МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кваліфікаційна наукова
праця на правах рукопису

Жовнір Юрій Іванович

УДК 004.056:004.45:728.2

ДИСЕРТАЦІЯ

МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНОЇ ПРОГРАМНОЇ СИСТЕМИ БЕЗПЕКИ ЖИТЛОВИХ КОМПЛЕКСІВ

01.05.03 - математичне та програмне забезпечення
обчислювальних машин і систем
05 - технічні науки

Подається на здобуття наукового ступеня кандидата технічних наук. Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело


Ю. І. Жовнір

Науковий керівник
Пасічник Володимир Володимирович,
Доктор технічних наук, професор

Ідентичність усіх примірників дисертації
ЗАСВІДЧУЮ:
Учений секретар спеціалізованої вченої ради
/Ростислав БУНЬ/

Львів – 2025

АНОТАЦІЯ

Жовнір Ю. І. Методи та засоби побудови інтелектуальної програмної системи безпеки житлових комплексів. — Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 01.05.03 – математичне та програмне забезпечення обчислювальних машин і систем (Технічні науки). — Національний університет «Львівська політехніка» МОН України, Львів, 2025.

У сучасних умовах швидкого розвитку технологій та зростаючих вимог до безпеки житлових комплексів, інтелектуальні програмні системи безпеки набувають особливої актуальності. Дисертаційна робота присвячена дослідженню методів і засобів побудови інтелектуальної програмної системи безпеки житлових комплексів, що використовує підходи ситуаційної обізнаності та технології інтернету речей.

Розглянуто основні підходи до побудови інтелектуальних систем безпеки, аналіз архітектурних рішень для розумних будинків, методи керування безпекою та правами доступу, а також інструменти DevOps у системах на базі IoT. Проведено порівняльний аналіз існуючих програмно-апаратних засобів, що дозволило визначити основні вимоги до ефективної інтелектуальної системи безпеки. Крім того, вивчено вплив архітектурних рішень на продуктивність і масштабованість системи, що дає можливість її адаптації під різні категорії житлових комплексів.

Проаналізовано побудову проблемно-орієнтованої онтології інтелектуальної програмної системи безпеки з ситуаційною обізнаністю. Запропоновано онтологічні моделі для міркувань про передбачувані ситуації, моделювання змін між ситуаціями, виконання дій та навчання на основі зворотного зв'язку. Розроблено доменно-орієнтовану онтологію системи безпеки житлового комплексу, а також досліджено відповідні метрики для її оцінювання. Особлива увага приділяється використанню ситуоїдів для подання та моделювання динаміки змін ситуацій, що підвищує ефективність розпізнавання загроз і реагування на них.

Представлено структуру та архітектуру інтелектуальної програмної системи безпеки. Запропоновано модель компонентів, що забезпечує інтеграцію системи з різними пристроями та сервісами IoT. Детально розглянуто функціональні особливості підсистем, зокрема управління доступом, відеоспостереження, аналізу поведінки та інтелектуального реагування на загрози. Досліджено алгоритми обробки даних з різних сенсорів, що дозволяє підвищити рівень автоматизації системи та скоротити час реагування на загрози.

Подано результати практичної реалізації інтелектуальної системи безпеки "АСТРА. Безпечний ЖК", концептуальну модель, розглянуто інструменти розроблення бекенду, а також методологію створення інтерфейсів на основі методу персон. Особливу увагу приділено класифікації користувачів за допомогою методу випадкового лісу, що дозволяє персоналізувати послуги та підвищити ефективність взаємодії користувачів із системою безпеки. Проаналізовано сценарії використання системи, включно з автоматичним розпізнаванням небезпечних ситуацій та алгоритмами реагування на основі машинного навчання.

Зроблено висновки щодо ефективності розробленої системи та визначено напрями подальших досліджень у цій сфері. Запропоновані методи та технології можуть бути використані для розробки та вдосконалення інтелектуальних програмних систем безпеки житлових комплексів та інших об'єктів критичної інфраструктури. Особливу увагу приділено перспективам подальшої інтеграції з іншими інтелектуальними системами, а також використанню хмарних технологій для підвищення масштабованості та безпеки опрацювання даних.

Ключові слова: інтелектуальна програмна система безпеки, онтологія, загальна формальна онтологія (GFO), мова веб-онтології, структура опису ресурсів, мова шаблонів, ситуаційна обізнаність, просторова та часова динаміка, житловий комплекс, аналіз ситуації, ситуоїд, обернено-адитивна метрика, метод персон, метод випадкового лісу, інтернет речей, DevOps.

ANNOTATION

Zhovnir Yuriy Ivanovych. Methods and Tools for Developing an Intelligent Software Security System for Residential Complexes. - On the rights of the manuscript. Dissertation for obtaining the scientific degree of candidate of technical sciences in the specialty 01.05.03 - mathematical and software support of computing machines and systems (technical sciences). - Lviv Polytechnic National University, Ministry of Education and Science of Ukraine, Lviv, 2025.

In the modern era of rapid technological development and increasing security requirements for residential complexes, intelligent security software systems are gaining particular relevance. This dissertation is dedicated to the study of methods and tools for building an intelligent security software system for residential complexes, utilizing situational awareness approaches and Internet of Things (IoT) technologies.

The research examines key approaches to constructing intelligent security systems, analyzes architectural solutions for smart homes, explores security management and access control methods, and investigates DevOps tools in IoT-based systems. A comparative analysis of existing software and hardware solutions has been conducted, enabling the identification of key requirements for an effective intelligent security system. Additionally, the impact of architectural decisions on system performance and scalability has been studied, facilitating adaptation to various categories of residential complexes.

The dissertation delves into the construction of a problem-oriented ontology for an intelligent security software system with situational awareness. Ontological models are proposed for reasoning about anticipated situations, modeling changes between situations, executing actions, and learning based on feedback. An ontology for a residential complex security system has been developed, along with relevant metrics for evaluating its efficiency. Special attention is given to the use of situoids for representing and modeling the dynamics of situational changes, enhancing the effectiveness of threat recognition and response.

The structure and architecture of the intelligent security software system are presented. A component model is proposed that ensures system integration with

various IoT devices and services. The functional characteristics of subsystems are examined in detail, including access control, video surveillance, behavior analysis, and intelligent threat response. The research also explores data processing algorithms from various sensors, which improve system automation levels and reduce response times to security threats.

The practical implementation of the intelligent security system "ASTRA. Safe Residential Complex" is presented, along with its conceptual model. Development tools for the system's backend are reviewed, as well as the methodology for designing interfaces based on the persona method. Special attention is paid to user classification using the Random Forest method, which enables service personalization and enhances user interaction with the security system. Various usage scenarios of the system are described, including automatic recognition of hazardous situations and response algorithms based on machine learning.

Conclusions are drawn regarding the efficiency of the developed system, and future research directions in this field are outlined. The proposed methods and technologies can be applied to the development and improvement of intelligent security software systems for residential complexes and other critical infrastructure facilities. Particular attention is given to further integration with other intelligent systems, as well as the use of cloud technologies to enhance scalability and data processing security.

Keywords: intelligent software security system, ontology, General Formal Ontology (GFO), Web Ontology Language, Resource Description Framework, template language, situational awareness, spatial and temporal dynamics, residential community, situational awareness, situation analysis, situoid, reverse-additive metric, Persona method, Random Forest method, Internet of Things (IoT), DevOps.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА:

Статті, опубліковані у періодичних виданнях, індексованих міжнародною наукометричною базою даних Scopus

1.Zhovnir Y., Kunanets N., Burov Y., Duda O., Pasichnyk V. Development of the structure and architecture of situational awareness security information systems for

residential complexes Eastern-European Journal of Enterprise Technologies, 2025, №1(133). P.63-98.

2. Vladov S., Avkurova Zh., Lytvyn V., Zhovnir Yu. Analytical Neural Network System for the Helicopter Turboshaft Engines Operating Modes Classification. International Journal of Computing. 2024. Vol. 23, No. 3. P. 342–359. https://doi.org/10.31891/csit-2024-3-10 .

Статті, опубліковані у періодичних виданнях, які входять до переліку наукових фахових видань України

3. Жовнір Ю., Буров Є. Еволюція архітектурних рішень для розумних будинків. Computer Systems and Information Technologies, 2024, Вип.3, С.74–85. https://doi.org/10.31891/csit-2024-3-10

4. Григорович А., Григорович В., Жовнір Ю., Грибовський О. Формування обернено-адитивної семантичної метрики для аналізу онтологій безпекових систем багатоквартирних будинків. Комп'ютерно-інтегровані технології: освіта, наука, виробництво, 2024, Вип. 56, С.12-30. https://doi.org/10.36910/6775-2524-0560-2024-56-02

5. Жовнір Ю., Грибовський О. Порівняльний аналіз програмно-апаратних інструментальних засобів для створення безпекової системи багатоквартирного будинку. Herald of Khmelnytskyi National University. Technical Sciences, 2024, Вип.339(4), С.344-358. https://doi.org/10.31891/2307-5732-2024-339-4-54

6. Жовнір Ю., Грибовський О., Пасічник С., Бобик І. Створення інтерфейсів безпекових систем багатоквартирних будинків з використанням методу Персон Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі, 2024, Випуск 16, С. 145 – 166.- https://doi.org/10.23939/sisn2024.16.145

7. Burov, E., Zhovnir, Y., Zakhariya, O. The vision and implementation of intelligent security system. Herald of Khmelnytskyi National University. Technical Sciences, 2024, 341(5), 497-509. https://doi.org/10.31891/2307-5732-2024-341-5-72

8. Цейтлін Г.Е., Захарія Л.М., Захарія О.В., Жовнір Ю.І Екологічні аспекти подання знань засобами алгебри алгоритміки Проблеми програмування. 2010. № 2–3. Спеціальний випуск, С.369-375.

9. Burov Y., Zhovnir Y., Zakharia O. Designing the ontology for intelligent security system of residential community Scientific journal of the Ternopil Ivan Puluj National Technical University, 2024, vol 116, no 4, pp. 111-124. https://doi.org/10.33108/visnyk_tntu2024.04.111.

10. Кунанець Н., Жовнір Ю., Веремеєнко А., Пущак С. Концептуальне моделювання системи відеоспостереження з ситуаційною обізнаністю Herald of Khmelnytskyi National University. Technical Sciences, 2025, №1. С.189-202.

11. Жовнір Ю. І., Грибовський О. М., Орлов М. В., Дуда О. М., Кунанець Н. Е. Методологія розроблення та супроводу інформаційних систем, базованих на технології інтернету речей Управління розвитком складних систем 2024.- Вип.60, С. 56-71. https://doi.org/10.32347/2412-9933.2024.60.56-70

12. Орлов М. В., Дуда О. М., Жовнір Ю. І., Грибовський О. М. Інструменти методології DevOps в інформаційних системах на основі технологій IoT

Комп'ютерно-інтегровані технології: освіта, наука, виробництво, 2024, Вип. 57.-С.128-139. https://doi.org/10.36910/6775-2524-0560-2024-57-15

13. Орлов М. В., Грибовський О.М., Жовнір Ю.І., Дуда О.М. Від концепції до реальності: роль DevOps в екосистемах IoT Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки, 2024, Том 35 (74) № 6, Ч.2.-С.157-164.

Праці та тези доповідей у збірниках матеріалів конференцій

1.Vaskiv R., Veretennikova N., Nebesnyi R., Bilovus H., Zhovnir Y. Formation of an IT Project Team by Analogy with a Flock IEEE 19th International Conference on Computer Science and Information Technologies (CSIT), 2024

2. Жовнір Ю. І., Кунанець Н. Е., Захарія О. В. Вимоги до інформаційних систем безпеки житлового кварталуProceedings the 5th International scientific and practical conference "Current trends in scientific research development", (December 12-14, 2024). Boston, .2024, P. 298–303. URL: https://sci-conf.com.ua/v-mizhnarodna-naukovo-praktichna-konferentsiya-current-trendsin-scientific-research-development-12-14-12-2024-boston-ssha-arhiv/.

3. Жовнір Ю. І., Кунанець Н. Е., Захарія О. В., Орлов М. В. Використання методологій devops та devsecops у IT проєктах Proceedings the 4th International scientific and practical conference "Science in the modern world: innovations and challenges", (December 19-21, 2024) Toronto, 2024, p.228-233URL: https://sci-conf.com.ua/iv-mizhnarodna-naukovo-praktichna-konferentsiya-sciencein-the-modern-world-innovations-and-challenges-19-21-12-2024-toronto-kanada-arhiv/.

4. Жовнір Ю. І., Кунанець Н. Е., Захарія О. В., Пасічник С. О. Формування бекенду та фронтенду інформаційної системи безпеки з ситуаційною обізнаністю Proceedings I International scientific and practical conference «European congress of scientific discovery» (December 29-31, 2024). Madrid, 2024, С.244-252. https://sci-conf.com.ua/wp-content/uploads/2024/12/EUROPEAN-CONGRESS-OF-SCIENTIFIC-DISCOVERY-29-31.12.2024.pdf

5. Жовнір Ю. І., Кунанець Н. Е., Захарія О. В., Орлов М. В. Планування та прогнозування ситуації в інтелектуальній інформаційній системі безпеки Proceedings II International scientific and practical conference Future of science: innovations and perspectives, (December 23-25, 2024). Stockholm, 2024, С.163-169. https://sci-conf.com.ua/wp-content/uploads/2024/12/FUTURE-OF-SCIENCE-INNOVATIONS-AND-PERSPECTIVES-23-25.12.24.pdf

6. Жовнір Ю., Захарія Л., Захарія Ю. Формалізація та породження знань засобами алгебри алгоритміки Комп'ютерні науки та інженерія: матеріали наукової конференції молодих вчених, 25-27 листопада 2010 р., Львів, Львів, 2010, С. 118-120

7. Жовнір Ю., Ваків М., Захарія Л. Віртуальна аудиторія як система електронного навчання інвалідів Комп'ютерні науки та інженерія: матеріали наукової конференції молодих вчених, 25-27 листопада 2010 р., Львів. Львів, 2010.- С.126-128.

ЗМІСТ

Вступ

Актуальність теми. Враховуючи виклики сучасності, дослідження в галузі інтелектуальних програмних систем безпеки має високу актуальність. В межах даного дослідження основна увага зосереджена на гарантуванні безпеки в контексті побудови інтелектуальних систем для потреб мешканців багатоквартирних будинків та житлових комплексів. З огляду на те, що житлові комплекси стають дедалі масштабнішими та популярнішими, постає питання розроблення та впровадження високотехнологічних інтелектуальних програмних систем управління безпекою та підвищення їх ефективності.

Зростання загроз для житлових комплексів включає кібератаки на IoT-пристрої, що можуть призвести до порушення приватності мешканців, фінансових втрат та фізичної небезпеки, компрометацію персональних даних мешканців, а також атаки типу DDoS, які можуть вивести з ладу критичні системи безпеки. Житлові комплекси є складними системами, де фізична безпека повинна бути інтегрована з інформаційними технологіями, що вимагає створення захищених каналів зв'язку, надійного управління доступом і постійного моніторингу кіберзагроз. Сучасні інтелектуальні безпекові системи повинні мати можливість прогнозувати загрози за допомогою прогностичних моделей та реагувати на них до того, як вони стануть критичними.

Соціальна значущість теми полягає у забезпеченні довіри мешканців до новітніх технологій, захисті їхньої приватності та створенні комфортного і безпечного середовища для життя. Ефективні методи підтримання належного рівня безпеки також дозволяють запобігти потенційним фінансовим втратам і зменшити витрати на реагування на інциденти. Таким чином, розроблення та впровадження методів і засобів побудови інтелектуальної програмної системи безпеки житлових комплексів є актуальною та життєво важливою задачею.

Використання інтелектуальних сенсорів, камер спостереження, зчитувачів та інших IoT пристроїв з мережевими інтерфейсами дає змогу інтегрувати їх у

єдиній інтелектуальній системі безпеки. Розроблення такої програмної системи безпеки з урахуванням ситуаційної обізнаності присвячене дисертаційне дослідження. В даній роботі запропоновано використовувати інфраструктуру інтернет-провайдерів (в даному випадку йдеться про Львівського регіонального інтернет провайдера - компанію АСТРА-ЛЬВІВ), зокрема, існуючу кабельну мережу, серверні кластери, інструменти методології DevOps, бази даних користувачів, персоналу для розгортання та супроводу інтелектуальної програмної системи безпеки житлового комплексу.

Задача створення інтелектуальної програмної системи безпеки житлових комплексів залишається актуальною через фрагментарність рішень, недостатню інтеграцію та адаптивність існуючих систем. Для її вирішення необхідно впроваджувати методи та засоби інтеграції фізичної та інформаційної безпеки, використовувати інструменти штучного інтелекту і застосовувати проактивні методи захисту. Таким чином, дослідження за цією тематикою має як науково-методичне, так і практично-прикладне значення для комерційного впровадження та широкого використання.

Зв'язок роботи з науковими програмами, планами, темами. Задачі, що вирішуються у дисертаційній роботі, випливають із завдань у сфері науки і техніки, сформульованих у Законі України № 2519-VI від 09.09.2010 р. «Про внесення змін до Закону України «Про пріоритетні напрямки розвитку науки і техніки», також у Постанові Кабінету міністрів України від 30 квітня 2024 р. за № 476 «Про затвердження переліку пріоритетних тематичних напрямів наукових досліджень і науково-технічних розробок на період до 31 грудня року, наступного після припинення або скасування воєнного стану в Україні». Результати наукових досліджень і практичних напрацювань, що наведені в дисертації, тісно пов’язані з напрямами науково-технічної діяльності кафедри інформаційних систем та мереж Національного університету «Львівська політехніка» – «Розроблення інтелектуальних агентів пошуку релевантної інформації в мережі Інтернет та її опрацювання з метою автоматичного наповнення баз даних та баз знань», «Розроблення системи підтримки прийняття рішень на основі онтологічних

знань та технологій машинного навчання».

Мета і задачі дослідження. Метою дисертаційної роботи є розроблення нових та вдосконалення існуючих методів та засобів побудови та супроводу інтелектуальних програмних систем безпеки для житлових комплексів і формування високотехнологічного, безпечного та комфортного середовища проживання.

Завдання дослідження:

- Провести аналіз підходів, методів та засобів для побудови інтелектуальних програмних систем безпеки багатоквартирних будинків та житлових комплексів.
- Сформувати комплекс методів для реалізації процесів розроблення та супроводу інтелектуальних програмних систем безпеки на основі технологій IoT.
- Спроектувати доменно-орієнтовану онтологію для програмної системи безпеки з ситуаційною обізнаністю на основі конструктивів базової чотиривимірної онтології GFO та сформувати обернено-адитивну метрику для її оцінювання.
- Сформувати структуру та архітектуру інтелектуальної програмної системи безпеки з ситуаційною обізнаністю для житлових комплексів та розробити її концептуальну модель.
- Розробити процедури побудови інтерфейсів інтелектуальних програмних систем безпеки житлового комплексу, системно поєднавши методи персон та випадкового лісу.
- Розробити програмну систему безпеки з ситуаційною обізнаністю житлового комплексу «АСТРА. Безпечний ЖК» та провести її тестові випробування .

Об'єктом дослідження є проєктування прикладних програмних систем гарантування безпеки мешканців та майна в житловому комплексі.

Предметом дослідження є методи та засоби розроблення інтегрованих програмних систем безпеки з ситуаційною обізнаністю для житлових комплексів.

Методи дослідження. Для вирішення поставлених у дисертаційній роботі

завдань використано такі методи дослідження: метод аналізу ієрархій - для визначення кращих програмно-апаратних платформ на основі технологій IoT та формування комплексів програмних інструментів методології розроблення і супроводу програмної системи безпеки; SWOT аналіз сприяв обранню методології управління ІТ проектами; методи онтологічного моделювання для проектування бази знань інтелектуальної системи безпеки; методи та засоби штучного інтелекту для створення програмної системи безпеки з ситуаційною обізнаністю; методи теорії графів при побудові обернено-адитивної метрики для оцінювання онтологічної близькості; методи структурного аналізу та концептуального моделювання для побудови структури та архітектури прикладної програмної системи безпеки з ситуаційною обізнаністю для житлових комплексів; метод персон в поєднанні з методом випадкового лісу для формування користувацьких інтерфейсів; методи об'єктно-орієнтованого програмування та системного аналізу для розроблення інтелектуальної програмної системи безпеки житлових комплексів.

Наукова новизна. Наукова новизна роботи полягає у розв'язанні важливої наукової задачі побудови інтелектуальної програмної системи безпеки житлових комплексів. В результаті розв'язання цієї задачі одержані такі наукові результати.

Вперше:

- Сформовано модель бази знань інтелектуальної програмної системи безпеки житлових комплексів у вигляді доменно-орієнтованої онтології, яка враховує просторово-часові, структурні і поведінкові аспекти взаємодії компонентів системи, та побудовано обернено-адитивну метрику для її оцінювання, зокрема ступеню зв'язності її елементів, що в сукупності підвищило ефективність аналізу критичних взаємозв'язків та структурної організації компонентів, а також адаптивність системи до потенційних змін середовища.

Удосконалено

- Методи прогнозування розвитку подій у інтелектуальних безпекових системах, які базуються на типових сценаріях і моделях поведінки, сформованих

як результати аналізу історичних даних ситуоїдів, що дало змогу підвищити точність і оперативність прогнозування потенційних загроз, забезпечити своєчасне виявлення аномальних ситуацій та підвищити адаптивність системи до нових викликів завдяки гнучкому врахуванню змін у поведінкових патернах.

Отримало подальший розвиток:

- Метод аналізу ієрархій - для визначення кращих програмно-апаратних платформ на основі технологій інтернету речей та формування комп-лексів програмних інструментів методології розроблення та супроводу програм-ної системи безпеки, що дало змогу обґрунтовано вибирати оптимальні програмно-апаратні платформи з урахуванням ключових критеріїв ефективності, надійності та масштабованості, а також забезпечити цілісність і узгодженість програмних інструментів для розроблення та супроводу інтелектуальних систем безпеки.
- Системне поєднання методів персон та випадкового лісу у процесах побудови ефективних інтерфейсів інтелектуальних програмних систем, яке дозволяє враховувати індивідуальні характеристики і поведінкові особливості груп користувачів, та дало змогу підвищити рівень персоналізації інтерфейсів, забезпечити адаптивність систем до різних сценаріїв користувацької взаємодії та покращити загальний досвід користувачів шляхом оптимізації навігації, зручності використання та швидкості доступу до ключових функцій системи безпеки.

