_result["raw"]["C0_i"],
        base_result["raw"]["C0p_i"],
        base_result["raw"]["BW_i"],
        base_result["raw"]["Linc_i"],
        base_result["raw"]["Qfat_raw"],
        base_result["raw"]["T_i"],
        base_result["raw"]["U_i"],
        base_result["f1_score"],
    ],
    "Оптимізована": [
        opt_result["raw"]["C0_i"],
        opt_result["raw"]["C0p_i"],
        opt_result["raw"]["BW_i"],
        opt_result["raw"]["Linc_i"],
        opt_result["raw"]["Qfat_raw"],
        opt_result["raw"]["T_i"],
        opt_result["raw"]["U_i"],
        opt_result["f1_score"],
    ],
})
param_melted = param_compare.melt(id_vars=["Параметр"], var_name="Конфігурація", value_name="Значення")
compare_chart = alt.Chart(param_melted).mark_bar().encode(
    x=alt.X("Параметр:N", title="Параметр"),
    y=alt.Y("Значення:Q", title="Значення"),
    color=alt.Color("Конфігурація:N", title="Конфігурація"),
    xOffset="Конфігурація:N",
    tooltip=["Параметр", "Конфігурація", "Значення"],
).properties(height=380)
st.altair_chart(compare_chart, use_container_width=True)

st.markdown("### Табличне порівняння параметрів")

```

```

merged_params = pd.merge(
    build_parameter_table(base_config, base_result, st.session_state["constraints"])[["Група", "Параметр",
"Значення"]].rename(columns={"Значення": "Базова"}),
    build_parameter_table(opt_config, opt_result, st.session_state["constraints"])[["Група", "Параметр",
"Значення"]].rename(columns={"Значення": "Оптимізована"}),
    on=["Група", "Параметр"],
    how="outer",
)
st.dataframe(merged_params, use_container_width=True, hide_index=True)

def main() -> None:
    ensure_state()
    st.title("Калькулятор та оптимізація ефективності інтегрованої системи відеоспостереження")
    st.caption("Базовий розрахунок, деталізація, оптимізація, результати оптимізації та порівняння базової й оптимізованої конфігурації в одному інструменті.")

    constraints_sidebar()
    weights_sidebar()

    tab1, tab2, tab3, tab4, tab5 = st.tabs([
        "Калькулятор ефективності",
        "Деталізовані результати",
        "Оптимізація",
        "Результати оптимізації",
        "Порівняння конфігурацій",
    ])

    with tab1:
        calculator_tab()
    with tab2:
        detailed_results_tab()
    with tab3:
        optimization_tab()
    with tab4:
        optimization_results_tab()
    with tab5:
        comparison_tab()

if __name__ == "__main__":
    main()

```



Проректор з навчальної роботи та
якості освіти
Лариса ШАУЛЬСЬКА
«02» 202_ року

АКТ ВПРОВАДЖЕННЯ
результатів дисертаційного дослідження в навчальний процес
Державного некомерційного підприємства
«Державний університет «Київський авіаційний інститут»

Ми, що нижче підписалися, декан факультету комп'ютерних наук та технологій, к.т.н., доцент Фесенко А.О., завідувач кафедри інтелектуальних кібернетичних систем, д.т.н., професор Нечипорук О.П. склали даний акт про те, що результати дисертаційної роботи Певнева Владислава Ігоровича "Методи оптимізації інтегрованої системи відеоспостереження з елементами аналізу даних" використовуються у навчальному процесі Державного некомерційного підприємства «Державний університет «Київський авіаційний інститут», факультету комп'ютерних наук та технологій на кафедрі інтелектуальних кібернетичних систем.

Вид результату, що впроваджується	Форма впровадження	Ефект від впровадження
1. Модель багатокритеріального оцінювання інтегральної ефективності систем відеоспостереження, що враховує взаємозв'язок між точністю нейромережевої аналітики, очікуваними фінансовими збитками від інцидентів та когнітивною втомою оператора. 2. Метод вибору раціональної конфігурації апаратно-програмних засобів відеоспостереження на основі інтегрального комплексного підходу та згортки різнорозмірних техніко-економічних критеріїв. 3. Комплексний метод управління апаратними та обчислювальними ресурсами розподілених систем відеоспостереження, що забезпечує адаптивний баланс між якістю детектування об'єктів, навантаженням на мережеву інфраструктуру та сукупними витратами (CAPEX/OPEX).	У навчальних програмах дисциплін: – "Методи машинного навчання" для спеціальності F6 Інформаційні системи і технології ОПП "Інтелектуальні системи та технології" № НМ-4-F6/26-2.1/5, що затверджена 01.09.2025 р.; – "Імітаційне моделювання", для спеціальності 123 Комп'ютерна інженерія ОПП "Системне програмування" № НБ-4-123-2/23-2.1.18, що затверджена 13.09.2024 р. У кваліфікаційних роботах.	Підвищення якості навчання. Інтенсифікація навчального процесу.

Декан факультету комп'ютерних
наук та технологій

Завідувач кафедри інтелектуальних
кібернетичних систем

Андрій ФЕСЕНКО

Олена НЕЧИПОРУК

АКТ

впровадження результатів дисертаційного дослідження
здобувача ступеня доктора філософії за спеціальністю 172 «Телекомунікації та радіотехніка»
ПЕВНЕВА Владислава Ігоровича
на тему «Методи оптимізації інтегрованої системи відеоспостереження з елементами аналізу
даних»

Комісія у складі директора Курінного Олександра Васильовича, розробника комп'ютерних програм Посохова Валерія Володимировича та інженера з налагодження й випробувань Бойчука Владислава Олеговича підтверджує, що результати дисертаційної роботи Певнева Владислава Ігоровича використовуються в діяльності компанії ТОВ «Аксонсофт», а саме:

Запропонована автором математична модель інтегральної оцінки ефективності інтегрованої системи відеоспостереження використана для аналізу базових конфігурацій систем відеонагляду та вибору раціональних архітектурних рішень з урахуванням технічних, економічних і функціональних показників. Проведені розрахунки підтвердили можливість підвищення інтегральної ефективності системи та покращення обґрунтованості вибору конфігурації для об'єктів з підвищеними вимогами до безпеки.

На тестових і проектних конфігураціях компанії проведено апробацію розробленого методу оптимізації інтегрованої системи відеоспостереження з елементами аналізу даних. Підтверджено працездатність підходу до автоматизованого розрахунку технічної, економічної та функціональної ефективності, а також можливість ранжування альтернативних конфігурацій і вибору найбільш раціонального варіанта в межах заданих обмежень.

Розроблений автором програмний прототип для оцінювання та оптимізації конфігурацій систем відеоспостереження прийнятий до використання як інструмент підтримки прийняття рішень у процесі проектування, модернізації та порівняльного аналізу архітектур систем відеонагляду.

Рекомендації автора щодо комплексного врахування впливу модулів відеоаналітики, очікуваних втрат від пропущених інцидентів, витрат на експлуатацію та навантаження на операторів прийняті технічним підрозділом компанії для використання в процесі підготовки проектних рішень у 2026–2027 роках.

Висновок: результати дисертаційної роботи оцінюються як такі, що мають практичну цінність та можуть бути використані для підвищення ефективності проектування, оцінювання й оптимізації інтегрованих систем відеоспостереження з елементами аналізу даних.

Посохов В.В. розробник комп'ютерних програм
Бойчук В.О. інженер з налагодження й випробувань

З повагою,
Директор



О.В. Курінний