Практичне значення одержаних результатів. Проаналізовано та використано прогресивні підходи для розроблення і впровадження інтелектуальних систем безпеки на основі існуючої інфраструктури інтернет провайдера. Розроблені та реалізовані програмні компоненти, що можуть бути розширені і вдосконалені в наступних версіях системи безпеки, а також можуть адаптуватися до нових сценаріїв і реалізацій. Сформовано методологічний підхід, що системно реалізує метод персон та випадкового лісу для побудови інтерфейсів системи безпеки з урахуванням результатів аналізу складу мешканців та працівників житлових комплексів. Розроблено і апробовано методи розпізнавання загроз безпеці і потенційні інструменти з відповідними процедурами прийняття рішень згідно сценаріїв безпеки та можливістю навчання системи на основі бази

знань. Одержані результати успішно імплементовано в програмній системі “АСТРА. Безпечний ЖК”, що перебуває на етапі впровадження та дослідної експлуатації. Результати дисертаційного дослідження впроваджені в навчальний процес підготовки ІТ фахівців у ряді провідних ЗВО України.

Особистий внесок здобувача. Усі наукові результати, викладені в дисертації, отримані здобувачем особисто. У працях, опублікованих у співавторстві, здобувачеві належать: розроблення структури та архітектури інтелектуальної системи безпеки [120]; розроблення підходів до створення модуля, який розраховує вагові коефіцієнти для кожного моменту часу, визначаючи його важливість для поточного завдання [126]; аналіз архітектурних рішень для інтелектуальних будинків, проаналізовані останні розробки в сфері безпеки та управління правами доступу для інтелектуальних будинків [72]; проведення аналізу та обґрунтування доцільності використання обернено-адитивної метрики для онтологій інтелектуальних систем безпеки житлових комплексів [117]; вибір програмно-апаратних засобів для створення інтелектуальних систем безпеки з метою подальшого аналізу методом аналізу ієрархій, визначення основних критеріїв для проведення аналізу, таких як: гнучкість, вартість, легкість використання та підтримка різнотипових пристроїв і протоколів [89]; програмне та інформаційне забезпечення для застосування алгоритмів випадкового лісу та методу персон [128]; визначення функціональних областей інтелектуальної системи безпеки житлового комплексу, проектування концепції підсистеми відеоспостереження [125]; розроблення ряду базових програмно-алгоритмічних конструкцій моделюючого комплексу “Мультипроцесист”, в якому реалізовано основні базові принципи системи алгоритмічних алгебр з використанням принципів клонування знань [119]; розроблення безпекових сценаріїв для побудови доменно-орієнтованої онтології, опис онтології мовою OWL [104]; побудова концептуальної моделі модуля відеоспостереження з використанням UML діаграм [124]; аналіз інструментарію методології DevOps, розроблення сценаріїв застосування DevOps в IT-інфраструктурі, зокрема опрацювання даних на периферійних та хмарних

платформах, автоматизація управління інфраструктурою та забезпечення кібербезпеки [90], запропоновано метод створення комплексів інструментів та обрання кращого набору для реалізації методології DevOps в інформаційних системах[99]; проаналізовано роль DevOps в екосистемах IoT [92].

Апробація результатів дисертації. Результати дослідження доповідались і обговорювались на міжнародних наукових та науково-практичних конференціях: IEEE 19th International Conference on Computer Science and Information Technologies (CSIT-2024), (16-19 жовтня 2024, Львів, Україна)(Scopus), та на науковій конференції молодих вчених (25-27 листопада 2010, Львів), а також на наукових семінарах кафедри інформаційних систем та мереж НУ "Львівська політехніка", V міжнародній науково-практичній конференції «Current trends in scientific research development» (12-14.12.2024 року, Бостон, США), 4 міжнародній науково-практичній конференції «Science in the modern world: innovations and challenges » 19-21 грудня 2024, Торонто, Канада), I Міжнародній науково-практичній конференції «European congress of scientific discovery» (29-31 грудня 2024, Мадрид, Іспанія), II Міжнародній науково-практичній конференції «Future of science: innovations and perspectives» (23-25 грудня 2024, Стокгольмі, Швеція).

Публікації. За результатами дисертаційного дослідження опубліковано 20 друкованих праць, з них 11 у виданнях, що включені МОН України до переліку фахових та 2 у журналах, що індексується у наукометричній базі Scopus, 7 тез доповідей у матеріалах наукових та науково-практичних конференцій.

Структура роботи. Дисертаційна робота загальним обсягом 202 сторінки складається зі вступу, п'яти розділів, висновків, списку використаних джерел із 129 найменувань і додатків. Основний текст роботи викладено на 153 сторінках.

Розділ 1. Аналіз підходів до побудови інтелектуальних програмних систем безпеки у житлових громадах: інструменти та методології втілення

1.1 Аналіз архітектурних рішень для розумного будинку

Розвиток комп'ютерної техніки та інформаційних технологій, в результаті якого зросли обчислювальні потужності і знизилася вартість обчислень, знайшов своє відображення в еволюції архітектури інформаційних систем [1]. Поява персональних комп'ютерів, виникнення Інтернету і мобільних пристроїв спричинили зростання кількості комп'ютерів, якими користується один користувач [2,3]. У США до 2023 року середня кількість підключених пристроїв на одне домогосподарство досягла 17 [4]. Паралельно зі зростанням використання комп'ютерних технологій, з розвитком Інтернету речей (IoT) спостерігається значне зростання потоків обміну інформацією M2M (machine to machine). Поширення пристроїв, пов'язаних з мережами IoT, і останні досягнення в галузі штучного інтелекту (ШІ) створюють можливість для реалізації ідеї формування розумного середовища для проживання людей, яке б підтримувало і забезпечувало їм повноцінне життя. Розглядається [5] два підходи до розуміння сприяння та забезпечення добробуту мешканців. Інтелектуальне середовища базується на принципах людиноцентричного проектування, що передбачає аналіз поведінки користувачів, їхніх потреб, уподобань, формуванні доступності та інклюзивності [6]. Це передбачає розроблення зручних у використанні інтерфейсів і забезпечення декількох режимів взаємодії (наприклад, за допомогою голосу, дотику, жесту) [7]. Адаптивність та персоналізація змушує систему підлаштовуватися під мінливі потреби та вподобання користувачів, досягаючи при цьому високого рівня персоналізації [8]. Плавна інтеграція передбачає, що технологія повинна легко інтегруватися в домашнє середовище, мінімізуючи нав'язливість і при цьому гарантуючи, що взаємодія користувача з системою є комфортною і природньою [9]. Конфіденційність та безпека набувають першочергового значення [10]. Людиноцентричні системи в розумних

будинках реалізують функції, які спрямовані на підвищення комфорту, зручності та ефективності.

Доцільно базувати функції розумного дому на основі вимог Міжнародної класифікації функціонування, інвалідності та здоров'я (МКФ) Всесвітньої організації охорони здоров'я (ICF), в якій подані всі базові функції та види людської діяльності [5]. На рис. 1.1 показані взаємозв'язки між групами факторів з моделі МКФ, використаної як основи концептуалізації інтелектуального будинку.

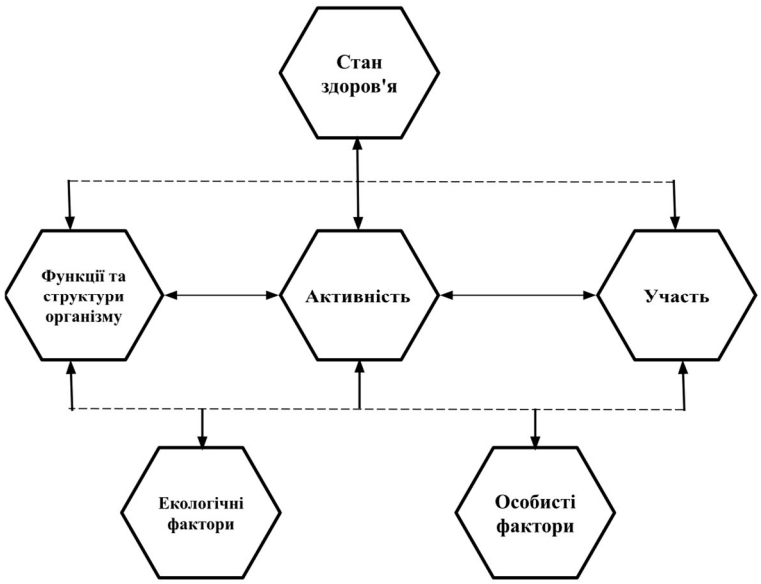


Рис. 1.1 Використання моделі МКФ для концептуалізації інтелектуального будинку

Розумний дім повинен самостійно визначати ситуації для допомоги мешканцям, пропонуючи послуги без вимоги наявності у них відповідних технічних знань [11]. Його функціональні області охоплюють моніторинг, міркування, моделювання та безпеку, що підкреслює високу значимість систем прийняття рішень [12]. Концепція соціального Інтернету речей (SIoT) інтегрує IoT-пристрої із соціальними мережами для інтелектуальної взаємодії [13].

Функціональна декомпозиція розумного будинку охоплює управління енергією, безпекою, охороною здоров'я та довкіллям [14]. Проведений аналіз еволюції архітектурних рішень дозволив системно розкрити виклики, що виникають при їх впровадженні та відповідні технологічні інновації. Ранні

системи домашньої автоматизації зосереджувалися на передачі даних у локальних мережах, що вимагало вибору відповідних технологій [15–17].

Інформаційні системи розумного будинку зазвичай мають багаторівневу архітектуру: фізичний шар (давачі та виконавчі механізми) забезпечує збір даних [18], проміжний рівень їх опрацьовує, а прикладний рівень - використання у взаємодії з користувачами. Деякі системи обмежуються застосунками для смартфонів і чат-ботами [19]. Загальна тенденція – підвищення інтелектуальності, розширення функцій та збільшення обсягів аналізованих даних. Архітектура інформаційної системи розумного будинку, в основі якої лежить сервісно-орієнтований підхід з динамічними сервісними композиціями [20], що генеруються з використанням методів та засобів штучного інтелекту, подана на рис.1.2.**Помилка! Джерело посилання не знайдено.** Такий підхід підвищує зручність і рівень безпеки для мешканців.

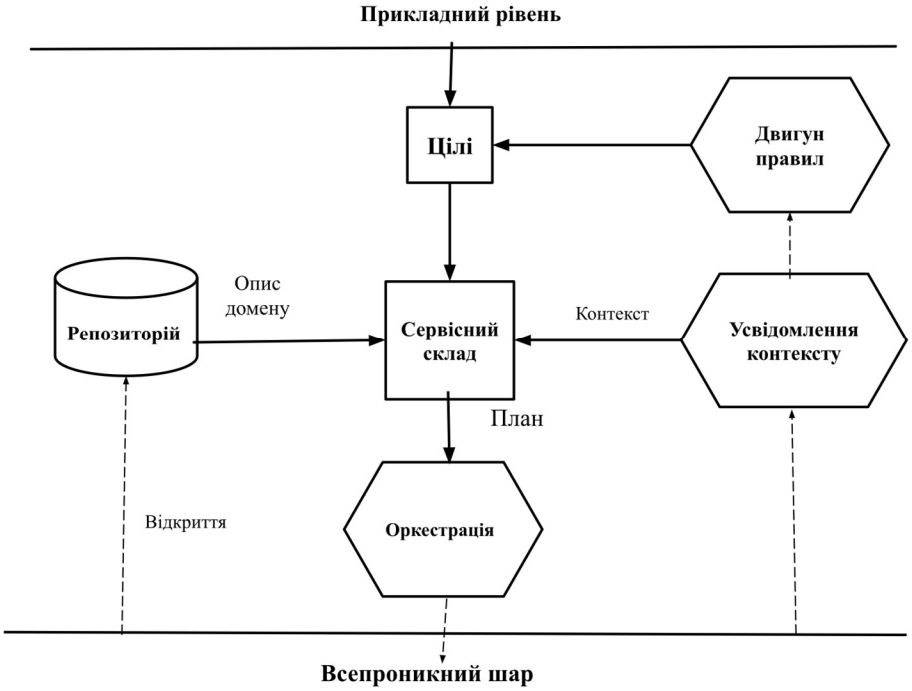


Рис.1.2 Архітектура інформаційної системи «розумний будинок» на основі композиції послуг

Архітектура інформаційної системи «розумний будинок», запропонована в роботі [20], має типові сенсорні і прикладні шари. Проте велике різноманіття потреб користувачів, мінливих ситуацій і варіабельність домашнього IT

середовища генерує ряд складностей при побудові інформаційної системи інтелектуального будинку з використанням сервісно-орієнтованого підходу. Дослідники пропонують архітектуру інтелектуальної інформаційної систему, побудованої на основі правил для вибору повного набору послуг з урахуванням актуального контексту. В сучасних дослідженнях [21] акцентується увага на реалізації інтелектуальної інформаційної системи управління агентами з урахуванням ієрархії цілей, запропонованої Гамільтоном [22, 23]. У цій ієрархії мета високого рівня може бути розкладена на завдання, безпосередній намір, безпосередню мету, дії і рухи. У запропонованій архітектурі кожна задача і мета реалізуються з використанням моделей середовищ, що включають відповідні елементи і їх взаємозв'язки. На кожному рівні зміни моделі плануються за допомогою середовищ і на останньому рівні переводяться в послідовність дій.

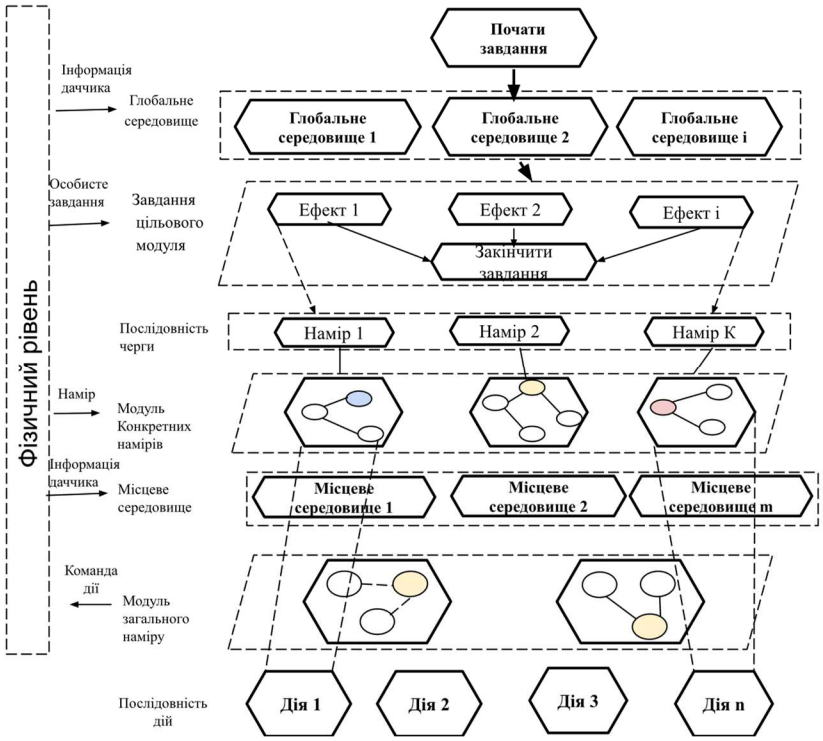


Рис. 1.3 Архітектура інформаційної системи, заснована на моделі ієрархічних цілей [19]

Ще одне архітектурне рішення інформаційної системи розумного дому базується на моделюванні (MBSE) та мультиагентних системах (MAS) [24], забезпечуючи інтелектуальну та самооптимізуючу поведінку пристроїв. Їхня

взаємодія в мережі координується для оптимального розподілу ресурсів. Кожен розумний пристрій має трирівневу архітектуру: нижній рівень – некогнітивна регуляція (контролер із пресетами), середній – міркування для контролю стану та аварій, верхній – когнітивний рівень з імітаційними моделями для самооптимізації. Подається лише концептуальна модель без деталей її реалізації. Давачі генерують великі потоки даних, що вимагає алгоритмів для виявлення нових дій та закономірностей через машинне навчання без учителя [25]. Це сприяє розпізнаванню та прогнозуванню активності, що дозволяє системі реагувати на дії користувача.

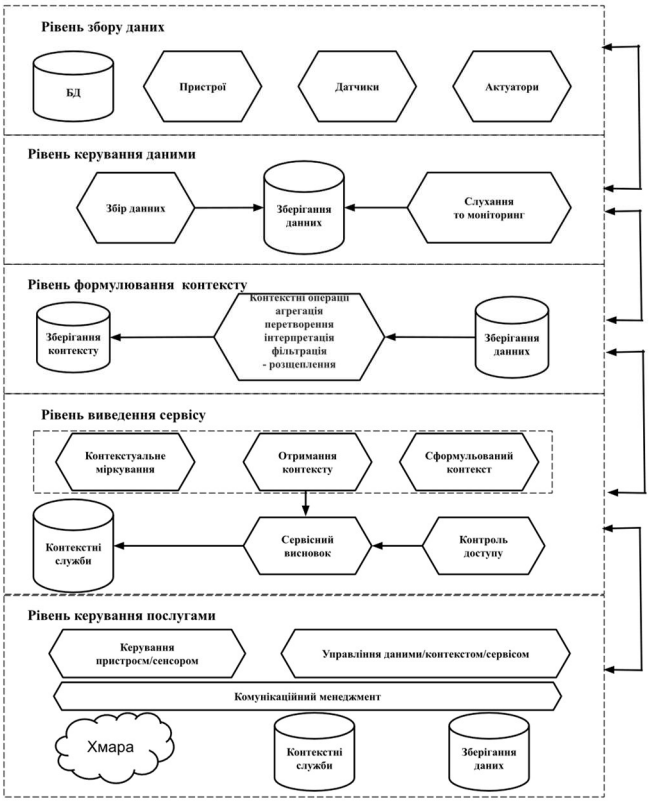


Рис. 1.4 Архітектура інформаційної системи «розумний будинок», що враховує контекст

Опрацювання великих обсягів даних вимагає обчислювальної потужності, яка є недоступною в локальних налаштуваннях. Тому дослідники [26] пропонують архітектуру домашньої інтелектуальної інформаційної системи з хмарними, периферійними або туманними обчисленнями, які використовуються для розвантаження обчислень в локальній системі.

Доцільним вважається використання [27] безпечної та ефективної архітектури інформаційної системи розумного будинку, яка інтегрує технології блокчейн і хмарних обчислень. Вона ґрунтується на децентралізованих технологіях блокчейн для надання послуг з опрацювання даних та забезпечення безпеки зібраних даних користувачів у інформаційних системах розумних будинків. При цьому використовується метод багатовимірного кореляційного аналізу для моніторингу мережевого трафіку та підвищення безпеки мережі розумного будинку. На основі процедури оцінювання з використанням такого критерію, як пропускна здатність, у дослідженні продемонстровано, що блокчейн служить ефективним рішенням гарантування безпеки. Запропонована п’ятирівнева архітектура інформаційної системи розумного будинку реалізує ідею усвідомлення контексту (Рис.1.4) [28](Додаток Б).

1.2 Терміни та означення

Еволюція інтелектуального середовища проживання узгоджується із законом Галла, який стверджує, що складні системи розвиваються з простих [29]. Важливо починати з мінімально життєздатного продукту, поступово додаючи складні функції. Сучасні системи домашньої автоматизації використовують технології Інтернету речей та включають давачі, прилади й виконавчі механізми для моніторингу та керування середовищем без втручання людини [30]. Вони зазвичай складаються з центрального контролера, що керує сенсорами, освітленням, безпекою та іншими приладами, забезпечуючи дистанційне управління через мобільні пристрої [31].

Термін Domotics описує дослідження та розробки розумних будинків, які використовують сенсори, мережі та простий інтелект [32, 33]. На наступному етапі розвитку з’являється концепція інтелектуального будинку – середовища, що інтегрує IoT та хмарні сервіси для оптимізації енергоспоживання, безпеки й комфорту [34]. В системах інтелектуальних будинків застосовуються методи аналізу даних та машинного навчання для розпізнавання дій мешканців, що дозволяє автоматизувати рутинні процеси й підвищити ефективність управління будинком [35].

Інша точка зору стосується інтелектуальних будинків як середовищ, заснованих на знаннях, де дані з різних сенсорів та пристроїв використовуються для розуміння і прогнозування поведінки користувачів, тим самим сприяючи проактивній допомозі і поліпшенню умов проживання [36]. Роблячи акцент на дизайні, орієнтованому на користувача, в інтелектуальних будинках необхідно забезпечити безперебійну та інтуїтивно зрозумілу взаємодію користувача з інформаційною системою, підвищуючи якість життя, роблячи повсякденну діяльність мешканців зручнішою та ефективнішою. Ці системи розробляються таким чином, щоб мати можливість адаптуватися до конкретних потреб і пріоритетів користувачів [37]. Концепція pervasive та повсюдного (ubiquitous) комп'ютингу описує інтеграцію великої кількості взаємопов'язаних пристроїв у побутові системи. Pervasive обчислення забезпечують розподілені, контекстно-залежні послуги, доступні користувачеві незалежно від локації [11]. Вони акцентують увагу на пристроях, мережах та опрацюванні даних, тоді як повсюдні обчислення пов'язані з взаємодією людини з комп'ютером. Ідеальною метою концепту «штучний інтелект» є мислити й діяти раціонально, подібно до міркувань і дій людини [38].

Перехід від концепції розумного дому до інтелектуального середовища відображається в термінологічному ряду розумного середовища, інтелектуального середовища та інтелекту навколишнього середовища. Розумне середовище містить сенсорні пристрої, що можуть локально зберігати й опрацьовувати дані [11]. Інтелектуальні середовища об'єднують розумне середовище з обчислювальним інтелектом та повсюдною доступністю послуг [11].

Існує кілька визначень інтелектуального середовища. Одні дослідники описують його як технологію, що адаптивно реагує на присутність людей і ненав'язливо підтримує їхню діяльність [1], інші – як програмне забезпечення, що інтелектуально допомагає в реалізації повсякденних завдань [11]. Загалом, інтелектуальне середовище інтегрує численні сенсори та обчислювальні пристрої, використовуючи алгоритми ШІ для надання послуг у зрозумілий, адаптивний і непомітний спосіб.

1.3 Керування безпекою та правами доступу в інтелектуальних будинках

Сучасні системи контролю доступу в інтелектуальних будинках використовують методи та засоби блокчейн, Інтернету речей і штучного інтелекту. Вони включають центр управління довірою та смарт-контракти для ідентифікації, оцінки надійності пристроїв і контролю доступу [39]. Наприклад, система розпізнавання обличчя базується на OpenMV, Arduino, RC522, Esp8266 і Ali Cloud IoT [40].

Розширені політики контролю доступу, такі як EGRBAC, забезпечують взаємодію користувачів і пристроїв із дотриманням вимог безпеки [41]. Дослідники аналізують адміністративні політики EGRBAC у контексті RBAC для запобігання несанкціонованому доступу та підвищення рівня прав доступу [42]. Інтеграція методів та засобів ШІ, зокрема глибинного навчання, покращує точність розпізнавання обличчя та об’єктів, що є ключовим для безпеки розумного будинку [43]. Також використовуються інтелектуальні системи з OpenMV, Arduino, Esp8266 WiFi, які підтримують NFC і керування через застосунки [44]. Блокчейн підвищує конфіденційність та безпеку, забезпечуючи реєстрацію пристроїв і управління довірою через смарт-контракти [45]. Використання методів та засобів ШІ у системах контролю доступу зумовлений необхідністю швидкого прийняття рішень у складних непередбачуваних сценаріях [46]. Інтелектуальна система безпеки повинна розуміти ситуацію, контролювати середовище, прогнозувати зміни та аналізувати їх. Реалізація такої системи передбачає використання емпіричних і контекстуальних знань, дій у реальному часі та оновлення бази знань, основою якої є онтологія. Вона забезпечує спільний словник для інтеграції методів ШІ, контролю доступу, спостереження та кібербезпеки, покращуючи сумісність, масштабованість і прийняття рішень. При цьому створення і підтримка онтології є одним з ключових завдань [47]. Онтологія визначається як формалізація спільної концептуалізації [47], проте це визначення критикується через нечіткість терміну «концептуалізація» [48]. Грубер [49] розглядає онтологію як інженерний

інструмент, визначають її як сукупність об’єктів, анотованих відповідно до логічної теорії. Формальний аналіз понять трактує «концепт» як вузол таксономії, що має єдине визначення, засноване на його атрибутах [50]. Інтелектуальні агенти самостійно генерують і змінюють концептуальні фреймворки на основі навчання. Їхні онтології є суб’єктивними та мають узгоджуватися із загальноприйнятими онтологіями для обміну знаннями. Розуміння концепту еволюціонує через спостереження та категоризацію об’єктів. Нові інтерпретації концепту можуть виникати через асоціації з іншими концептами [51]. Відстеження змін онтологій здійснюється через мережі онтологій із взаємозв’язками між компонентами [48]. Локальні онтології, сформовані агентами, враховують їх унікальні інтерпретації й називаються контекстними онтологіями [52]. Контекстні онтології [52] поділяються на локальні та спільні концептуалізації. Локальні концептуалізації зберігаються в пам’яті інтелектуального агента та узгоджуються зі спільними при необхідності взаємодії. Контексти визначаються як локальні концептуалізації [52], а багатоконтекстні онтології містять поняття з множинними інтерпретаціями [53]. Для подання контекстуальних відтінків значення понять використовується теорія прототипів і динамічний підбір значення залежно від ситуації [54]. Для автоматизованого опрацювання складних знань розроблена мова представлення онтологій OWL [55], що базується на дескриптивній логіці та полегшує встановлення зв’язків між поняттями, об’єктами й ролями. При значних розмірах онтологій [56,57], що можуть містити сотні тисяч понять (наприклад, Cyc [58] та WordNet [59]), кількість відношень зростає нелінійно, ускладнюючи обчислення. Оптимізація структурної складності онтологій передбачає їх поділ на загальні та доменні компоненти. Загальні онтології містять багаторазово використовувані елементи, тоді як доменні зосереджені на спеціалізованих концептах [60]. Проблемою залишається їх інтеграція, а також надмірна складність доменних онтологій у практичному застосуванні. Потенційне рішення – використання ретельно відібраних базових компонентів онтології. Процедура з’ясовування і формалізації найбільш фундаментальних понять і відношень, які можуть бути

послідовно використані в різних онтологіях, спричинила розвиток фундаментальних онтологій, таких як верхня об’єднана онтологія (Suggested Upper Merged Ontology, SUMO), Узагальнена фундаментальна онтологія (Unified Foundational Ontology, UFO) і Загальна формальна онтологія (General Formal Ontology, GFO) [61, 62], які функціонують разом з бібліотеками об’єктів і шаблонів [63]. Різноманітність і плинність онтологій предметної області вимагала дослідження закономірностей в побудові онтологій. Ця робота розпочалася з дослідження закономірностей концептуалізації при проектуванні інтелектуальних програмних систем [64].

Сучасний підхід ґрунтується на використанні патернів проектування онтологій (Ontology Design Patterns, ODPs) як будівельних блоків [65]. Патерн є формальною основою для спільного проектування онтологій. Усунення притаманної предметній області складності полягає в тому, щоб визначити невеликі, задачні та контекстно-орієнтовані онтології, які можуть бути використані як будівельні блоки формування цілісної онтологій. Дослідники зазвичай зосереджуються в основному на шаблонах контенту (CP). Ідея проектування онтологій патернів була розроблена після введення поняття «онтологія мови патернів» для організації пов’язаних онтологічних шаблонів в онтологічній інженерії [66]. Через схожість патернів проектування при розробленні програмного забезпечення та онтології запропоновано формулювати онтології предметної області у вигляді шаблонів проектування.

У роботі [67] наведено приклад побудови онтології тестування з використанням мови шаблонів онтології програмного процесу (Software Process Ontology Pattern Language, SP-OPL). Запропоновано формальне визначення онтологічно-орієнтованої уніфікованої мови моделювання (Ontology-driven Unified Modeling Language, OntoUML), концептуальної мови моделювання, з використанням граматики графа та онтологічних закономірностей [68] не залежить від мета-моделі UML і включає в себе мікротеорії з UFO. Розширення мови шаблонів онтологій (Ontology Pattern Language, OPLa) деталізовані в реорганізованому просторі онтологій: Мова шаблонів онтологій для базових

шаблонів (Ontology Pattern Language for core patterns, OPLo-core), що містить оригінальні анотації; Мова шаблонів онтологій для проектування програмного забезпечення (Ontology Pattern Language for software design, OPLe-SD) [69]. Останні тенденції в розробленні доменних онтологій як мов шаблонів полягають у адаптивності, модульності та практичному застосуванні, що зумовлено потребою в більш ефективному та безпомилковому розвитку онтологій, особливо в складних областях. Використання шаблонів проектування загальної онтології (Generic Ontology Design Patterns, GODPs) та розширень до існуючих мов є важливим для цих рішень. Використання GODPs як методології для подання та створення шаблонів проектування онтологій пов'язане із адаптивністю та зручністю використання експертами у певній галузі, не створюючи надлишковості у їхніх онтологіях [70]. Спостерігається [71] тенденція до використання мови шаблонів онтологій (ODP) для розроблення доменних онтологій. Цей підхід поєднує універсальні та спеціалізовані шаблони, забезпечуючи послідовне представлення сутностей, зменшення помилок і економію ресурсів. Використання ODP стало загальноприйнятим підходом для проектування модульних, контекстно-орієнтованих онтологій, що знижує витрати на їх обслуговування [72]. Онтології безпеки сприяють стандартизації знань, підтримці експертів і вдосконаленню кібербезпеки. Вони слугують основою для графа знань, що інтегрує методи штучного інтелекту для автоматизованого аналізу загроз [73]. Порівняльний аналіз онтологій кібербезпеки [74] показує важливість їх розширення для протидії загрозам, зокрема із застосуванням машинного навчання [75]. Окрему увагу приділено розробленню онтології кіберзахисних вправ (CDX) [76]. Вона використовує RDF та OWL для організації даних, отриманих під час CDX, сприяючи інтеграції знань з кібербезпеки та підтримці навчання. CDX-онтологія розглядається як цінний актив для подальших досліджень і розвитку галузі. Фундаментальні онтології надають базові набори елементів, словник і синтаксичні правила для побудови на їх основі систем міркувань та прийняття рішень. GFO як фундаментальна онтологія забезпечує системну основу для опису форм, способів

та поглядів на реальні об’єкти на різних рівнях абстракції та деталізації. Вона поєднує методи математичної логіки, філософії, штучного інтелекту та лінгвістики. GFO є компонентом інтегрованої системи фундаментальних онтологій (Integrated System of Foundational Ontologies, ISFO). ISFO є частиною інтегрованої структури для розроблення та застосування онтологій (Integrated Framework for Developing and Applying Ontologies, IFDAO). Крім ISFO, система IFDAO містить бібліотеку мов онтології та систему засобів розроблення. Ця система інструментів підтримує створення доменно-орієнтованих та загальних онтологій. І UFO, і GFO багаті деталями і добре опрацьовані. Однак важливою додатковою вимогою до інструментальної платформи, що працює з передбаченням ситуацій, є підтримка просторових і часових концептуальних конструкцій. Як основу моделювання бази знань інтелектуальної системи безпеки житлових комплексів обрано GFO, яка є 4d-онтологією [92], що підтримує просторові та часові концептуалізації.

1.4 Метрики для мережевих та ієрархічних структур

Наука починається там де є вимірювання, цей крилатий вислів дуже точно та об’ємно подає фундаментальне філософське підґрунтя науково-матеріалістичного сприйняття світу.

Метрика - це міра, яка дозволяє отримати числове значення певних вимірюваних властивостей та характеристик. Актуальною задачею у цьому контексті є формування метрик, тобто мір, для вимірювання параметрів, такого специфічного об’єкту дослідження, як онтологія. Однією з найпоширеніших форм подання онтологічних конструкцій є графи. У цьому контексті вимірювання характеристик онтологій зводиться до вимірювання певних параметрів різних видів графових структур.

В контексті побудови 4d онтологій таких як GFO для інтелектуальних програмних систем з ситуаційною обізнаністю аналіз повних конструктивних наборів метрик залишається актуальною науково-прикладною проблемою.

Метрикам для мережевих та ієрархічних структур, а також метрикам для оцінювання онтологій та концептів в онтологіях присвячено доволі багато

публікацій. Описані в джерелах метрики можна поділити на такі види: морфологічні метрики, ймовірнісні (Байесові), метрики для адаптивних онтологій, метрики для оцінювання онтологій, метрики для оцінювання зв'язків між концептами в онтологіях.

Морфологічні метрики для ієрархічних дерев. Фентон Н.Е. та Пфлеер С.Л. [77] описали набір простих морфологічних метрик для ієрархічних дерев, що грунтуються на характеристиках графів. Розмір *Size* = *n*, де *n* – кількість вершин. Густина взаємодії $R = e/n$ (відношення кількості ребер до кількості вершин). Для дерева $e = n–1$. Коефіцієнт розгалуження за виходом *Fan_out(i)* – це кількість дочірніх вершин *i*-тої вершини.

Первинними характеристиками графу є кількість вершин *n* та кількість ребер *e*. Для дерева до них додаються ще дві глобальні характеристики – висота і ширина. Висота – кількість рівнів (кількість вершин в найдовшому шляху від кореневої вершини до листової). Ширина – максимальна кількість вершин, розміщених на будь якому одному рівні дерева. Ширина рівня – це кількість вершин дерева на даному рівні, тоді ширина дерева – це максимальна ширина на всіх рівнях.

q-метрика для зваженого графа та природня метрика для звичайного графа. В роботі [78] наведено метрику для звичайного графа. Нехай $L[q] = (X,U;q)$ – звичайний граф з ваговою функцією *q*, яка ставить у відповідність до кожного ребра $u \in U$ дійсне число $q(u) > 0$ як довжини. Якщо Q – маршрут, то сума $q(Q) \equiv \sum_{u \in Q} q(u)$ по всім його ребрам називається його *q*-довжиною, а просто довжина – це кількість ребер маршруту (в обох випадках кожне ребро слід рахувати стільки разів, скільки воно зустрічається в маршруті).

Число $$d(x, y) \equiv d_L^q(x, y) \equiv \min\{q(Q) \mid Q \in \mathrm{Q}(x, y)\} \qquad (1.1)$$

де $\mathrm{Q}(x,y)$ – множина всіх простих ланцюгів із *x* до *y*, називається q-відстанню між вершинами $x, y \in X$ зваженого графа $L[q]$: якщо $x = y$, то Q – ланцюг нульової довжини і його *q*-довжина $q(Q) \equiv 0$, а якщо вершини *x* та *y* – відокремлені, то $\rho(x, y) \equiv +\infty$. Легко побачити, що q-відстань задовольняє трьом аксіомам метрики Фреше:

$$\forall\, x, y \in X\,[d(x, y) = 0 \Leftrightarrow x = y], \tag{1.2}$$

$$\forall\, x, y \in X\,[d(x, y) = d(y, x)], \tag{1.3}$$

$$\forall\, x, y \in X\,[d(x, y) + d(y, z) \geq d(x, z)], \tag{1.4}$$

тобто, є метрикою на множині X. В частковому випадку, коли всі $q(u) = 1$, тобто, коли q-відстань кожного ланцюга збігається з його звичайною довжиною, метрика $d(x, y) \equiv d_L^q(x, y)$ графа $L[q]$ називається природньою метрикою звичайного графа $L = (X, U)$.

Ймовірнісні (Байесові) метрики. В роботі К. Нейлора [79] запропоновано ймовірнісний підхід, що ґрунтується на теоремі Байеса, для оцінювання ієрархічних структур при побудові експертних систем. Метрика для таких систем ґрунтується на теоремі Байеса: ймовірність здійснення деякої гіпотези H за наявності певних свідчень E, які підтверджують цю гіпотезу (тобто, при настанні подій E), обчислюється на основі апріорної ймовірності даної гіпотези без свідчень-підтверджень E та ймовірності здійснення свідчень за умов, що гіпотеза вірна або хибна:

$$P(H \mid E) = \frac{P(HE)}{P(E)} \quad \Rightarrow \quad P(HE) = P(H \mid E) \cdot P(E) = P(E \mid H) \cdot P(H) \tag{1.5}$$

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}, \tag{1.6}$$

де $P(H)$ – апріорна ймовірність гіпотези H; $P(H \mid E)$ – ймовірність гіпотези H при настанні подій E (апостеріорна ймовірність); $P(E \mid H)$ – ймовірність настання подій E при істинності гіпотези H; $P(E)$ – ймовірність настання подій E.

Г. С. Теслер в роботі [80] розвиває запропонований К. Нейлором підхід. Нехай $G = (V, E)$ – зв'язний граф, u та v – дві його різні вершини. Тоді відстанню між вершинами u та v буде довжина найкоротшого маршруту, яка позначається $d(u, v)$. При цьому виконуються всі аксіоми метрики. Відомо, що всякий граф взаємно однозначно подається бінарним відношенням, яке можна задати матрицею суміжності. Елементи матриці суміжності $A(G)$ мають вигляд

$$a_{ij} = \begin{cases} 1, & \text{якщо вершини з номерами } i \text{ та } j \text{ - сумісні} \\ 0, & \text{в іншому випадку} \end{cases} \tag{1.7}$$

Рангом графа G називається ранг його матриці суміжності, позначається $rank(G)$. Якщо u – деяка вершина графу $G = (V, E)$, то величина $e(u) = \max d(u, v)$, $v \in V$ називається ексцентриситетом вершини u. Діаметром графа називають максимальний ексцентриситет серед його вершин і позначають $d(G) = \max e(u)$, $u \in V$. У цьому випадку як міру можна було б використовувати діаметр графа і побудувати метрику графів на основі їх діаметрів – що еквівалентно одній з морфологічних метрик.

Як приклад ієрархії Г. С. Теслер наводить дерево хвороб людини та їх зв'язки в залежності від причин їх виникнення та механізмів їх розвитку. Як метрику для подібних систем використовують теорему Байеса. Використання такого підходу при побудові експертної системи для медичної бази знань MYCIN наведено в роботі [79]; подібний підхід для аналізу кристалічних структур хімічних сполук описано в роботі [81].

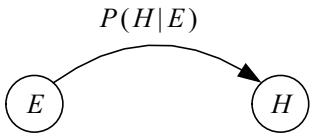


Рис.1.5 Ілюстрація до ймовірнісної метрики

Слід зауважити, що система MYCIN оперує поняттям «степінь певності»: процедурні правила в ній формулюються у вигляді «ЯКЩО … ТО … З ПЕВНІСТЮ Р», де степінь певності – «приблизно те саме, що ми називаємо умовною ймовірністю $P(H \mid E)$ – ймовірність гіпотези H за умови, що подія E відбулася» [79]. При побудові системи MYCIN експерти-медики пропонували правила і вказували степінь довіри до кожного правила в діапазоні від 1 до 10 – такі експертні оцінки і стали степеню певності для відповідних процедурних правил. Отже, набір процедурних правил такої системи можна описати за допомогою орієнтованого графу, кожне ребро якого має вагу – умовну ймовірність переходу $E \rightarrow H$, тобто ймовірність гіпотези H за умови, що відбулася подія E (Рис.1.5). Такий підхід дозволяє оцінити лише окремі частини

даного орієнтованого графу і не придатний для порівняння різних графів між собою, бо загальна ймовірність повної системи повинна дорівнювати одиниці $P(G=(V,E))=1$.

В роботах [82-84] запропоновано метрики на основі адаптивних онтологій для семантичних (наприклад, задач класифікації) та ознакових (наприклад, пошук релевантних прецедентів) задач.

Для семантичних задач відстань між прецедентом і ситуацією визначається як сума відстаней між „найважливішими" поняттями прецедента та поточного випадку. Найважливіше поняття відповідає центру ваг концептуального графа, за допомогою якого подається адаптивна онтологія. Розглядається максимум три «найважливіших» поняття концептуального графа. В цьому випадку отримуємо три центри ваг *i*-го прецедента та три центри ваг поточної ситуації s^1, s^2, s^3. Відстань між прецедентом та поточною ситуацією визначається як

$$d(pr,s)=\arg\min\sum_{n=1}^{3}d_n,\ d_n=d(pr_i^j,s^k),\ j=1,2,3,\ k=1,2,3 \qquad (1.8)$$

– з дев'яти різних відстаней $d\left(pr_i^j,s^k\right)$, j=1,2,3; k=1,2,3 вибираються такі три, щоб їх сума була мінімальною. Отримана сума і буде відстанню між прецедентом та поточною ситуацією.

1.5 Постановка проблеми в загальному вигляді та її зв'язок з важливими науковими або практичними завданнями

Основним трендом розвитку інформаційних систем сьогодні є впровадження методів та засобів штучного інтелекту у всі сфери життя. Успіх генеративного штучного інтелекту з використанням великих мовних моделей забезпечує значне підвищення продуктивності людини, забезпечуючи спрощений доступ до інформації та знань, створюючи контент, навчаючи та міркуючи.

При розробленні сучасних систем безпеки важливим є застосування методів штучного інтелекту, що сприятиме швидкому оцінюванню ситуацій та ухваленню ефективного рішення в реальному масштабі часу, оскільки, складно заздалегідь описати всі можливі сценарії безпеки. Тому інтелектуальна система

безпеки розглядається як ситуаційна система, здатна стежити за середовищем, аналізувати та передбачати зміни в ньому. Впровадження інтелектуальної системи безпеки передбачає урахування широкого спектру можливих ситуацій.

Беручи до уваги складність створення систем з урахуванням ситуації та дотримуючись ідеї поступового підходу до розроблення продукту, створення інтелектуальної системи безпеки житлового комплексу відбувалося як еволюційний, поетапний процес, починаючи з мінімально життєздатного продукту, одержуючи зворотний зв'язок від його використання та додавання нових функцій на його основі впродовж усього життєвого циклу. Реалізовуючи ітерації при розробленні інтелектуальної програмної системи безпеки формувалося її глобальне стратегічне бачення, що сприяло скороченню термінів розробки і чіткішому розумінню перспективних напрямків розвитку продукту, що створюється.

1.6 Вимоги до інтелектуальної програмної системи безпеки

Впровадження інтелектуальної системи безпеки в житлових будинках потребує складної та стійкої інфраструктури, здатної керувати численними точками доступу, комунальними об'єктами та великими фізичними середовищами. Це включає в себе інтеграцію різноманітних компонентів у системі безпеки, включаючи камери спостереження, механізми контролю доступу та системи сигналізації, розподілених по різних будівлях і спільних просторах [85].

Аналіз літератури [86-88] дозволяє визначити декілька функціональних профілів системи безпеки, включаючи контроль доступу, спостереження та моніторинг, кібербезпеку, управління реагуванням на надзвичайні ситуації, моніторинг навколишнього середовища, технічне обслуговування та оновлення (Рис.1.6). Функціонал інтелектуальної програмної системи безпеки повинен включати необхідність збору даних з IoT-пристроїв, забезпечення інтеграції з камерами спостереження, давачами руху, диму, температури, вологості, відкривання дверей і вікон, а також забезпечення у реальному масштабі часу консолідації даних з усіх підключених пристроїв. Інтелектуальна програмна

система повинна аналізувати зібрані дані для виявлення аномалій, загроз і потенційно небезпечних ситуацій, використовуючи алгоритми машинного навчання для прогнозування інцидентів та опрацьовуючи великі обсяги даних

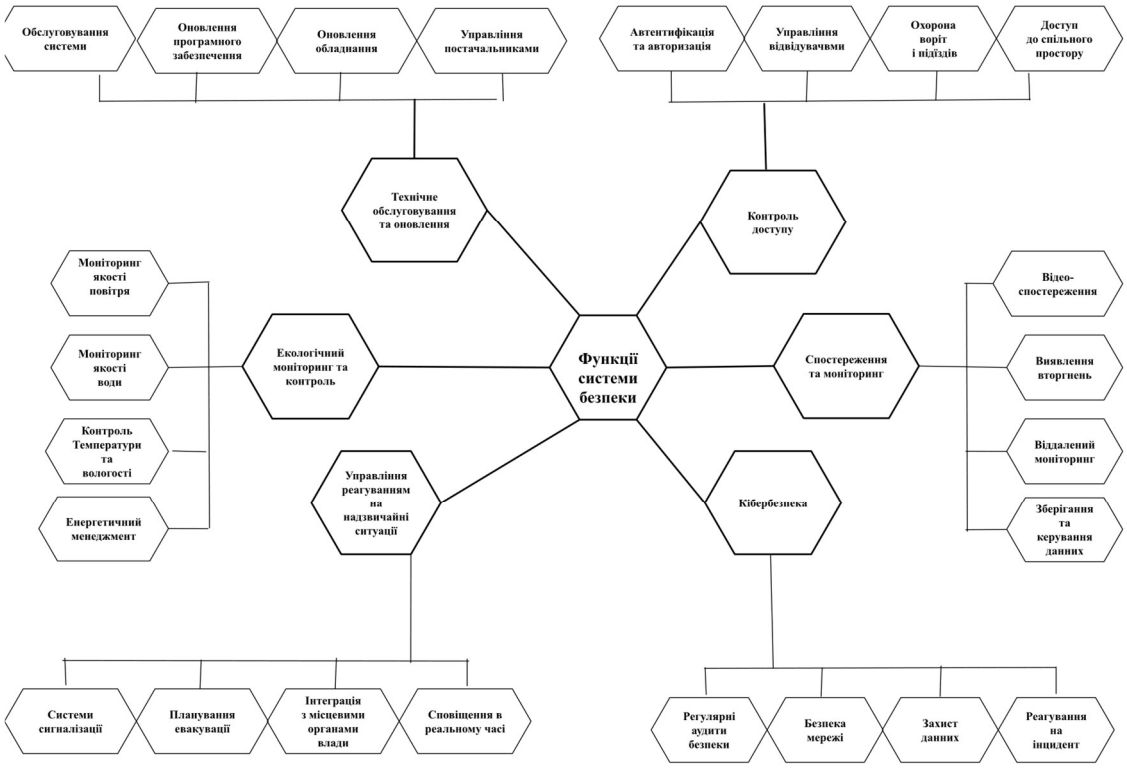


Рис. 1.6 Функції інтелектуальної програмної системи безпеки

для швидкого реагування на небезпечні ситуації. Для цього має відбуватися автоматичне формування моделей поточної ситуації на основі даних із сенсорів, генерація прогностичних моделей розвитку ситуації, можливість перегляду поточного стану системи і прогнозів у зручному інтерфейсі, які є важливими аспектами ситуаційної обізнаності.

Система повинна автоматично сповіщати мешканців, служби безпеки або екстрені служби про виявлену загрозу, активувати заходи безпеки, такі як увімкнення сигналізації, блокування дверей або відключення електропостачання в зоні ризику, і логувати всі події та реакції. Управління доступом передбачає ідентифікацію користувачів через картки доступу, мобільні застосунки чи біометричні дані, контроль доступу до певних зон кварталу і інтеграцію з системами розпізнавання обличчя та номерних знаків. Система повинна

забезпечувати мешканцям і адміністраторам зручний веб- або мобільний інтерфейс для моніторингу стану житлового комплексу, відображення в реальному часі відео-потоків із камер, графіків із даними сенсорів та інтерактивних карт, а також підтримувати функцію історичних записів для перегляду подій.

Функціональні профілі в свою чергу поділяються на специфічні функції. Контроль доступу включає автентифікацію та авторизацію відвідувачів, безпеку воріт і входів, доступ до спільного простору. Моніторинг відеоспостереження включає виявлення вторгнень, віддалений моніторинг, зберігання та керування відеоданими. В дисертаційній роботі основна увага зосереджена на реалізації підсистем контролю доступу, спостереження та моніторингу.

Висновки до розділу 1

Аналіз еволюції архітектурних інформаційно-технологічних та програмно-алгоритмічних рішень для реалізації концепту «розумний будинок» відображає стрімкий розвиток технологій, починаючи з простих автоматизованих систем до комплексних інтегрованих платформ, які об’єднують різноманітні «розумні» пристрої та сервіси. Сучасний етап розвитку такого роду архітектур зосереджується на інтеграції систем у єдину IT екосистему, що забезпечує високу зручність і повнішу функціональність. Одним із ключових викликів залишається забезпечення кібербезпеки в умовах взаємодії великої кількості пристроїв. Інтеграція методів та засобів штучного інтелекту дозволяє будинковим системам швидко та ефективно адаптуватися до потреб та поведінкових характеристик користувачів, забезпечуючи при цьому високу енергоефективність, комфорт та автоматизацію повсякденних рутинних функцій. У центрі сучасних архітектурних рішень при побудові такого роду систем перебуває користувач, для якого зручність управління, наявність інтуїтивно зрозумілих інтерфейсів та високої адаптивності системи до індивідуальних потреб мешканців є пріоритетними. Керування безпекою та правами доступу в розумних будинках реалізується зазвичай багатокомпонентною системою, яка покликана гарантувати безпеку мешканців, зручність управління та мінімізувати ризики

несанкціонованого доступу. Водночас не існує єдиної уніфікованої архітектури інформаційно-технологічних та програмно-алгоритмічних комплексів для розумних будинків. В роботах різних дослідників пропонуються широкий спектр архітектур, заснованих на розумних сервісах, інтелектуальних агентах або таких, що ґрунтуються на засадах централізованого інтелектуального оркестрування. Генеративний штучний інтелект, заснований на використанні великих мовних моделей, демонструє значний вплив на підвищення продуктивності у всіх сферах діяльності людини. Ця технологія спрощує доступ до інформації та знань, сприяє створенню контенту й підтримує процеси навчання. При цьому системи безпеки є однією з ключових областей застосування методів та засобів штучного інтелекту, оскільки вони потребують оперативної оцінки ситуацій і прийняття швидких рішень у динамічних часто непередбачуваних умовах. Інтеграція генеративного штучного інтелекту у системи безпеки відкриває нові можливості для автоматизації процесів, підвищення точності аналізу та швидкості прийняття рішень у складних і динамічних середовищах. Аналіз джерел показує, що використання онтологічних шаблонних мов є загальноприйнятим підходом для проектування доменних онтологій. Це дозволяє будувати модульні, контекстно-орієнтовані онтології та зменшує витрати на їх обслуговування.

Сформовано базові вимоги до функціоналу інтелектуальної системи безпеки житлового комплексу, які покладені в основу при її розробленні.

Розділ 2. Інструменти побудови інтелектуальних систем безпеки

2.1. Порівняльний аналіз програмно-апаратних засобів для створення системи безпеки житлового комплексу

Створення інформаційної системи безпеки житлового комплексу є необхідним та обумовленим кроком на шляху гарантування безпеки мешканців, підвищення якості їх життя, зниження ризиків та покращення економічної привабливості об’єктів нерухомості. Вибір IT платформи для створення систем безпеки житлових комплексів залежить від потреб та бюджету проекту. Тому перш ніж ухвалити відповідне рішення необхідно ретельно вивчити та проаналізувати характеристики та можливості різних IT платформ. Є доволі багато методів, якими можна послуговуватись в таких ситуаціях. В дисертаційній роботі фокус зроблений на використанні методу аналізу ієрархій, який в повній мірі задовольнив вимоги, які були сформовані у завданнях дисертаційного дослідження. Метод аналізу ієрархій або метод Т.Сааті – це структурований метод прийняття рішень, який дозволяє оцінити різні варіанти рішень шляхом розбиття проблеми за ієрархією критеріїв. Було проведено порівняльний аналіз платформ для контролерів, які потенційно можуть використовуватись для створення систем безпеки [89].

Реалізація методу передбачає виконання таких кроків:

Крок 1. Визначення мети.

Крок 2. Визначення критеріїв.

Крок 3. Побудова ієрархії.

Крок. 4. Оцінка критеріїв.

Крок 5. Оцінка варіантів за критеріями.

Крок 6. Розрахунок загального балу.

Метою використання зазначеного методу є вибір найкращої платформи для створення системи безпеки житлового комплексу. На другому кроці визначаємо критерії оцінювання у контексті системи безпеки, які умовно сформульовані таким чином: «гнучкість», «вартість», «легкість використання», «підтримка

пристроїв», «необхідні технічні знання». Критерій «гнучкість» визначає здатність платформи адаптуватися до різних умов, сценаріїв і вимог користувача. Критерій «вартість» визначає загальні витрати, пов'язані з встановленням, налаштуванням та обслуговуванням платформи. Критерій «легкість використання» визначає, наскільки проста та інтуїтивно зрозуміла платформа для кінцевих користувачів. Критерій «підтримка пристроїв» визначає здатність платформи працювати з широким спектром смарт-пристроїв та технологій. Критерій «необхідні технічні знання» визначають рівень технічних компетентностей, який потрібен користувачеві для ефективного встановлення, налаштування, управління та обслуговування платформи.

На третьому кроці побудовано ієрархію (Рис.2.1). Вершина визначає мету, середній рівень – критерії оцінювання, нижній рівень – альтернативи IT платформ.

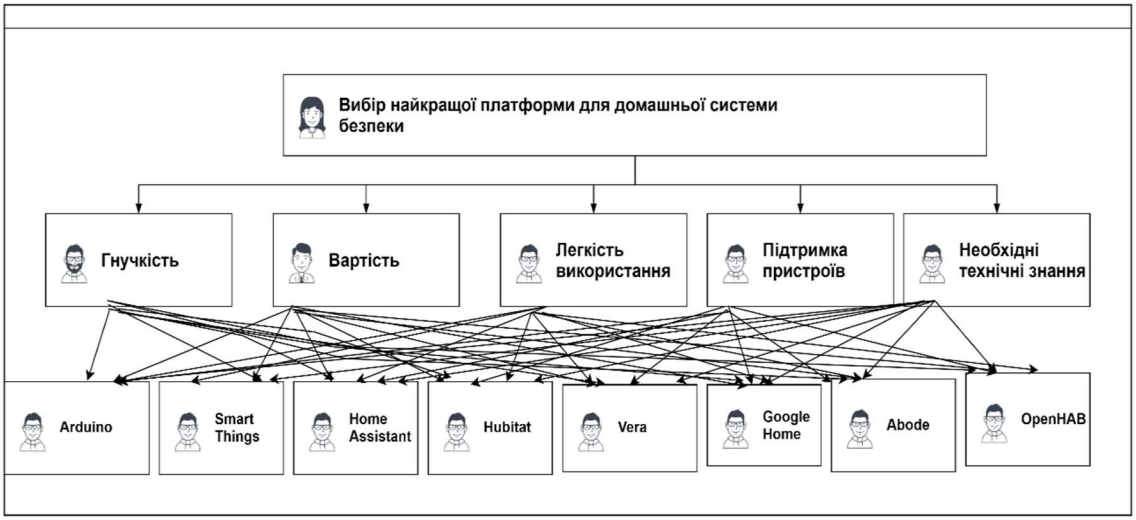


Рис. 2.1 Побудована ієрархія.

На наступному етапі були призначені ваги кожному критерію, використовуючи наступний алгоритм.

Крок 4.1. Вибір експертів.

На цьому кроці сформовано групу експертів, які володіють досвідом та знаннями у сфері систем безпеки житлових кварталів. Оскільки група високо фахових спеціалістів за даним профілем є доволі малочисельною, для проведення

числових розрахунків послуговувалися оцінками 5 експертів з числа профільних фахівців компанії «АСТРА».

Крок 4.2. Оцінка експертами важливості критеріїв.

Експерти провели оцінювання важливості кожного критерію за шкалою від 1 до 10, де 1 – мінімальна важливість, 10 – максимальна важливість. Для експертних оцінок використано традиційну шкалу: 1 – однакова значущість, 3 – слабка значущість, 5 – суттєва значущість, 7 – очевидна значущість, 9 – абсолютна значущість. При цьому 2, 4, 6, 8 – проміжні значення між сусідніми значеннями шкали. Якщо для порівняння першого елемента з другим задано одне з перелічених значень, то результат порівняння другого об'єкта з першим матиме обернене значення.

Крок 4.3. Агрегування оцінок**.**

На основі отриманих від експертів оцінок, було здійснено розрахунки середньої оцінки для кожного критерію.

Крок 4.4. Нормалізація оцінок**.**

Було проведено нормалізацію оцінок (1.1).

$$C_i = \frac{1}{m}\sum_{k=1}^{m} E_{jk} \qquad (2.1)$$

де C_i- оцінка важливості критерію, $\sum_{k=1}^{m} E_{jk}$ - середня оцінка важливості j-го критерію, m — кількість експертів, $\sum_{k=1}^{m} E -$ сума оцінок всіх експертів для j-го критерію, $E_{jk} -$ оцінка важливості j-го критерію, надана k-м експертом, це числове значення, яке показує, наскільки важливим вважає k-й експерт j-й критерій. За формулою (2.1) визначено середні оцінки важливості критеріїв (C_i):

Таблиця 2.1 Оцінки експертів

Критерій/ Експерти	1	2	3	4	5	Середня оцінка
C_1 (Гнучкість)	9	8	9	7	8	8.2
C_2 (Вартість)	6	7	6	5	6	6.0
C_3 (Легкість використання)	7	7	8	8	7	7.4
C_4 (Підтримка пристроїв)	8	7	8	7	7	7.4
C_5 (Необхідність технічних знань)	4	5	4	5	5	4.6

Крок 4.5. Розрахунок суми середніх оцінок.

На основі даних таблиці 2.1 обраховано суму середніх оцінок, яка склала 33,6.

Крок 4.6. Нормалізація оцінки кожного критерію.

Нормалізовані оцінки ваги кожного критерію визначено таким чином:

$$W_i = \frac{C_i}{\sum_{j=1}^{n} C_j}, \quad (2.2)$$

де W_i - вага *i* критерію, C_i - це середня оцінка важливості *i*-го критерію, $\sum_{j=1}^{n} C_j$ - сума середніх оцінок важливості всіх критеріїв, де *j*=1 до *j*=n, *n* – загальна кількість критеріїв.

Обчислені нормалізовані ваги критеріїв подані у таблиці 2.2.

Таблиця 2.2 Нормалізовані ваги

Критерій	Середня Оцінка	Нормалізована вага (частка)	Нормалізована вага (відсоток)
Гнучкість	8.2	$\frac{8.2}{33.6} = 0.244$	24.4%
Вартість	6.0	$\frac{6.0}{33.6} = 0.179$	17.9%
Легкість використання	7.4	$\frac{7.4}{33.6} = 0.220$	22.0%
Підтримка пристроїв	7.4	$\frac{7.4}{33.6} = 0.220$	22.0%
Необхідність технічних знань	4.6	$\frac{4.6}{33.6} = 0.137$	13.7%

Крок 4.7. Перерахунок оцінок експертів

Було проведено перерахунок оцінок експертів та отримані скориговані середні оцінки важливості критеріїв: гнучкість $C_1 = 0.3 \times 10 = 3.0$; вартість $C_2 = 0.2 \times 10 = 2.0$; легкість використання $C_3 = 0.2 \times 10 = 2.0$; підтримка пристроїв $C_4 = 0.2 \times 10 = 2.0$; необхідні технічні знання C_5=0.1×10=1.0.

Крок 4.8. Пропорційне масштабування**.**

Проведено пропорційне масштабування важливості кожного критерію і подано скориговані оцінки експертів: гнучкість $C_1^m = 10.0$; вартість $C_2^m = 6.7$; легкість використання $C_3^m = 6.7$; підтримка пристроїв $C_4^m = 6.7$; необхідні технічні знання $C_5^m = 3.3$.

Таблиця 2.3 Скориговані оцінки експертів.

Критерій	Експерт 1	Експерт 2	Експерт 3	Експерт 4	Експерт 5	Середня оцінка
Гнучкість	10	10	10	10	10	10.0
Вартість	6	6	6	6	6	6.0
Легкість використання	6	6	6	6	6	6.0
Підтримка пристроїв	6	6	6	6	6	6.0
Необхідні технічні знання	3	3	3	3	3	3.0

Крок 4.9. Перевірка нормалізації

Було виконано розрахунки суми скоригованих середніх оцінок, що склало 31.0 та проведено нормалізацію оцінок кожного критерію за формулою (2.2), результати якого подано у (Таблиця 2.4).

Таблиця 2.4 Нормалізовані ваги.

Критерій	Середня оцінка	Нормалізована вага (частка)	Нормалізована вага (відсоток)
Гнучкість	10.0	$\frac{10.0}{31.0}$ =0.323	32.3%
Вартість	6.0	$\frac{6.0}{31.0}$ =0.194	19.4%
Легкість використання	6.0	$\frac{6.0}{31.0}$ =0.194	19.4%
Підтримка пристроїв	6.0	$\frac{3.0}{31.0}$ =0.194	19.4%
Необхідні технічні знання	3.0	$\frac{6.0}{31.0}$ =0.097	9.7%

Крок 5. Оцінка альтернатив за критеріями

Експертами було проведене оцінювання IT платформ за кожним з критеріїв за шкалою від 1 до 5 (1 – найгірше, 5 – найкраще).

Таблиця 2.5 Оцінювання платформ за кожним критерієм

Платформа	Гнучкість	Вартість	Легкість використання	Підтримка пристроїв	Необхідні технічні знання
Home Assistant	5	4	2	3	2
SmartThings	4	3	5	4	4
Arduino	5	4	2	5	2
Hubitat	4	3	3	4	3
Vera	3	2	5	3	5
Google Home	3	3	5	4	4
Abode	3	3	5	4	4
OpenHAB	5	4	2	5	2

Крок 6. Побудова ієрархії

На основі сформованих оцінок експертів побудовано ієрархію IT платформ за кожним з критеріїв (див. Рис.2.2 та Додаток Г).

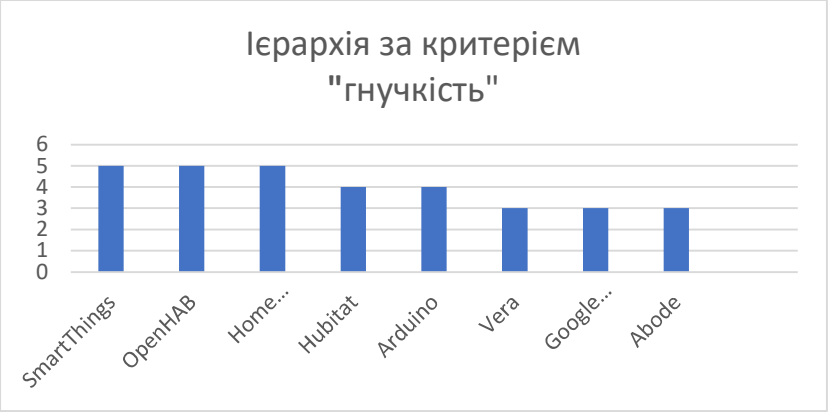


Рис. 2.2 Ієрархія за критерієм "гнучкість"

Крок 7. Розрахунок загального балу

Проведено розрахунки загального балу для кожної з пропонованих інструментальних платформ. Загальний бал для кожної платформи розраховано як сума добутків ваг критеріїв помножена на оцінки платформи за цими критеріями:

$$Z_j = \sum_{i=1}^{n}(P_i \times W_i) \quad (2.3)$$

де Z_j – загальний бал j-ї платформи, тобто загальна інтегральна оцінка платформи за всіма критеріями, n — кількість критеріїв, P_i— оцінка платформи за i-м критерієм, W_i — вага i-го критерію, яка відображає відносну важливість цього критерію. Для конкретної платформи формула виглядає наступним чином:

$$Z_j = (P_1 \times W_1) + (P_2 \times W_2) + (P_3 \times W_3) + (P_4 \times W_4) + (P_5 \times W_5), \quad (2.4)$$

де Z_1 загальний бал оцінки певної платформи, P_1 – оцінка за критерієм «гнучкість», W_1 - вага критерію «гнучкість», P_2 - оцінка за критерієм «вартість», W_2 – вага критерію «вартість», P_3 – оцінка за критерієм «використання», W_3 - вага критерію «легкість використання», P_4- оцінка за критерієм «підтримка пристроїв», W_4 – вага критерію «підтримка пристроїв», P_5 - оцінка за критерієм «необхідні технічні знання», W_5 –вага критерію «необхідні технічні знання».

Результати порівняльного аналізу інструментальних ІТ платформ для реалізації систем на основі інформаційних технологій IoT в системах безпеки житлових

Таблиця 2.6 Результати оцінювання платформи

Платформа	Загальний бал
Home Assistant	3.9
Arduino	4.0
SmartThings	3.5
Hubitat	3.5
Vera	3.4
Google Home	3.7
Abode	3.7
OpenHAB	3.9

комплексів, що виконані на базі проведеного експертного оцінювання фахівцями проектного відділу компанії «АСТРА», подані у Таблиця 2.6 та візуалізовано на рис.2.3.

Рис. 2.3 Сформована ієрархія за загальною оцінкою інструментальних ІТ платформ

На основі методу аналізу ієрархій (AHP) і експертного оцінювання вдалося встановити ряд важливих висновків:

1. Проаналізовані програмно-апаратні засоби мають різний ступінь гнучкості щодо налаштування відповідно до потреб користувачів та сумісності з іншими розумними пристроями. Це дозволяє користувачам обирати рішення, які найкраще відповідають їх вимогам та технічним умовам.

2. Вартість обладнання та його обслуговування є важливими факторами, що впливають на доступність і економічну доцільність використання конкретних рішень. Більш дорогі системи зазвичай пропонують вищу функціональність та надійність, проте потребують більших інвестицій.

3. Аналіз показав, що деякі системи є більш зручними для користувачів завдяки інтуїтивно зрозумілим інтерфейсам та мінімальним вимогам до технічних знань. Це спрощує процес налаштування та експлуатації системи для мешканців будинку.

4. Досліджувані платформи відрізняються за кількістю підтримуваних давачів, камер та інших пристроїв, а також за типами протоколів, що використовуються для зв'язку. При виборі платформи необхідно враховувати потреби у підтримці специфічних типів пристроїв та протоколів, які використовуються в конкретному будинку.

5. Більшість проаналізованих систем надають широкі можливості для автоматизації процесів, що дозволяє покращити ефективність управління безпекою, знижуючи втручання людини і мінімізуючи ризики людського фактору.

Кожна з розглянутих платформ має свої унікальні переваги та недоліки. Вибір конкретного рішення базується на ретельному аналізі потреб і пріоритетів користувачів, а також на основі наданих у статті порівняльних характеристик.

2.2 Методологія розроблення та супроводу інтелектуальних програмних систем, базованих на технології інтернету речей

У сучасних високотехнологічних проектних командах вибір методології розроблення та супроводу проектів відіграє ключову роль у досягненні успіху. Не виключенням у цьому плані є і проектні команди компанії «АСТРА», яка є львівським регіональним лідером надання інтернет послуг. Кожен підхід має свої специфічні характеристики, що впливають на процеси розроблення, тестування, розгортання і підтримки продукту, адаптуючи його до різних потреб та умов. Методології наділені певними перевагами та обмеженнями, що визначають доцільність їх застосування у певних типах проектів [90]. Тому для ухвалення обґрунтованого рішення, по визначенню методології, що найкраще відповідає цілям проекту зі створення інтелектуальних програмних систем безпеки, враховуючи поточні можливості команди, виклики та умови, було використано SWOT-аналіз (Таблиця 2.7).

Таблиця 2.7 Результати SWOT аналізу методологій управління проектами

Методологія	Сильні сторони	Слабкі сторони	Сфери застосування
Agile	Гнучкість у розробці, часті релізи, швидке адаптування до змін	Неврахування специфіки апаратної частини, можливі нестабільності	Підходить для динамічних проектів з часто змінюваними вимогами
DevOps	Автоматизація процесів, безперервна інтеграція та доставка, моніторинг у реальному часі	Висока вартість впровадження, потреба у високій автоматизації	Оптимально для великих, масштабованих IoT-систем у промислових умовах
Waterfall	Чітка структура, контроль кожного етапу, підходить для стабільних проектів	Низька гнучкість, тривалий процес розробки	Для проектів з фіксованими вимогами, стабільних умовах
Lean	Оптимізація ресурсів, швидке прототипування, мінімізація витрат	Обмежена масштабованість, можливі компроміси у якості	Для стартапів та малих проектів з обмеженими ресурсами
V-модель	Чітка структура з акцентом на тестуванні, висока якість продукту	Вимога чітко визначених вимог, низька гнучкість	Для критичних проектів з високими вимогами до надійності
Scrum	Часті релізи, короткі спринти, регулярна взаємодія з клієнтами	Вимога високого рівня комунікації, недостатня структура для великих проектів	Підходить для динамічних проектів з частими змінами, споживчих IoT-рішень

Проаналізовано особливості кожної з методологій управління проектами, їхні сильні та слабкі сторони, а також обставини, в яких їх найдоцільніше використовувати. На сьогодні популярними в професійних середовищах IT фахівців є методології Agile, DevOps, Waterfall, Lean, V-модель та Scrum, кожна з яких має своє призначення й особливості. Agile і Scrum орієнтовані на швидку адаптацію до змін, DevOps фокусується на автоматизації та безперервному розгортанні, Waterfall забезпечує чітку послідовність етапів, Lean оптимізує витрати, а V-модель акцентує увагу на тестуванні та якості. Аналіз існуючих методологій розроблення та супроводу IoT-систем, до яких належить проектована інформаційна система безпеки, включає розгляд основних підходів, якими є Agile, DevOps, Waterfall, Lean. Вибір конкретної методології в тому чи іншому випадку суттєво залежить від галузі застосування IoT-системи, вимог до її гнучкості, надійності, масштабованості та доступності ресурсів. Вибір на користь DevOps як методології для розроблення і супроводу IT-проектів із розроблення інтелектуальних програмних систем, базованих на технології IoT, обумовлений рядом вагомих переваг, які особливо актуальні у сучасних динамічних умовах розвитку IT-сфери та зростаючої складності інформаційних

систем. Методологія DevOps об’єднує розроблення та операційний супровід із фокусом на автоматизацію та безперервну інтеграцію, що значно підвищує ефективність управління IoT-системами, зокрема досліджуваною в дисертації. DevOps і DevSecOps інтегрують безпеку на всіх етапах життєвого циклу розроблення [91]. Методологія DevOps передбачає об’єднання команд розробників та операційних фахівців, що сприяє більш тісній їх взаємодії[92]. Це в свою чергу допомагає швидше вирішувати проблеми, обмінюватися інформацією та координувати дії, що особливо важливо для проектів із розподіленими командами [93]. Об’єднання команд розробників та операційних фахівців покращує комунікацію і пришвидшує вирішення проблем.

Автоматизація тестування, інтеграції, доставки та розгортання прискорює розробку та мінімізує кількість помилок. CI/CD забезпечує швидке впровадження змін без порушення стабільності системи, що критично для масштабних проектів. Автоматизований моніторинг і швидке розгортання оновлень мінімізують простої та ризики відмови, що особливо важливо для критичних і масштабованих систем. Вбудовані інструменти безпеки DevOps знижують ризик вразливостей, а безперервне тестування дозволяє виявляти і виправляти помилки на ранніх етапах. DevOps забезпечує стабільність, швидкість оновлень та високу якість продукту, роблячи її оптимальним вибором для складних та критичних IT-проектів.

IoT-системи зазвичай включають велику кількість пристроїв, підключених до мережі, що робить процес оновлення програмного забезпечення на кожному з них складним і трудомістким. Методологія DevOps дозволяє автоматизувати цей процес, забезпечуючи централізоване розгортання та оновлення програмного забезпечення на всіх пристроях одночасно. Для забезпечення стабільної роботи IoT-систем необхідний постійний моніторинг стану пристроїв та системи в цілому.

2.3 Інструменти методології DevOps в інтелектуальних програмних системах безпеки на основі технологій IoT

Розроблення інтелектуальних систем безпеки житлових комплексів вимагає застосування сучасних методологій, зокрема DevOps. Використання DevOps у системах на основі IoT супроводжується рядом викликів, таких як глибока інтеграція програмного та апаратного забезпечення, налагодження стабільності роботи при змінному навантаженні та підвищенні вимог до рівня кібербезпеки.

Аналіз DevOps у контексті систем на базі IoT зосереджено на взаємозв'язку методології з програмними комплексами безпеки, які використовують розгалужені мережі IoT-пристроїв. Проведене аналітичне дослідження дало підстави виокремити шість груп інструментів, що використовуються при цьому як компоненти методології DevOps. Проаналізовані та виокремлені інформаційно-технологічні інструменти як засоби побудови інтелектуальних програмних систем безпеки дозволяють реалізувати безперервну інтеграцію та доставку, автоматизацію процесів, моніторинг та управління конфігураціями, що є критично важливими для успішного розроблення IoT-застосунків [94-97].

Інтеграція факторів безпеки на кожному з етапів розроблення інтелектуальних програмних систем безпеки засобами методології DevSecOps (Розроблення. Безпека. Операційність) дозволили зреалізувати проактивний підхід в забезпеченні належного рівня безпеки. Це, зокрема, включає автоматизоване сканування вразливостей, тестування рівнів безпеки та впровадження кращих практик безпеки [98]. Досягнення належного рівня безпеки засобами методології DevOps в IoT-застосунках вимагає впровадження спеціальних методів та практик, оскільки IoT-застосунки часто містять унікальні безпекові ризики, пов'язані з фізичними пристроями, мережами і даними. На рис. 2.4 наведено декілька ключових методів убезпечення DevOps процесів, які використовуються в IoT-застосунках, і отримали назву DevSecOps. Безпечне програмування включає використання статичного та динамічного аналізу коду як інструментів для автоматизованого аналізу з метою виявлення вразливостей на

ранніх етапах розроблення. Використання безпечних бібліотек і фреймворків сприяло вибору перевірених бібліотек з активною підтримкою та регулярними оновленнями. Ідентифікація і управління доступом забезпечуються за рахунок аутентифікації та авторизації, зокрема двофакторної аутентифікації, що гарантує коректний контроль доступу до пристроїв і даних, а також ідентифікації кожного IoT-пристрою для контролю і обмеження доступу до мережі. Шифрування даних відбувається як у стані спокою, так і під час передачі між пристроями та серверами, із застосуванням VPN і протоколів TLS для захисту каналів зв'язку. Моніторинг та логування включає централізоване логування для збору і аналізу даних з усіх IoT-пристроїв і серверів, а також моніторинг у реальному часі для оперативного реагування на атаки.

Рис. 2.4 Ключові методи убезпечення DevOps процесів в IoT-застосунках.

Оновлення та патчинг забезпечують усунення вразливостей IoT-пристроїв, мінімізуючи час між виявленням і виправленням. Контейнеризація та мікросервісна архітектура ізолюють компоненти системи, покращуючи контроль і безпеку. CI/CD процеси включають автоматизоване тестування та використання інфраструктури як коду (IaC) для узгодженості налаштувань. Планування реагування на інциденти передбачає швидке усунення атак і постійне навчання команди.

Методологія DevOps сприяє масштабованості IoT-систем. Автоматизоване розгортання дозволяє швидко оновлювати програмне забезпечення на багатьох пристроях. IaC спрощує управління інфраструктурою, а CI/CD гарантує якість коду. Розподіл навантаження та мікросервіси забезпечують незалежне масштабування компонентів. Контейнери та оркестратори, такі як Kubernetes, автоматизують управління та забезпечують гнучкість. Централізований моніторинг і аналіз журналів спрощують виявлення проблем. Хмарні провайдери та serverless-обчислення забезпечують ефективне горизонтальне масштабування без значних витрат.

В результаті проведеного аналітичного дослідження груп інформаційно-технологічних інструментів зроблено висновок, що актуальною задачею є формування сукупностей сумісних DevOps інструментів, які б охоплювали весь життєвий цикл IoT застосунку, а саме розроблення та управління кодом, конфігурування та управління інфраструктурою, розгортання та управління контейнерами, тестування, моніторинг та логування. Вибір сукупності визначається типом проекту та особливостями вимог до інформаційних систем.

При обранні DevOps інструментів для побудови інформаційної системи безпеки житлового комплексу враховувалась її специфіка як IoT системи, зокрема її складність, технологічний стек, вимоги до безпеки тощо. Важливим етапом було залучення фахових експертів [99]. Під час оцінювання DevOps інструментів було залучено фахівців з різних областей, зокрема розробників IoT пристроїв, системних адміністраторів та архітекторів. При наявності значної кількості проаналізованих у кожній категорії DevOps інструментів (Рис. 2.4), множина можливих комбінацій є доволі великою. Експертами було запропоновано сформувати три комплекси інструментів із сукупності проаналізованих, з врахуванням їх сумісності для використання у різних типах проектів (таблиця 2.7):

- Tlt_1^{DevOps} – сукупність DevOps інструментів для проектів із створення інформаційних систем з використанням хмарної інфраструктури;

- Tlt_2^{DevOps} – сукупність DevOps інструментів для великих enterprise-проектів із створення інформаційних систем з підвищеними вимогами до безпеки та масштабованості;
- Tlt_3^{DevOps} – сукупність DevOps інструментів для невеликих проектів із створення інформаційних систем з можливостями оперативного розгортання.

Таблиця 2.8 Сукупності DevOps інструментів

	DevOps інструменти		
Сукупність інструментів	Tlt_1^{DevOps}	Tlt_2^{DevOps}	Tlt_3^{DevOps}
CI/CD	AWS CodePipeline	GitLab CI/CD	CircleCI
Управління конфігураціями	AWS IoT Device Management	Puppet	Ansible
Моніторинг	AWS CloudWatch	Datadog	Prometheus, Grafana
Контейнери	Amazon ECS або EKS	Kubernetes (OpenShift)	Docker
Управління версіями	Git (CodeCommit)	GitLab	Git (GitHub)
Тестування	Device Farm (AWS)	Selenium, TestComplete	JUnit, Mockito

Для вибору набору інструментів знову був використаний метод Т.Сааті. Для цього було сформовано множину критеріїв:

$$Cri_{tools}^{DevOps} = \{Tool_{SCLB}^{DevOps}, Tool_{INTGRN}^{DevOps}, Tool_{CSTCMM}^{DevOps}, Tool_{FNCT}^{DevOps}, Tool_{SCRT}^{DevOps}\} \quad (2.7)$$

де $Tool_{SCLB}^{DevOps}$ – *«масштабованість»* – здатність DevOps інструмента обслуговувати значну кількість IoT-пристроїв і передавати великі за обсягом потоки даних;

$Tool_{INTGRN}^{DevOps}$ – *«інтеграція»*, що характеризує наскільки легко DevOps інструмент інтегрується з іншими інструментами та системами, наприклад, хмарними платформами;

$Tool_{CSTCMM}^{DevOps}$ – *«вартість та спільнота»*, на цей критерій впливають особливості ліцензування, можливість оперативного використання хмарних ресурсів, супроводу, розлогість спільноти користувачів тощо;

$Tool_{FNCT}^{DevOps}$ – *«функціональність»*, що характеризує наскільки повно DevOps інструмент покриває необхідні функції, інтуїтивність інтерфейсу, наявність документації та навчальних матеріалів;

$Tool_{SCRT}^{DevOps}$ – *«безпека»*, визначає рівень захисту даних і відповідність стандартам безпеки.

Побудовано ієрархічну модель (Рис. 2.5), де на верхньому рівні подана мета, яка передбачала створення оптимальної сукупності, на наступному рівні – критерії Cri_{tools}^{DevOps}, а на нижньому – сукупності DevOps інструментів $Tlt_{i=1..3}^{DevOps}$.

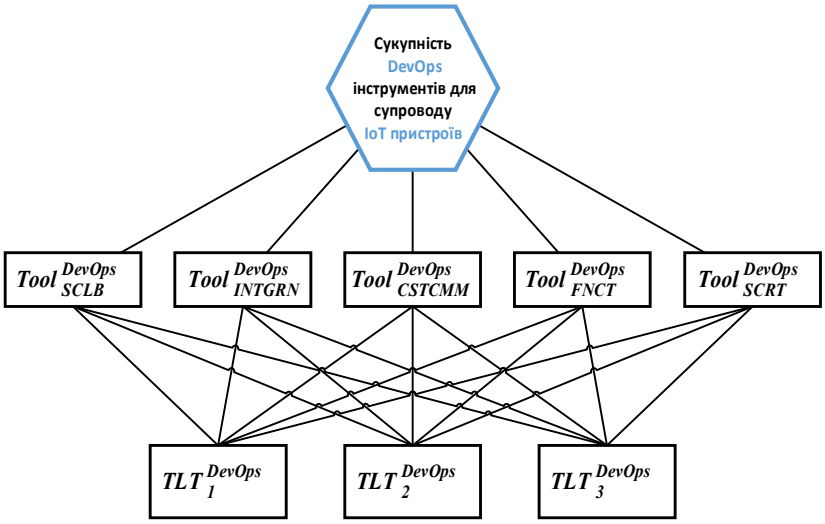


Рис. 2.5 Ієрархічна модель вибору DevOps інструментів для супроводу IoT систем

Для експертних оцінок використано традиційну шкалу. Проведено нормалізацію оцінок за формулою (2.5). В таблиці 2.8 подано матрицю попарних порівнянь для альтернатив Tlt_1^{DevOps}, Tlt_2^{DevOps} та Tlt_3^{DevOps} за критерієм «*масштабованість*».

$\lambda_{max} = 3{,}108$ та індекс узгодженості $IP = 0{,}054$. Для $n = 3$ випадковий індекс $VI = 0{,}58$. Відношення узгодженості для поданої в таблиці 2.8 матриці попарних порівнянь для альтернатив за критерієм $Tool_{SCLB}^{DevOps}$ обчислено $VP = 0{,}093$. Оскільки $VP < 0{,}1$, то матрицю попарних порівнянь за критерієм «масштабованість» вважаємо узгодженою. В Таблиці 2.9 подано матрицю

Таблиця 2.9 Матриця попарних порівнянь для альтернатив за критерієм $Tool_{SCLB}^{DevOps}$ –«масштабованість»

$Tool_{SCLB}^{DevOps}$ «масштабованість»	Tlt_1^{DevOps}	Tlt_2^{DevOps}	Tlt_3^{DevOps}	Вектор локальних пріоритетів	Нормований вектор
Tlt_1^{DevOps}	1	2	3	1,817	0,517
Tlt_2^{DevOps}	½	1	4	1,260	0,359
Tlt_3^{DevOps}	1/3	1/4	1	0,437	0,124
Сума	1,83	3,25	8,00	3,514	1

Для цієї матриці обчислено оцінку максимального власного значення попарних порівнянь для альтернатив Tlt_1^{DevOps}, Tlt_2^{DevOps} та Tlt_3^{DevOps} за критерієм «інтеграція».

Таблиця 2.10 Матриця попарних порівнянь для альтернатив за критерієм $Tool_{INTGRN}^{DevOps}$ – «інтеграція»

$Tool_{INTGRN}^{DevOps}$ «інтеграція»	Tlt_1^{DevOps}	Tlt_2^{DevOps}	Tlt_3^{DevOps}	Вектор локальних пріоритетів	Нормований вектор
Tlt_1^{DevOps}	1	4	5	2,714	0,674
Tlt_2^{DevOps}	¼	1	3	0,909	0,226
Tlt_3^{DevOps}	1/5	1/3	1	0,405	0,101
Сума	1,45	5,33	9,00	4,028	1

Для цієї матриці обчислено оцінку максимального власного значення $\lambda_{max} = 3{,}086$ та індекс узгодженості $IP = 0{,}043$. Для $n = 3$ випадковий індекс $VI = 0{,}58$. Відношення узгодженості для поданої в таблиці 2.9 матриці попарних порівнянь для альтернатив за критерієм $Tool_{INTGRN}^{DevOps}$ обчислено $VP = 0{,}074$. Оскільки $VP < 0{,}1$, то матрицю попарних порівнянь за критерієм «інтеграція» вважаємо узгодженою. В таблиці 2.10 подано матрицю попарних порівнянь для альтернатив Tlt_1^{DevOps}, Tlt_2^{DevOps} та Tlt_3^{DevOps} за критерієм «вартість та спільнота».

Для цієї матриці обчислено оцінку максимального власного значення $\lambda_{max} = 3{,}025$ та індекс узгодженості $IP = 0{,}01$. Для $n = 3$ випадковий індекс

Таблиця 2.11 Матриця попарних порівнянь для альтернатив за критерієм $Tool_{CSTCMM}^{DevOps}$ – «вартість та спільнота»

$Tool_{CSTCMM}^{DevOps}$ «вартість та спільнота»	Tlt_1^{DevOps}	Tlt_2^{DevOps}	Tlt_3^{DevOps}	Вектор локальних пріоритетів	Нормований вектор
Tlt_1^{DevOps}	1	1/5	1/4	0,368	0,097
Tlt_2^{DevOps}	5	1	2	2,154	0,570
Tlt_3^{DevOps}	4	1/2	1	1,260	0,333
Сума	10,00	1,70	3,25	3,783	1

$VI = 0{,}58$. Відношення узгодженості для поданої в таблиці 2.10 матриці попарних порівнянь для альтернатив за критерієм $Tool_{CSTCMM}^{DevOps}$ обчислено $VP = 0{,}02$. Оскільки $VP < 0{,}1$, то матрицю попарних порівнянь за критерієм «вартість та спільнота» вважаємо узгодженою. В таблиці 2.11 подано матрицю попарних порівнянь для альтернатив Tlt_1^{DevOps}, Tlt_2^{DevOps} та Tlt_3^{DevOps} за критерієм «функціональність».

Таблиця 2.12 Матриця попарних порівнянь для альтернатив за критерієм $Tool_{FNCT}^{DevOps}$ – «функціональність»

$Tool_{FNCT}^{DevOps}$ функціональність	Tlt_1^{DevOps}	Tlt_2^{DevOps}	Tlt_3^{DevOps}	Вектор локальних пріоритетів	Нормований вектор
Tlt_1^{DevOps}	1	1/4	1/3	0,437	0,117
Tlt_2^{DevOps}	4	1	3	2,289	0,614
Tlt_3^{DevOps}	3	1/3	1	1,000	0,268
Сума	8,00	1,58	4,33	3,726	1

Для цієї матриці обчислено оцінку максимального власного значення $\lambda_{max} = 3{,}074$ та індекс узгодженості $IP = 0{,}04$. Для $n = 3$ випадковий індекс $VI = 0{,}58$. Відношення узгодженості для поданої в таблиці 2.11 матриці попарних порівнянь для альтернатив за критерієм $Tool_{FNCT}^{DevOps}$ обчислено $VP = 0{,}06$. Оскільки $VP < 0{,}1$, то матрицю попарних порівнянь за критерієм «функціональність» вважаємо узгодженою. В таблиці 2.12 подано матрицю попарних порівнянь для альтернатив Tlt_1^{DevOps}, Tlt_2^{DevOps} та Tlt_3^{DevOps} за критерієм «безпека».

Таблиця 2.13 Матриця попарних порівнянь для альтернатив за критерієм $Tool_{SCRT}^{DevOps}$ – «безпека»

$Tool_{SCRT}^{DevOps}$ «безпека»	Tlt_1^{DevOps}	Tlt_2^{DevOps}	Tlt_3^{DevOps}	Вектор локальних пріоритетів	Нормований вектор
Tlt_1^{DevOps}	1	3	5	2,466	0,662
Tlt_2^{DevOps}	1/3	1	2	0,874	0,234
Tlt_3^{DevOps}	1/5	1/2	1	0,464	0,125
Сума	1,53	4,50	8,00	3,804	1

Для цієї матриці обчислено оцінку максимального власного значення $\lambda_{max} = 3{,}066$ та індекс узгодженості $IP = 0{,}03$. Для $n = 3$ випадковий індекс $VI = 0{,}58$. Відношення узгодженості для поданої в таблиці 2.12 матриці попарних порівнянь для альтернатив за критерієм $Tool_{SCRT}^{DevOps}$ обчислено $VP = 0{,}06$. Оскільки $VP < 0{,}1$, то матрицю попарних порівнянь за критерієм «функціональність» вважаємо узгодженою. В таблиці 2.13 подано матрицю попарних порівнянь критеріїв Cri_{tools}^{DevOps} щодо мети створення оптимальної сукупності DevOps інструментів для супроводу IoT пристроїв.

Таблиця 2.14 Матриця попарних порівнянь критеріїв щодо мети

	$Tool_{SCLB}^{DevOps}$	$Tool_{INTGRN}^{DevOps}$	$Tool_{CSTCMM}^{DevOps}$	$Tool_{FNCT}^{DevOps}$	$Tool_{SCRT}^{DevOps}$	Вектор локальних пріори-тетів	Нормо-ваний вектор
$Tool_{SCLB}^{DevOps}$	1	2	2	3	1	1,644	0,286
$Tool_{INTGRN}^{DevOps}$	1/2	1	3	1	1/7	0,735	0,128
$Tool_{CSTCMM}^{DevOps}$	1/2	1/3	1	1	1/3	0,561	0,098
$Tool_{FNCT}^{DevOps}$	1/3	1	1	1	½	0,699	0,122
$Tool_{SCRT}^{DevOps}$	1	7	3	2	1	2,112	0,367
Сума	3,33	8,00	7,00	8,00	3,33	5,750	1

Для цієї матриці обчислено оцінку максимального власного значення $\lambda_{max} = 5{,}442$ та індекс погодженості $IP = 0{,}11$. Для $n = 5$ випадковий індекс $VI = 1{,}12$. Відношення погодженості для поданої в таблиці 2.13 матриці попарних порівнянь критеріїв Cri_{tools}^{DevOps} обчислено $VP = 0{,}1$. Оскільки значення

$VP = 0{,}1$, рівне граничному значенню узгодженості, то матрицю попарних порівнянь вважаємо узгодженою. Результати обчислення оцінок ваг альтернатив критеріїв подано у таблиці 2.14, а вектор глобальних пріоритетів подано в таблиці 2.15.

Таблиця 2.15 Результати обчислення оцінок ваг альтернатив критеріїв

	$Tool_{SCLB}^{DevOps}$	$Tool_{INTGRN}^{DevOps}$	$Tool_{CSTCMM}^{DevOps}$	$Tool_{FNCT}^{DevOps}$	$Tool_{SCRT}^{DevOps}$
Tlt_1^{DevOps}	0,517	0,674	0,097	0,117	0,662
Tlt_2^{DevOps}	0,359	0,226	0,570	0,614	0,234
Tlt_3^{DevOps}	0,124	0,101	0,333	0,268	0,125

Таблиця 2.16 Вектор глобальних пріоритетів

$Tool_{SCLB}^{DevOps}$	$Tool_{INTGRN}^{DevOps}$	$Tool_{CSTCMM}^{DevOps}$	$Tool_{FNCT}^{DevOps}$	$Tool_{SCRT}^{DevOps}$
0,286	0,128	0,098	0,122	0,367

Зважені результати вибору подано в таблиці 2.16 та на рис.2.6.

Таблиця 2.17 Зважені результати вибору

Альтернатива	Пріоритет
Tlt_1^{DevOps} – сукупність DevOps інструментів для проектів з акцентом на IoT-платформах AWS	0,50075
Tlt_2^{DevOps} – сукупність DevOps інструментів для великих enterprise-проектів IoT систем з підвищеними вимогами до безпеки та масштабованості	0,34765
Tlt_3^{DevOps} – сукупність DevOps інструментів для невеликих IoT проектів з можливостями оперативного розгортання	0,15925

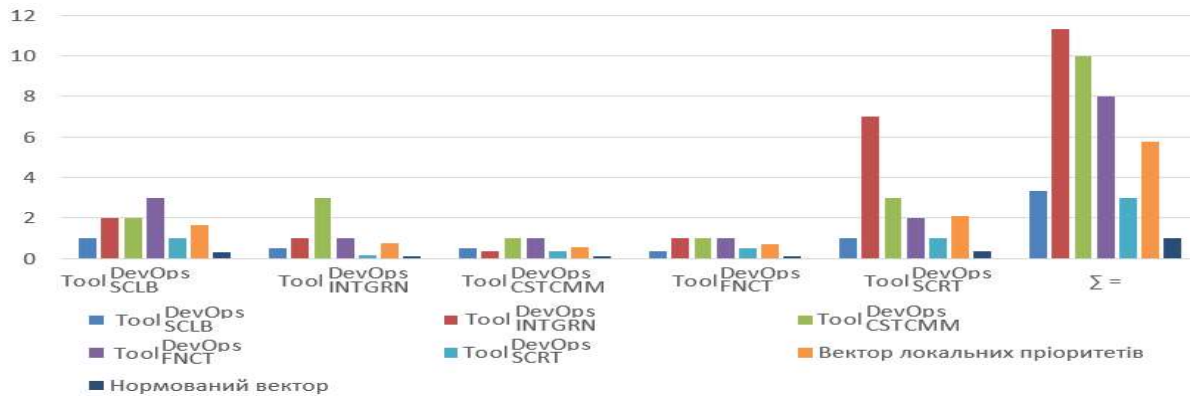


Рис. 2.6 Результати обчислень ієрархічної моделі вибору DevOps інструментів для супроводу IoT систем

З використанням методу аналізу ієрархій, експертного оцінювання на основі поданих обчислень обрано сукупності DevOps інструментів Tlt_1^{DevOps} з акцентом на IoT-платформах AWS, оскільки вона має найбільше розрахункове значення ваги та комплексно підходить для формування IoT систем (Рис.2.6). А саме CI/CD інструмент AWS CodePipeline, для управління конфігураціями AWS IoT Device Management, моніторинг – AWS CloudWatch, для управління контейнеризацією Amazon ECS або EKS, для управління версіями – Git (CodeCommit) та для тестування – Device Farm (AWS). Використання цих методів і практик DevOps в процесі побудови та супроводу інтелектуальних програмних систем безпеки дозволяє забезпечити для реальної IoT-системи ефективну адаптацію до суттєвого зростання в перспективі кількості підключених пристроїв, збільшення обсягу даних і змін вимог з боку замовників.

Висновки до розділу 2

У результаті проведеного дослідження було здійснено порівняльний аналіз ряду програмно-апаратних інструментальних засобів, що доцільно було б використовувати при створенні інформаційно-технологічної платформи для інтелектуальної системи безпеки житлового комплексу. Застосування методу аналізу ієрархій дозволило систематизувати та структуровано порівняти ряд інструментальних засобів, з метою ухвалення обґрунтованих рішень щодо вибору оптимальних програмно-апаратних рішень при створенні інтелектуальної системи безпеки багатоквартирних будинків. Проведений SWOT аналіз методологій управління IT проектами сприяв обранню методології DevOps, як базової, що забезпечує ефективну реалізацію функцій безперервної інтеграції, тестування та доставки, які є критично важливими для ефективного розгортання та експлуатації динамічних IoT-систем із великою кількістю компонентів і пристроїв. Ефективне використання методології DevOps та DevSecOps дозволило максимально адаптувати проектовану IoT-систему до функціональних вимог та забезпечити безперервність процесів розвитку і системну підтримку розширення функціональності.

Дослідження засвідчили, що використання методології DevOps при розробленні інтелектуальної системи безпеки житлових комплексів на базі технологій IoT забезпечує ефективність, безпеку та надійність проектних рішень, дозволяючи системі швидко адаптуватися до змін і забезпечувати стабільну роботу розлогих мереж з використанням великої кількості підключених різнотипових пристроїв. На основі проведених досліджень для розроблення інтелектуальної системи було обрано сукупність DevOps-інструментів, яка містить такі інструменти як: AWS CodePipeline для реалізації функцій CI/CD, AWS IoT Device Management для управління конфігураціями, AWS CloudWatch для моніторингу, Amazon ECS або EKS для ефективної контейнеризації, Git для управління версіями, а AWS Device Farm для забезпечення процесів тестування.

Розділ 3. Побудова проблемно-орієнтованої онтології інтелектуальної програмної системи безпеки з ситуаційною обізнаністю

3.1 Онтологічні засади системи міркувань ситуаційної обізнаності

3.1.1 Міркування про передбачувані ситуації

Для цілей даного дослідження була обрана загальна формальна онтологія (GFO) [101], яка забезпечує структуру для концептуалізації форм, модальностей та сутностей на різних рівнях абстракції та деталізації. Розглянемо складові GFO, пов'язані з поданням змін у стані середовища, що мають важливе значення для моделювання прогнозів [100-104]. Існування об’єкта в часі описується трьома взаємопов’язаними поняттями: присутній (або пов’язаний із безпосередньою присутністю) (presential), постійний або стійкий (persistent) і тривалий, безперервний (perpetuant). Presentials - це індивіди, які повністю присутні в певний момент часу. Presential - це стан об’єкту в конкретний момент часу. Persistent - це універсал, який інстанціює presential. Ця інстанціація відбувається в конкретний момент часу. Persistent відповідає сутності, яка може змінюватися і зберігати свою ідентичність (це сукупність presentials в декількох моментах часу). Perpetuant - це об’єкт, що інстанціюється persistent об’єктом і описує конкретного індивіда, який змінюється з плином часу.

Топоїд – базова онтологічна категорія GFO, що подає обмежену область простору. Він використовується для моделювання просторових локацій і може змінювати розмірність (точки, лінії, поверхні, об'єми). Топоїди мають просторову протяжність, розмірність і межі, що дозволяє описувати їхнє розташування. Вони беруть участь у просторових відношеннях, таких як примикання, перекриття та відокремлення, і інтегруються з іншими онтологічними категоріями GFO, зокрема об'єктами та процесами [105]. Хроноїд – онтологічна категорія GFO, що подає обмежену область часу. Він використовується для моделювання часових інтервалів і подій, характеризується часовою протяжністю та межами. Хроноїди відіграють ключову роль у часовій онтології, забезпечуючи опис часових

відношень, таких як перекриття, пріоритети та спадковість. Вони інтегруються з іншими категоріями GFO, зокрема об'єктами та процесами [106]. Конфігуроїд – онтологічна категорія GFO, що подає структуровані конфігурації сутностей. Він описує спосіб організації та зв’язки між частинами структури, маючи композиційний характер і реляційні властивості. Конфігуроїди беруть участь у структурних відношеннях, таких як «частина-ціле» та суміжність, і слугують абстрактною моделлю для представлення складних структур.

Ситуоїд у GFO подає ситуації або контексти [107] і є ключовим для моделювання контекстуальних аспектів реальності. Він має контекстний комплекс, що інкапсулює сутності та їх взаємозв’язки, часові та просторові межі, а також динамічну природу, що відображає зміни у часі. Ситуоїд Su може бути визначений через його мету Gl і розглядатися як перехід між двома граничними ситуаціями ($Sit_{st}^{su}, Sit_{end}^{su}$), а саме початковим станом і передбачуваним цільовим станом. Передбачається існування декількох проміжних ситуацій між цими обмежувальними станами. Ситуації мають ситуоїдну конфігурацію в конкретні моменти часу. Ситуоїд визначається метою або завданням і розглядається як перехід між початковою та цільовою ситуаціями, з можливими проміжними станами. Він інтегрується з іншими онтологічними категоріями GFO, такими як об’єкти, процеси, топоїди та хроноїди, маючи асоційований просторовий і часовий вимір, що змінюється під час його існування.

Інтелектуальні системи безпеки житлових комплексів потребують прогностичних механізмів для аналізу потреб користувачів, ухвалення рішень і планування. ІА у «розумному» житловому кварталі має обмежений набір завдань у межах конкретного часу та простору, що робить його функціонування менш ресурсозатратним. Фреймворк передбачення та планування розглядається з точки зору ситуаційної обізнаності ІА [108]. Оскільки передбачення та моделювання майбутнього є ключовими функціями ІА [109], його структура представлена як сукупність взаємодіючих функціональних модулів.

Основні припущення та вимоги до системи передбачення та планування ґрунтуються на забезпеченні базових концептуальних компонентів онтологією

GFO, що дозволяє організувати планування і моделювання у вигляді ситуативної динаміки в межах ситуоїдів. У фреймворку прогнози та моделі спираються на емпіричні знання про ситуативну динаміку в аналогічних ситуаціях, причому знання і моделювання організовані контекстно, де контекст визначається як конкретне завдання або мета, що досягається в певному середовищі. Специфікація ситуації включає лише ті елементи, що мають безпосередній стосунок до цього контексту. Фреймворк підтримує процедуру планування виконання завдань і досягнення цілей, що дозволяє також моделювати розвиток ситуацій, у яких агент перебуває у стані простою. Оновлення моделей знань відбувається на основі порівняння передбачуваних ситуацій з реальними, де інформація про реальний стан надходить безпосередньо із сенсорів. Важливим аспектом є підтримка різних рівнів деталізації та специфіки планування, що дає можливість компенсувати нестачу інформації та поступово здійснювати самокоригувальний рух до мети. Фреймворк передбачає багатовимірне моделювання з урахуванням різних сценаріїв розвитку ситуацій у майбутньому, а також використання історичної інформації про минулі ситуації для виявлення закономірностей і прийняття рішень.

3.1.2 Використання ситуоїдів для подання та моделювання динаміки зміни ситуації

ІА в кожен момент часу виконує поставлене завдання Tsk. Це завдання виконується в часі і просторі, а тому має моменти початку t_{st}і завершення t_{end}. Навіть якщо такий агент нічого не робить, він усвідомлює це і здійснює спостереження за навколишнім середовищем, передбачаючи природні зміни, що відбуваються в спостережуваному середовищі. Автоматизація управління сервісами може працювати з кількома завданнями та проблемами паралельно. З метою спрощення виключимо з аналізу координацію різних завдань і зосередимося на прогнозах і плануванні в межах одного завдання.

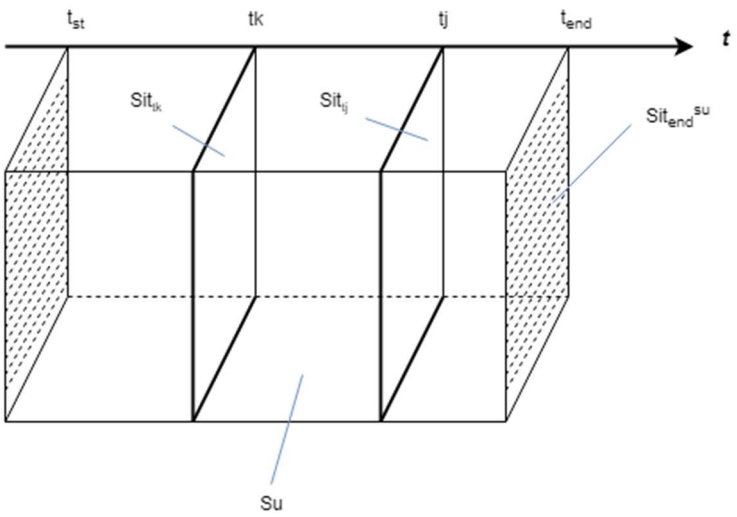


Рис. 3.1 Подання ситуативної динаміки у ситуоїді.

Ситуоїдний об’єкт GFO є зручним для подання такого завдання, оскільки динаміка його виконання та прогнозування майбутніх станів може бути подана у вигляді послідовності ситуацій усередині ситуоїда. Кожен ситуоїд Su обмежений ситуаціями. Стартова ситуація Sit_{st}^{su} відповідає стану, коли було видано завдання. Ситуація закінчення Sit_{end}^{su} відповідає стану, коли завдання виконано або від нього відмовилися. Початкова і кінцева ситуації проектуються на хроноїд Ch , в даному випадку – на часову шкалу, де розміщуються ситуації. Між початковою і кінцевою ситуаціями в ситуоїді існує ряд проміжних ситуацій $(Sit_{t1}, Sit_{t2}, .., Sit_{tk})$, які також приєднані до часової шкали (Рис.3.1). Розглядаються тільки ті ситуації, які представляють певний інтерес для процесів прогнозування або планування. Крім завдання, ситуоїд створений для досягнення конкретної мети або вирішення виявленої проблеми. Процес моделювання ситуацій у всіх випадках схожий. Кожен створений ситуоїд розглядається як єдине ціле в контексті виконання завдання в поточному середовищі. Контекст для ситуацій забезпечується поточним станом середовища і поточним наміром у виконанні в ньому завдання $Gl_{int,tc}$. Модель середовища агента $Cm'_{env,tc}$ надається управлінню інфраструктурою центру опрацювання даних як частина моделі середовища $Cm_{env,tc} \supseteq Cm'_{env,tc}$ для поточного часу t_c. Контекст формує і обмежує кількість елементів, що входять в модель ситуації,

використовуючи тільки ті, що мають відношення до наміру в поточний момент часу.

$$Cm_{con,tc} = (Cm'_{env,tc}, Gl_{int,tc}, t_c) \quad (3.1)$$

Процес виконання завдання моделюється у вигляді послідовності ситуацій $(Sit_{t1}, Sit_{t2}, .., Sit_{tk})$, і відповідної послідовності конфігурацій $(Cf_{t1}, Cf_{t2}, \dots, Cf_{tk})$. Кожна конфігурація в послідовності представлена у вигляді графа знань:

$$Cf_{ti} = (SV_{con}, SE_{rel}, t_i) \quad (3.2)$$

де SV_{con} - множина вузлів, що відповідають об'єктам і SE_{rel}- множина відношень, що використовуються при уточненні ситуації. Як об'єкти, так і відношення класифікуються відповідно до онтології системи безпеки. Конкретні конфігурації в формулі (3.2) можуть бути різними, що дозволяє відображати динаміку конфігурацій ситуації в процесі виконання завдання. У кожному ситуоїді можна вказати поточну ситуацію, минулі ситуації та кількість прогнозованих ситуацій. Всі ці ситуації враховуються при ухваленні рішень щодо виконання завдання. Зберігаються лише ті ситуації, що пов'язані з важливими подіями або змінами в процесі виконання завдання. Всі інші відновлюються/апроксимуються за потреби з використанням наявних ситуаційних знань. Збереження історичної інформації відбувається для можливості виявляти закономірності в історичних даних, які можуть впливати на рішення та прогнози. Інтелектуальна система створює тільки одну версію поточної ситуації. Вона формується як оновлення прогнозованої ситуації та даних моделі середовища. Ці дані вносять корективи в проектовану модель, отримані на основі знань, отриманих з досвіду.

3.1.3 Подання та опрацювання знань, моделювання ситуацій

Основою для прогнозів слугують емпіричні знання. Ці знання зберігаються контекстуально, тому ключем до пошуку є схожість контексту. Коли інтелектуальна система проводить пошук інформації в базі знань, вона видобуває знання про виконання аналогічного завдання в аналогічному середовищі. Після встановлення подібності, інтелектуальна система робить зіставлення між

ситуацією і зразком знань. Таким чином забезпечується доступ до знань, представлених зразками.

Тому повинна бути реалізована функція F_{sim}, що вимірює відстань між поточним контекстом (1) і ключем-контекстом $\left(Cm_{env}^{kb}, Gl_{int}^{kb}\right)$ в базі знань. У процесі пошуку значення цієї функції повинно бути зведено до мінімуму

$$F_{sim}: \left(\left(Cm'_{env,tc}, Gl_{int,tc}\right), \left(Cm_{env}^{kb}, Gl_{int}^{kb}\right)\right) \rightarrow min \qquad (3.3)$$

Знання зберігаються у вигляді структурованих шаблонів (патернів), які містять інформацію про умови їхнього застосування, зокрема винятки та варіації, що залежать від навколишнього середовища. Шаблони знань подаються у вигляді концептуальних моделей як зважені часові графи знань. Ваги в графі знань використовуються для задання важливості/актуальності конкретних частин графа в конкретній ситуації. Часовий вимір графів допомагає спроектувати застосування знань у майбутньому та розмістити їх на часовій шкалі ситуоїда.

Моделювання переходів між ситуаціями є важливою частиною прогнозування майбутніх ситуацій. Ці переходи пов'язані зі зміною параметрів, викликаних різними причинами. Переходи можуть бути природними і відбуватися без втручання агента, або вони можуть бути спланованими, включаючи проактивні дії агента.

Моделюємо передбачення змін у вигляді двох ситуацій і дій/подій, що ведуть від однієї до іншої (Рис.3.2).

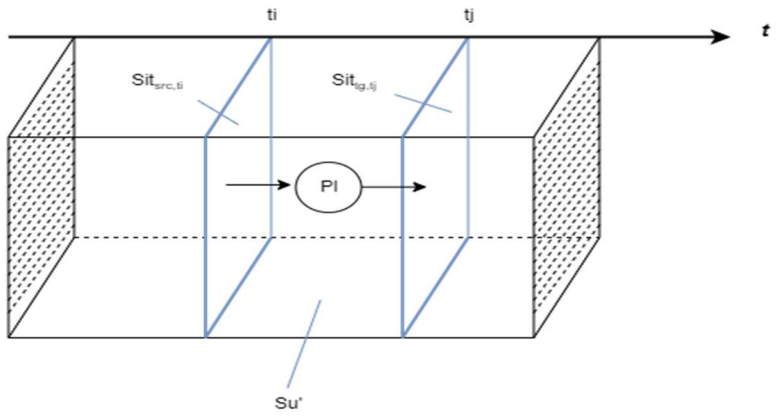


Рис. 3.2 Моделювання змін між ситуаціями

Ситуація джерела $Sit_{src,ti}$ розглядається як відправна точка для змін, цільова ситуація $Sit_{tg,tj}$ є кінцевою точкою змін. Немає обмежень щодо того, наскільки далеко в часі знаходиться цільовий стан. Єдине обмеження полягає в тому, що він повинен бути після вихідної ситуації ($tj > ti$). Після конкретизації вихідної та цільової ситуації інформаційна система визначає дію, процес або подію зі своїх знань, отриманих на основі досвіду, які можуть змінити стан джерела на цільовий стан. Цю дію або процес ми узагальнено позначимо як план Pl. Ці передбачені зміни можна розглядати як ситуоїд Su', що повністю міститься в початковому ситуоїді Su: $Su \supseteq Su'$. Він подається у вигляді кортежу

$$Su' = (Sit_{src,ti}, Sit_{tg,tj}, Pl, Cm_{con}) \quad (3.4)$$

де Pl це план. У випадку з ситуоїдом Su', що відповідає найменшій зміні, яка відображає поточний намір, замість плану ми вказуємо дію Ac. Контекстна інформація, що відображає залежність від середовища і мети, подана Cm_{con} .

При створенні завдання спочатку виконується операція передбачення між початковою і кінцевою границями ситуоїда. Інтелектуальна система моделює пошук процесів, що виконували аналогічні завдання в минулому. Такі знання забезпечують виконання процесу план Pl, досягнення проміжних станів і підтвердження здійсненності поставленого завдання. Далі ситуоїд розбивається на проміжні ситуації з використанням наявних знань. Таким чином будується план Pl досягнення мети, заданої кінцевою ситуацією. Цей план може бути поданий складним графом, таким як діаграма Ганта, що включає декілька проміжних ситуацій. Альтернативно, інтелектуальна система може фіксувати тільки початкові і кінцеві ситуації і відзначати деякі проміжні стани, без детального розроблення плану. Однак кожен раз передбачення включає в себе розгляд найменшого переходу між поточною і наступними ситуаціями. Таку зміну можна уявити як результат природного перебігу подій (всі зміни викликані природними причинами, ніяких дій з боку агента не виконується) або деякого конкретного набору зовнішніх подій, або деяких дій, виконаних розумним агентом. Інтелектуальна система моделює створення і аналізує декілька

майбутніх ситуацій, порівнює їх і підбирає найбільш придатний для агента порядок дій.

Зміни між ситуаціями подано у вигляді переходів між точками в багатовимірному просторі параметрів об’єкта в просторі концептів [110]. Вони також можуть бути вказані як зміни конфігурацій між ситуаціями, визначаючи динаміку конфігурації. У процесі прогнозування та планування можливе відстеження траєкторії зміни параметрів. У запропонованому фреймворку передбачаємо наявність декількох траєкторій. Траєкторія змін щодо конкретного об’єкту або його параметрів - зміна параметрів (конфігурацій) об’єкта або його відношень між декількома проміжними ситуаціями. Наприклад, така траєкторія може відображати пересування людини по житловому комплексу.

Траєкторія розвитку подій, коли ситуація розглядається як єдине ціле, як цілісна модель. Ця траєкторія відстежує зміни між ситуаціями. Вона використовується для передбачення багатоваріантного розвитку ситуацій в межах одного і того ж ситуоїда. Інколи потрібне розроблення декількох ситуаційних траєкторій, щоб пристосуватися до можливих зовнішніх подій, і створення планів на випадок непередбачених обставин. Іншим випадком використання багатоваріантного аналізу ситуації є передбачення і порівняння наслідків дій різних агентів (Рис.3.3). Траєкторія руху ситуоїдів використовується, коли виконання певного завдання генерує інші, які не пов'язані або не містяться в батьківському завданні та ситуоїді. У цьому випадку агент оперує залежними проектами і завданнями, кожне з яких представлене в своєму власному ситуоїді.

Інтелектуальна система будує декілька моделей передбачення і отримує корисну інформацію при їх порівнянні. Першою такою моделлю є модель спостерігача. Вона не передбачає дій з боку інтелектуального агента, а лише слідування природному перебігу подій. Іншою моделлю є модель наміру, що моделює результат негайної дії агента. Моделі намірів завжди будуються до того, як буде виконана реальна дія, щоб передбачити дії. Моделі довгострокового

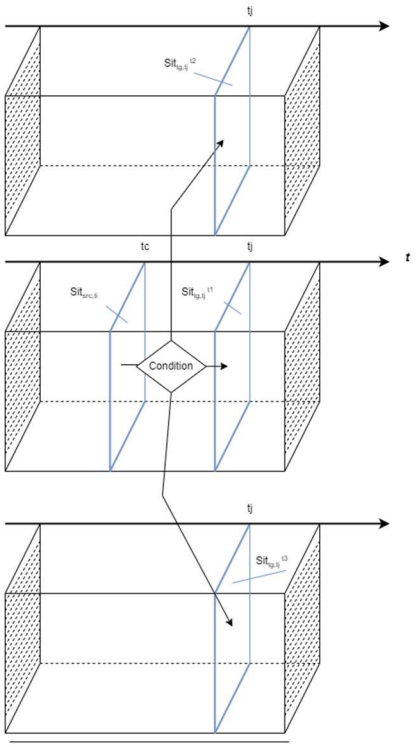


Рис. 3.3 Багатоваріантний аналіз з траєкторіями ситуацій

планування мають проміжні ситуації і цілі. Вони часто не вказуються детально на старті і модифікуються в процесі виконання завдання.

3.1.4 Виконання дій та навчання з використанням зворотному зв'язку

Модель поточної ситуації будується на основі моделі передбачення. Ця модель оновлюється з використанням даних з моделі середовища, наданих центром управлінню інфраструктурою центру опрацювання даних. Під час цієї

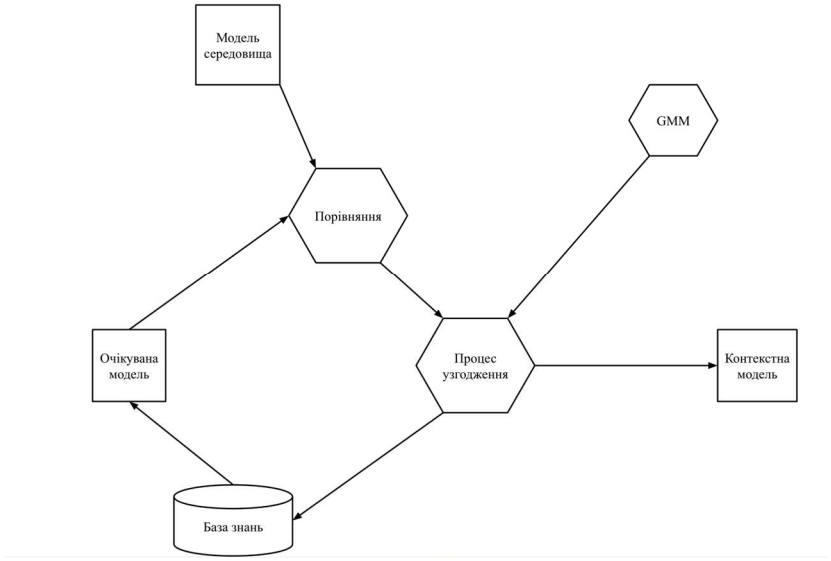


Рис. 3.4 Оновлення моделі на реальних даних

операції з центру передаються лише відмінності між очікуваною конфігурацією та реальною конфігурацією. Це дозволяє зменшити трафік даних і підвищити продуктивність (Рис.3.4).

Далі інтелектуальна система визначає модель наміру Sit_{tg} для переходу на наступну ситуацію і шукає в базі знань можливі дії, що призводять до цієї ситуації. На цьому етапі може бути побудовано декілька моделей $SSit_{tg} = \{Sit_{tg}\}$, включаючи модель спостерігача або різні варіанти моделей наміру з використанням різних дій. Наслідки таких дій прогнозуються у вигляді траєкторій ситуації. Далі обирається найбільш ефективний варіант дії з використанням функції вибору F_{sel} і критеріїв вибору Cr.

$$F_{sel}: \left(SSit_{tg}, Cr\right) \rightarrow min \tag{3.5}$$

Результат дій тестують, спостерігаючи за змінами в моделі середовища. Відзначаються помилки і відхилення від очікуваних результатів і плануються відповідні коригувальні дії, що змінюють прогнози майбутньої ситуації.

Оцінюються відмінності між побудованими передбаченнями та реальними даними. У випадку, якщо вони не можуть бути компенсовані простою зміною параметрів шаблонів, що використовуються для передбачення, шаблони оновлюються, або створюється новий шаблон у базі знань для поточного контексту. Таким чином, інтелектуальна система постійно узгоджує шаблони своєї бази знань з реальним зворотним зв'язком і оновлює свої знання (Рис. 3.4).

3.2 Проектування онтології інтелектуальної системи безпеки житлового комплексу

3.2.1 Основні вимоги і припущення використання керівних принципів при проектуванні онтології

Аналіз сучасних тенденцій в проектуванні онтологій дозволяє сформулювати основні вимоги і припущення, які використані як керівні принципи при проектуванні онтології для інтелектуальної системи безпеки житлового комплексу. Розглядаємо онтологію як програмний артефакт, документ, що підтримує розвиток інтелектуальної системи безпеки. Вона містить

мінімально необхідне число понять і відношень, що відображають функції системи в сучасному стані її розвитку. Онтологія оновлюється, коли створюється нова версія інтелектуальної системи, що володіє більшою функціональністю, в результаті чого з'являється нова версія онтології.

Створена онтологія є локальною, але базується на фундаментальній 4D-онтології, що дозволяє моделювати просторово-часові явища. Це забезпечує її розширюваність та інтеграцію з іншими інтелектуальними системами. Вона представлена у вигляді патернів, що полегшує розробку та передачу знань.

Інтелектуальна система безпеки працює на основі ситуаційної обізнаності, використовуючи історичні та емпіричні дані для моніторингу та прогнозування загроз. Довгострокове стратегічне планування необхідне для зменшення технічного боргу [11] та забезпечення стійкого розвитку. Функції онтології охоплюють контроль доступу, спостереження, кібербезпеку, управління кризами, екологічний моніторинг та модернізацію [112]. Аналіз літератури [113-115] дозволяє виділити декілька функціональних областей в системах безпеки, серед яких контроль доступу, спостереження та моніторинг, кібербезпека, управління реагуванням на надзвичайні ситуації, екологічний моніторинг, технічне обслуговування та модернізація (Додаток Г. Таблиця Д.2).

Змодельована інтелектуальна система безпеки як ситуаційно-орієнтована система [116], здатна виявляти, розпізнавати важливі ситуації, приймати в них рішення та вживати відповідні дії. Таким чином, ситуоїди Su і їх часові зрізи - ситуації Sit - стають центральними елементами моделювання в запропонованій системі. Робота інтелектуальної системи безпеки організована навколо опрацювання взаємопов'язаних концептуальних моделей знань. Моделі середовища Cm_{env} будуються на основі об'єктів, що розпізнаються в середовищі, і зберігають параметри цих об'єктів, що надаються давачами або зовнішніми джерелами даних. Контекстні/задачні моделі Cm_{con}– це відображення даних, що відповідають конкретному завданню, ситуації, меті. Вони формуються як підмножина моделі середовища $Cm_{env,tc} \supseteq Cm'_{env,tc}$ для поточного моменту часу

t_c. з додатковими об'єктами, що відповідають наміру $Gl_{int,tc}$, наданому за відповідним шаблоном з бази знань

$$Cm_{con,tc} = (Cm'_{env,tc}, Gl_{int,tc}, t_c) \quad (3.6)$$

Інтелектуальні агенти відстежують параметри об'єктів на основі інформації від давача або групи давачів, що інтерпретуються як параметри об'єктів з онтології On. Робота і прийняття рішень агентами залежить від набору концептуальних моделей, що описують дії, які необхідно зробити в різних контекстах. Бек-енд сервіс збирає інформацію від інтелектуальних агентів, створює і підтримує власну глобальну модель середовища $Cm_{env,gb}$. Ця модель використовується для виявлення та прогнозування ситуацій, координації дій агентів, прийняття загальносистемних рішень та подання інформації персоналу, який працює з фронт-ендом.

Система використовує емпіричні знання для виявлення ситуацій, планування та прогнозування розвитку ситуації. Коли система шукає інформацію в базі знань, вона шукає знання про аналогічну задачу в аналогічній конфігурації середовища. У випадку виявлення ситуацій, це виглядає як можливі ситуації/загрози, які можуть виникнути в контекстах, подібних до поточних. Коли подібність встановлена, система робить зіставлення між ситуацією і зразком знань. Таким чином забезпечується доступ до знань, поданих шаблонами. Тому реалізується функція F_{sim}, що вимірює відстань між поточним контекстом (3.1) і ключем-контекстом $\left(\mathrm{Cm}_{\mathrm{env}}^{\mathrm{kb}}, \mathrm{Gl}_{\mathrm{int}}^{\mathrm{kb}}\right)$ в базі знань. У процесі пошуку значення цієї функції повинно бути зведено до мінімуму:

$$\mathrm{F_{sim}}: \left(\left(\mathrm{Cm'_{env,tc}}, \mathrm{Gl_{int,tc}}\right), \left(\mathrm{Cm}_{\mathrm{env}}^{\mathrm{kb}}, \mathrm{Gl}_{\mathrm{int}}^{\mathrm{kb}}\right)\right) \rightarrow \min \quad (3.7)$$

Бек-енд сервіси отримують доступ, підтримують та оновлюють контекстну базу знань. Виявлення ситуації здійснюється службою, яка відстежує поточну модель середовища за ознаками та шаблонами, пов'язаними з ситуаціями, які могли б статися в поточному контексті $Cm_{env,tc}Cm_{con,tc}$. Для цього відстежується набір ознак (ключів). Кожен такий набір є умовою (зразком) з вагою, що відображає його важливість.

$$Cue = (Cm_{cue}, w_{cue}) \quad (3.8)$$

Інтелектуальна система безпеки використовує ситуоїди GFO для моделювання динаміки розвитку ситуацій. Даний розвиток представляється у вигляді впорядкованої в часі послідовності ситуацій $(Sit_{t1}, Sit_{t2}, .., Sit_{tk})$ з відповідною послідовністю конфігурацій $(Cf_{t1}, Cf_{t2}, \dots, Cf_{tk})$. Кожна конфігурація в послідовності представлена у вигляді графа знань:

$$Cf_{ti} = (SV_{con}, SE_{rel}, t_i), \quad (3.9)$$

де SV_{con} - множина вузлів, що відповідають об'єктам і SE_{rel}- множина відношень, що використовуються при специфікації ситуації, t_i −фактор часу. Як об'єкти, так і відношення класифікуються відповідно до онтології системи. Конкретні конфігурації в їх послідовності можуть бути різними, що відображає динаміку конфігурацій ситуації в процесі виконання завдання. Переходи між ситуаціями моделюються за допомогою емпіричних знань як структур дій або подій, що викликають перехід.

Побудова структури міркувань для моделювання ситуаційної динаміки у вигляді послідовності ситуацій $Sit_1, Sit_2, \dots, Sit_m$ з використанням бази знань Knb та функцію подібності F_{sim} для сценарію - моніторинг безпеки у житловому комплексі. Поточний контекст $Cm_{con,tc}$ - стан середовища, зафіксований у момент часу $t_{c;}$ Сенсори руху- виявлено активність у зоні паркінгу; камера подає зображення сторонньої особи; температура - нормальна.

Knb - база знань містить попередньо зафіксовані ситуації. Sit_1 - звичайна активність (мешканці користуються паркінгом); Sit_2 - аномалія (проникнення сторонньої особи, без авторизації); Sit_3 - пожежа (різке підвищення температури); Sit_4 - несправність обладнання (збій у роботі сенсорів). Оцінка подібності Fsim - функція подібності, яка оцінює поточний контекст щодо ситуацій із бази знань. Алгоритм дозволяє перетворити $Cm_{con,tc}$ у векторні представлення (наприклад, через числове кодування параметрів); обчислити F_{sim} $(Cm_{con,tc}, Sit_i)$ для кожного Sit_i за допомогою обраної метрики (наприклад, косинусної подібності або Евклідової відстані).

Отримані результати:

$Fsim(Cm_{con,tc},Sit_1)=0.45$ (низька схожість).

$Fsim(Cm_{con,tc},Sit_2)=0.85$ (висока схожість).

$Fsim(Cm_{con,tc},Sit3)=0.25$.

$Fsim(Cm_{con,tc},Sit4)=0.30$.

Прогноз наступного стану Sitpred відбувається на основі найвищого значення F_{sim}, поточний контекст відповідає ситуації Sit_2 (проникнення сторонньої особи). Прийняття рішень передбачає, що система активує протокол для ситуації Sit_2; надсилається сповіщення охороні; камера фокусується на зоні проникнення; увімкнення додаткових сенсорів руху для точнішої локалізації. Моделювання динаміки полягає у тому, що система прогнозує, як ситуація може змінитися залежно від сценаріїв; якщо охорона прибуває, очікується повернення до Sit_1 (звичайна активність); якщо активність у зоні збільшується, можливий перехід до іншої ситуації, наприклад Sit_4 (несправність сенсорів). Аналіз результатів відбувається після завершення події, система зберігає нові дані в базу знань Knb; оновлює параметри функції подібності F_{sim} для покращення точності у майбутньому. Переваги моделювання полягають у прогнозуванні сценаріїв розвитку подій; автоматизованому прийнятті рішень у реальному масштабі часу; постійному вдосконаленні системи на основі накопичених даних. Для оцінки ефективності сценарію вирішення проблеми можна використовувати кількісні метрики. Точність визначення ситуації оцінюється кількістю правильно визначених ситуацій із бази знань порівняно з усіма можливими ситуаціями.

$$Accuracy = \frac{TP}{TP+FP+FN} \quad , \tag{3.10}$$

де TP (True Positive) - правильно визначені аномальні ситуації; FP (False Positive) - хибні спрацьовування системи; FN (False Negative) - пропущені аномальні ситуації.

Якщо TP = 85 (правильно визначено проникнення сторонньої особи), FP = 5 (система помилково визначила аномалії); FN = 10 (система не виявила аномалій). Тоді Accuracy=85: (85+5+10)=0.85 (85%)

За критерієм час реагування визначаємо середній час, який потрібен системі для визначення ситуації та запуску відповідного сценарію.

$$T_{response}=T_{analysis}+T_{action} \quad (3.11)$$

де $T_{analysis}$ - час аналізу ситуації, T_{action} - час виконання дій ().

Якщо $T_{analysi}$ =0.5 секунд, T_{action} = 2.5 секунд, отримуємо результат

$T_{response}$=0.5+2.5=3.0 секунд

За критерієм подібність визначаємо подібність між поточним контекстом ($Cm_{con,tc}$) і ситуаціями з бази знань (Sit_i), розраховується за допомогою косинусної подібності:

$$F_{sim} = \frac{\sum_{i=1}^{n} Cm_i \cdot Sit_i}{\sqrt{\sum_{i=1}^{n} Cm_i^2} \cdot \sqrt{\sum_{i=1}^{n} Sit_i^2}} \quad (3.12)$$

Якщо $Cm_{con,tc}$=[0.8,0.6,0.9], Sit_2=[0.7,0.6,1.0], отримуємо результат

$$F_{sim} = \frac{(0.8 \cdot 0.7)+(0.6 \cdot 0.6)+(0.9 \cdot 1.0)}{\sqrt{0.8^2+0.6^2+0.9^2} \cdot \sqrt{(0.7^2+0.6^2+1.0^2)}}$$

$$F_{sim} = \frac{0.56+0.36+0.9}{\sqrt{0.64+0.36+0.81} \cdot \sqrt{0.49+0.36+1.0}} = 0.91$$

Подібність із Sit_2 дорівнює 91%.

Прогнозована точність реакції системи визначає ймовірність того, що система коректно виконує дії після розпізнавання ситуації:

$$Precision = \frac{TP}{TP+F} \quad (3.13)$$

Якщо TP = 85, а FP = 5, отримуємо результат $Precision$ = 85: (85+5)=0.944 (94.4%)

За критерієм «кількість опрацьованих ситуацій за годину» визначаємо кількість ситуацій, які система може проаналізувати та класифікувати за годину.

$$Rate = \frac{3600}{T_{response}} \quad , \quad (3.14)$$

Якщо $T_{response}$=3.0 секунди, то Rate=3600:3.0=1200 ситуацій/год.

За критерієм «вартість помилкових спрацьовувань» проведемо розрахунок втрат через хибні позитивні або негативні спрацьовування системи.

Якщо кожне хибне спрацьовування коштує 10 USD, а FP = 5.

$$Cost_{FP} = FP \cdot Cost_{single} \quad (3.15)$$

Вартість помилкових спрацьовувань складає $Cost_{FP}$=5·10=50 USD.

На основі наведених кількісних метрик було оцінено ефективність сценарію, виявлено сильні сторони системи (наприклад, високу точність і швидкість реагування) та визначено області вдосконалення (зменшення хибних спрацьовувань або підвищення точності прогнозів).

3.2.2 Створення онтології інтелектуальної програмної системи безпеки

Охарактеризуємо та обґрунтуємо процес створення онтології інтелектуальної системи безпеки. На першому етапі побудови онтології важливо вибрати корпус знань, з якого походить онтологія.

Для першої ітерації інтелектуальної системи безпеки подано такий корпус як набір сценаріїв безпеки, які повинні підтримуватися системою. Кожен сценарій є ситуоїдом, що має початкову ситуацію з заданими умовами спрацьовування, послідовність проміжних ситуацій, опис дій і можливих траєкторій ситуації і кінцеву ситуацію, що завершує ситуоїд. У Таблиці 3.1 наведено як приклад 20 вибраних сценаріїв та їх опис.

На другому етапі проектування онтології виведено класи і відношення з сценаріїв. GFO включає в себе такі категорії, як об’єкти, процеси, ролі, відношення і час, кожна з яких може бути використана для моделювання компонентів в системі безпеки.

Основними процесами в системі є аутентифікація доступу, відеоспостереження та генерація сповіщень. Аутентифікація доступу охоплює біометричну автентифікацію, коли особа резидента перевіряється за допомогою біометричного пристрою, а також процес проведення карткою, коли мешканець

Таблиця 3.1 Типові безпекові сценарії

#	Назва сценарію	Опис сценарію
1	Вхід резидента за допомогою картки доступу	Мешканець проводить карткою біля воріт, яка розпізнається панеллю управління, і ворота відкриваються. Камера фіксує подію
2	Гостьовий вхід з дозволу охорони	Гість приходить, розмовляє з охоронцем на вході, а охоронець перевіряє особу гостя за допомогою відеозв’язку та надає доступ
3	Виявлення підозрілих рухів у нічний час	Давач руху фіксує рух біля периметра після настання темряви. Камери збільшують масштаб і записують. Охорона попереджена

Продовження Таблиця 3.1

4	Доступ персоналу доставки з обмеженим у часі кодом	Кур'єри використовують одноразовий код доступу на бічному вході. Код перевіряється панеллю управління, а подія записується на відео
5	Активація екстреної евакуації	Пожежна сигналізація запускає автоматичне відмикання всіх воріт і дверей для евакуації. Камери фіксують процес евакуації.
6	Відмова у вході через невірні облікові дані	Фізична особа використовує про терміновану картку доступу. Панель управління відмовляє в доступі і фіксує подію в системі. Камера фіксує спробу
7	Спроба несанкціонованого проникнення виявлена за допомогою відеоаналітики	Відеоаналітика фіксує спробу людини перелізти через ворота. Система запускає сигнал тривоги, і співробітники служби безпеки отримують сповіщення
8	Віддалене відеоспостереження охоронцями	Співробітники служби безпеки дистанційно відстежують декілька відеопотоків з камер у диспетчерській. Вони вручну втручаються, коли бачать підозрілу поведінку
9	Біометрична ідентифікація резидента	Для входу мешканець використовує сканер відбитків пальців. Система перевіряє відбиток та надає доступ, поки камера фіксує подію
10	Нічні патрулі з виявленням руху	Охоронці патрулюють територію, а їхній рух запускає відеозапис у конкретних точках. Записи з камер спостереження аналізуються на предмет аномалій
11	В'їзд автомобіля через розпізнавання номерних знаків	Автомобіль мешканця в'їжджає на територію комплексу, а система за допомогою розпізнавання номерних знаків автоматично відкриває ворота. Камери відстежують транспортний засіб
12	Нічне спостереження в умовах поганої видимості	Інфрачервоні камери знімають кадри при поганому освітленні. Відео зберігається, а охоронці отримують сповіщення, якщо незвичайний рух фіксується давачами руху
13	Запланований доступ для обслуговуючого персоналу	Працівники з технічного обслуговування мають графік роботи, а їхні картки доступу дійсні лише впродовж конкретного часу. Камери фіксують їхню активність для притягнення до відповідальності

Продовження Таблиця 3.1

14	Збій живлення та перехід на резервне	У разі відключення електроенергії система переходить на резервне живлення. Ключові точки доступу залишаються працездатними, а критичні камери продовжують запис
15	Моніторинг віддаленого доступу мешканцями	Мешканець використовує мобільний застосунок для перегляду відеозаписів оточення свого будинку в реальному масштабі часу, зафіксованих камерами спостереження
16	Доступ за допомогою охорони для VIP-відвідувачів	Приїжджає VIP-гість, охоронець особисто їх супроводжує, надаючи доступ через панель управління, фіксуючи їх рух через камери
17	Перевірка працездатності системи	Система автоматично виконує перевірку працездатності, перевіряючи працездатність усіх камер, давачів і точок доступу. Сповіщення надсилаються в разі виникнення будь-яких проблем
18	Виявлення входу двох осіб з використанням однієї картки	Відеоаналітика виявляє, що дві особи входять після того, як одна авторизована особа проводить карткою. Оповіщення надсилається охоронцям для перевірки вручну
19	Виявлено та повідомлено про несправність камери	Система виявляє, що камера більше не веде запис або була підроблена. Буде повідомлено диспетчерський центр, і буде відправлено технічне обслуговування
20	Оповіщення про порушення периметра	Давач периметра огорожі виявляє порушення, а камери поблизу автоматично панорамують і масштабують, щоб зафіксувати подію. Система сповіщає охоронців і фіксує відзнятий матеріал

використовує картку доступу для проходження верифікації. Відеоспостереження включає процес відеозйомки, коли камери записують активність відповідно до розкладу або за тригером, а також виявлення та запис руху, що запускається після активації давача руху. Використовуючи методи штучного інтелекту аналізуються відеодані в реальному часі для виявлення аномалій і незвичайної поведінки. Генерація сповіщень здійснюється у випадках спроби несанкціонованого доступу, коли невдала спроба входу викликає попередження, а також при виявленні підозрілої поведінки, що ідентифікується компонентою, реалізованою з використанням методів штучного інтелекту у режимі реального часу, після чого надсилається відповідне сповіщення. Ці процеси можна розглядати і у вигляді

ситуоїдів. У додатку Г подано опис ситуоїда «Спроба несанкціонованого доступу».

Таблиця 3.2 Відношення в онтології

#	Назва відношення	Приклад використання
1	Відслідковує	Камера стежить за мешканцями, відвідувачами та охороною.
2	Надає доступ	Панель управління надає доступ мешканцю за допомогою картки доступу або біометричного пристрою.
3	Ініціалізує, запускає	Давач руху запускає камеру для початку запису. Система виявлення аномалій штучного інтелекту запускає сповіщення при виявленні підозрілої поведінки.
4	Записує	Камера записує відеодані мешканців та гостей міста.
5	Опрацьовує дані	Агент моніторингу штучного інтелекту опрацьовує дані з каналів камер і журналів доступу.
6	Має роль	Мешканець має роль уповноваженої фізичної особи. Охоронець виконує роль наглядача за охороною.
7	Входить до складу	Камера входить до складу підсистеми відеоспостереження. Карта доступу входить до складу підсистеми контролю доступу.

Ролі безпеки використовуються для того, щоб абстрагувати об’єкти на різні категорії і використовувати їх в моделях поведінки. Ролі представляють контекстуальні функції сутностей. Роль охоронця передбачає відстеження оповіщень та вжиття відповідних заходів у разі виникнення загроз. Водночас, роль моніторингу, реалізованого з використанням методів та засобів штучного інтелекту, полягає у постійному аналізі отриманих даних для виявлення потенційних загроз. Щодо доступу, резидент виконує роль уповноваженої особи, яка має можливість в’їзду у визначений час і у визначених місцях. Відвідувач є тимчасовою особою, доступ якої має бути схвалений мешканцем або охоронцем. Обслуговуючий персонал має обмежений доступ, який регламентується певними часовими інтервалами.

Відношення, що використовуються в онтології, подані у Таблиці 3.2. Наступним кроком розвитку онтології є групування об’єктів і відношень у значущі патерни, які утворюють мову патернів системи. Ці шаблони є базовими

повторюваними взаємодіями між об’єктами і використовуються як питання компетентності для перевірки змісту онтології. Ці закономірності використовуються для виділення значущих взаємодій у системі.

У процесі розвитку онтології були виявлені патерни, що відповідають окремим сценаріям безпеки. На Рис.3.6-3.11 подані приклади таких патернів.

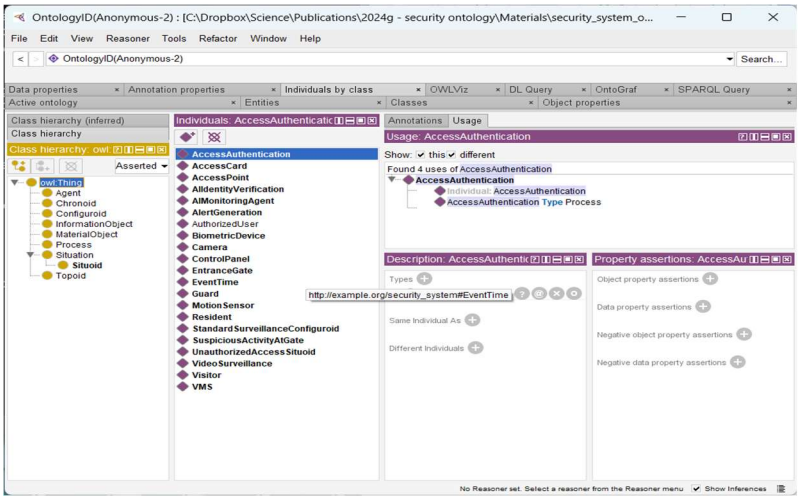


Рис. 3.5. Розроблена онтологія системи безпеки в Protégé.

Перша версія онтології інтелектуальної системи безпеки була розроблена у форматах OWL та RDF (Рис.3.5). Вона використовувалась як схема даних при розробленні інтелектуальної системи безпеки житлових комплексів.

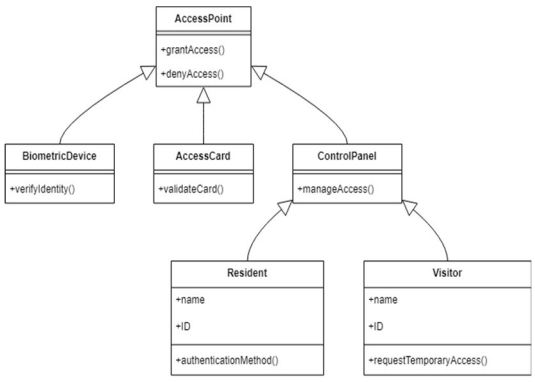


Рис. 3.6 Патерн автентифікації контролю доступу

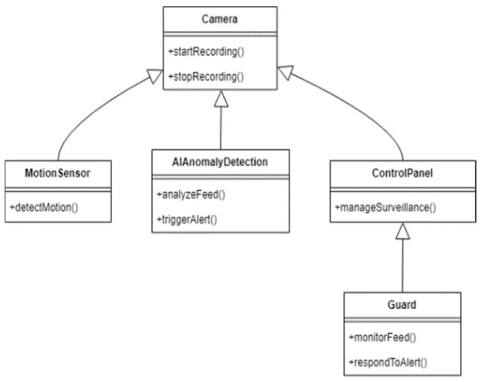


Рис. 3.7 Патерн моніторингу відеоспостереження

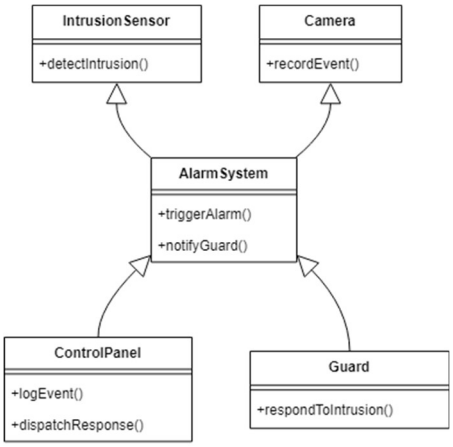


Рис. 3.8 Патерн виявлення та реагування на вторгнення

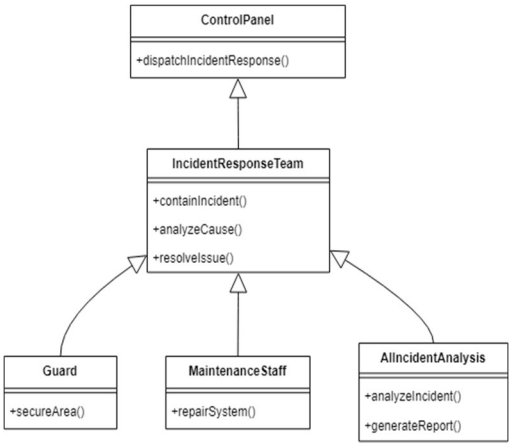


Рис. 3.9 Патерн реагування на інциденти

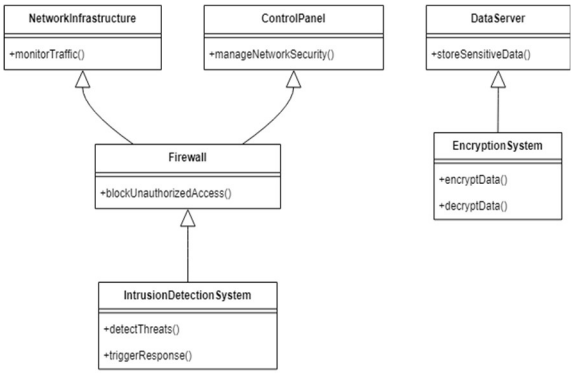


Рис. 3.10 Патерн кібербезпеки

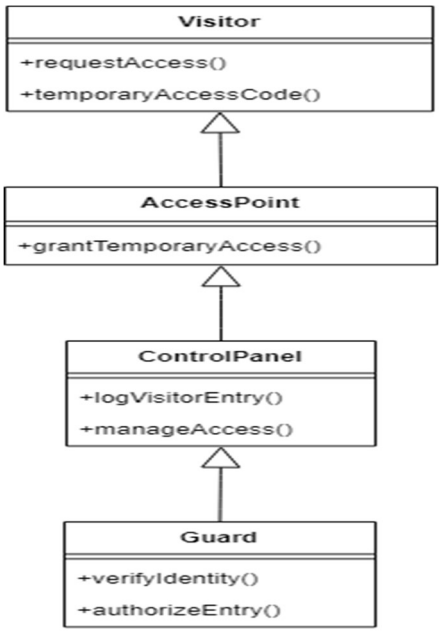


Рис. 3.11 Патерн управління відвідувачами

3.3 Метрики онтологій

3.3.1 Мережева модель проблемної області

В сучасних інформаційних системах моделюються доволі складні предметні області, з використанням мережевих моделей для повного та прецезійного опису об’єктів та їхніх взаємозв’язків. Визначення та використання при цьому відповідних метрик дозволяє оцінювати розміри системи та відстані між її елементами, що є важливим кроком в переходах від якісних до кількісних характеристик.

Оцінка відстаней між концептами в онтології зазвичай передбачає існування єдиного зв'язку між ними, але в реальності можуть існувати нетаксономічні зв'язки. Для їх моделювання будується нова метрика, що дозволяє описувати безпекові системи на рівнях від багатоквартирного будинку до міста. Онтологія інтегрується з наявними системами безпеки та управління, забезпечуючи розширення охоплення, розподілену архітектуру, ієрархічну структуру, багатоагентний підхід та динамічну адаптацію.

3.3.2. Обернено-адитивна метрика

Перехід до ширших онтологічних систем включає нові сутності, такі як територія кварталу, інфраструктура, в'їзди, персонал, системи моніторингу та управління. Ієрархічні відношення уточняються за допомогою композиції та успадкування, що дозволяє моделювати зв'язки між будівлями, мешканцями, охоронними службами та централізованими пультами управління.

Позначимо N_i – кількість переходів від концепту A до концепту B по i-му шляху, i=1, …, K, де K – кількість різних шляхів, якими можна перейти по орієнтованому графу певної онтології від концепту A до концепту B.

Визначимо відстань $d(A, B)$ між концептами A та B наступним чином:

$$\frac{1}{d(A,B)} = \sum_{i=1}^{K} \frac{1}{N_i} \tag{3.16}$$

Припустимо, що між концептами A та B є два шляхи. Перший шлях містить один перехід, а другий – два (Рис.3.12):

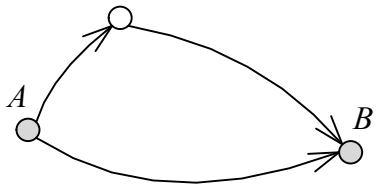


Рис. 3.12 Ілюстрація обернено-адитивної метрики

Тоді відстань $d(A, B)$ між ними:

$$\frac{1}{d(A,B)} = \frac{1}{1} + \frac{1}{2} = \frac{3}{2}, \quad d(A,B) = \frac{2}{3} \tag{3.17}$$

Аналогією до цієї метрики є правило обчислення електричного опору для послідовного і паралельного з'єднання. На основі закону Ома, для послідовного з'єднання опорів R_1 та R_2 загальний опір

$$R = R_1 + R_2$$

для паралельного з'єднання:

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} \tag{3.18}$$

Як відомо, метрика ґрунтується на понятті відстані. Відстань $d(x,y)$ – однозначна, невід'ємна, дійсна функція d: $X \times X \to \mathbb{R}$, визначена для $\forall$ x, y $\in$ X, яка задовольняє трьом аксіомам метрики:

1) $d(x, y) = 0 \Leftrightarrow x \equiv y$ (аксіома тотожності)

2) $d(x, y) = d(y, x)$ (аксіома симетрії)

3) $d(x, z) \le d(x, y) + d(y, z)$ (аксіома трикутника)

Доведемо, що ці аксіоми справджуються для вказаної метрики:

Аксіома тотожності

$d(x, y) = 0 \Leftrightarrow x \equiv y$ (аксіома тотожності)

В нашому випадку $R(A, A) = N$, де N – кількість переходів від вузла A до вузла A, N=0.

Перша аксіома – справджується.

Аксіома симетрії

$d(x, y) = d(y, x)$ (аксіома симетрії)

Слід зауважити, що в загальному випадку для орієнтованого графа не може існувати симетрії в тлумаченні другої аксіоми метрики. Тобто, не може виконуватися правило $R(A, B) = R(B, A)$, оскільки кількість переходів від вузла A до вузла B буде збігатися із кількістю переходів від вузла B до вузла A лише у тому випадку, коли є цикл $A \to B \to A$ та відстань (кількість переходів) на шляху $A \to B$ дорівнює відстані на шляху $B \to A$.

Введення пар симетричних зв'язків, наприклад, для онтології – це пара зв'язків використовує (*uses-of*) – використовується (*used-in*), дозволяє

забезпечити виконання аксіоми симетрії для запропонованої метрики в наступному тлумаченні:

$$d_{used-}\ (A,B) = d_{uses-of}(B,A) \qquad (3.19)$$

Якщо розглядати пару взаємно симетричних відношень (відношення *uses-of* симетричне до відношення *used-in*), то друга аксіома – справджується.

Аксіома трикутника

$d(x,z) \le d(x,y) + d(y,z)$ (аксіома трикутника)

Розглянемо орієнтований граф, для якого обґрунтуємо правило трикутника (Рис.3.13):

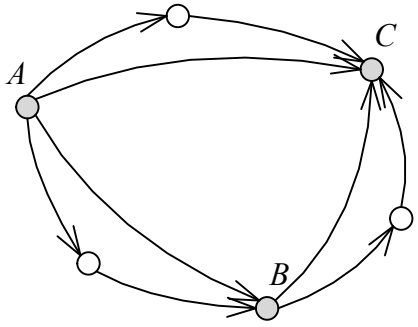


Рис. 3.13 Ілюстрація до правила трикутника для обернено-адитивної метрики

Позначимо:

$d(A,B) = d(\widehat{AB})$ – відстань між вузлами *A* та *B*. $\widehat{AB}$ – всі шляхи від вузла *A* до вузла *B*.

$d(B,C) = d(\widehat{BC})$ – відстань між вузлами *B* та *C*. $\widehat{BC}$ – всі шляхи від вузла *B* до вузла *C*.

$d(A,C)$ – відстань між вузлами *A* та *C*.

$d(\widehat{AC})$ – відстань між вузлами *A* та *C* по шляху $\widehat{AC}$, який не проходить через вузол *B*.

$d(\widehat{ABC})$ – відстань між вузлами *A* та *C* по шляху $\widehat{ABC}$, який проходить через вузол *B*.

Згідно з визначенням цієї метрики, після того, як до шляхів $\widehat{AB}$ та $\widehat{BC}$ буде додано шлях $\widehat{AC}$, відстань *d*(*A*,*C*) буде визначатися формулою

$$\frac{1}{d(A,C)} = \frac{1}{d(\widehat{AC})} + \frac{1}{d(\widehat{ABC})} \tag{3.20}$$

Тоді

$$d(A,C) = \left(\frac{1}{d(\widehat{AC})} + \frac{1}{d(\widehat{ABC})}\right)^{-1} = \left(\frac{1}{d(\widehat{AC})} + \frac{1}{d(\widehat{AB}) + d(\widehat{BC})}\right)^{-1} =$$

$$= \left(\sum_{i=1}^{K_{\widehat{AC}}} \frac{1}{N_i} + \left(\left(\sum_{i=1}^{K_{\widehat{AB}}} \frac{1}{N_i}\right)^{-1} + \left(\sum_{i=1}^{K_{\widehat{BC}}} \frac{1}{N_i}\right)^{-1}\right)^{-1}\right)^{-1} \tag{3.21}$$

Оскільки

$$d(A,B) = d(\widehat{AB}) = \left(\sum_{i=1}^{K_{AB}} \frac{1}{N_i}\right)^{-1} = \left(\sum_{i=1}^{K_{\widehat{AB}}} \frac{1}{N_i}\right)^{-1} \tag{3.22}$$

та

$$d(B,C) = d(\widehat{BC}) = \left(\sum_{i=1}^{K_{BC}} \frac{1}{N_i}\right)^{-1} = \left(\sum_{i=1}^{K_{\widehat{BC}}} \frac{1}{N_i}\right)^{-1} \tag{3.23}$$

то

$$\frac{1}{d(A,C)} \geq \frac{1}{d(A,B)+d(B,C)} \tag{3.24}$$

Звідси

$$d(A,C) \leq d(A,B) + d(B,C) \tag{3.25}$$

Що і слід було довести.

Випадок, зображений на рис. 3.13:

$$d(A,B) + d(B,C) = \frac{4}{3} \tag{3.26}$$

$$\frac{1}{d(A,C)} = \sum_{i=1}^{6} \frac{1}{N_i} = \frac{1}{1} + \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{3} + \frac{1}{4} = \frac{24+8+}{12} \tag{3.27}$$

$$d(A,C) = \frac{12}{35} \leq R(A,B) + R(B,C) = \frac{4}{3} \tag{3.28}$$

Онтологія подається у вигляді мультиграфа – оскільки допускається наявність кратних ребер - прості ребра, які мають одні й ті ж самі кінцеві вершини. Іншими словами, дві вершини можуть бути з’єднані більш ніж одним ребром.

Проблема симетрії зв'язків між концептами – проблема симетрії зв'язків для орієнтованого графа полягає в тому, що в загальному випадку, для орієнтованого графа онтології правило симетрії не виконується:

$$d(A,B) \neq d(B,A) \tag{3.29}$$

Тут відстань між концептами визначена на основі морфологічної метрики – як кількість переходів *N* між вузлами орієнтованого графа онтології на шляху від концепту *A* до концепту *B*:

$$d(A,B) = N \tag{3.30}$$

Цю проблему можна вирішити шляхом побудови пар взаємо-обернених зв'язків, наприклад для кожного зв'язку будуємо обернений до нього зв'язок [117]. Можна виділити два підходи до побудови нетаксономічних відношень між концептами в онтологіях. Перший підхід забезпечує максимальну повноту опису предметної області в онтології. Згідно цього підходу, ми якнайповніше представляємо зв'язки між поняттями предметної області за допомогою відношень між концептами в онтології.

Якщо розглядати не таксономічні відношення, то кожна пара концептів, які представлені сусідніми вузлами в семантичній мережі онтології, буде представлена парою зв'язків *used-in* та *uses-of*.

При цьому велика кількість зв'язків між концептами робить онтологію занадто «громіздкою», що ускладнює її практичне використання.

3.3.3. Проблема правила трикутника для орієнтованого графа

Як і з аксіомою симетрії, є проблеми з аксіомою трикутника: в загальному випадку правило трикутника для орієнтованого графа не виконується.

Знову, розглянемо випадок, коли відстань між концептами визначена на основі морфологічної метрики – як кількість переходів *N* між вузлами орієнтованого графа онтології на шляху від концепту *A* до концепту *B*:

$$d(A,B) = N \tag{3.31}$$

Наступний приклад (Рис. 3.14):

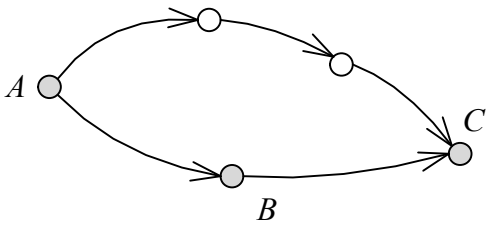


Рис. 3.14 Ілюстрація до проблеми правила трикутника для морфологічної метрики в орієнтованому графі

Для цього прикладу

$$d(A,B)=1, d(B,C)=1, d(A,C)=3 \tag{3.32}$$

Очевидно, що у цьому випадку правило трикутника

$$d(A,C)\leq d(A,B)+d(B,C) \tag{3.33}$$

– не виконується.

Природня метрика [118] дозволяє вирішити цю проблему: відстань між вузлами визначена як $d(A,B) = \min\{N_i\}$, де N_i – кількість переходів від концепту A до концепту B по i-му шляху, $i=1\ldots,K$, де K – кількість різних шляхів, якими можна перейти по орієнтованому графу певної онтології від концепту A до концепту B.

Тоді, враховуючи, що від вузла A до вузла C існує 2 шляхи: шлях $\widehat{ABC}$, який проходить через концепт B, та шлях $\widehat{AC}$, який не проходить через концепт B, і що $d(A,B)=1$, $d(B,C)=1$,

$d\left(\widehat{ABC}\right) = 2,\ d\left(\widehat{AC}\right) = 3$, $d(A,C)=\min\{ d\left(\widehat{ABC}\right), d\left(\widehat{AC}\right)\}=\min\{2,3\}=2$,

отримаємо правило трикутника

$$2=d(A,C)\leq d(A,B)+d(B,C)=2 \tag{3.34}$$

Ще одним варіантом вирішення проблеми може стати введення оберненоадитивної метрики, яка теж враховує ситуацію, при якій є два шляхи від концепту A до концепту C.

Тоді d(A,B)=1,d(B,C)=1

$$\frac{1}{d(A,C)} = \frac{1}{d(\widehat{ABC})} + \frac{1}{d(\widehat{AC})} = \frac{1}{d(A,B)+d(B,C)} + \frac{1}{d(\widehat{AC})} = \frac{1}{2} + \frac{1}{3} = \frac{5}{6} \tag{3.35}$$

Звідси

$$d(A,C) = \frac{6}{5} = 1{,}2 \tag{3.36}$$

У цьому випадку правило трикутника виконується:

$$1{,}2 = d(A,C) \le d(A,B) + d(B,C) = 2 \qquad (3.37)$$

Якщо є проблема циклічних зв'язків між концептами, таку ситуацію слід розглядати як помилку в поданні онтології. Проблема виникає із-за наявності в онтології рекурсивних циклів. Граф зв'язків між такими концептами зображено на Рис.3.15:

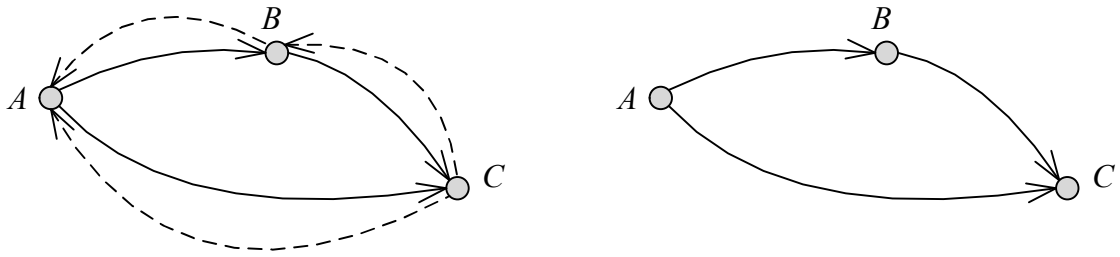


Рис. 3.15 Ілюстрація семантичних мереж, що мають рекурсивні зв'язки між концептами; ліворуч – «використовує» (uses-of) та «використовується в» (used-in); праворуч – лише «використовує» (uses-of)

Тоді відстань $d(A,B)$ між концептами A та B в семантичній мережі можна визначити так:

$$\frac{1}{d(A,B)} = \frac{1}{d_{AB}(A,B)} + \frac{1}{d_{BCA}(B,A)} \qquad (3.38)$$

де $d_{AB}(A,B)$ – відстань від вузла A до вузла B по шляху $A \rightarrow B$, $d_{BCA}(B,A)$ – відстань від вузла B до вузла A по шляху $B \rightarrow C \rightarrow A$. Визначена таким чином відстань $d(A,B)$ враховує наявність двох шляхів, які з'єднують вузли A та B. Запропонований підхід за своєю сутністю дозволяє зробити ще один крок до автономної побудови систем алгоритмічних алгебр, які містили б оператори згортання структур проблемних областей для автономної побудови так званих знаннєвих клонів. Засоби алгоритмічної формалізації і синтезу знань предметних областей є тріадою – абстракції, біології та екології програмування. Як абстрактний механізм використовується алгебраїчний апарат теорії клонів. Біологічна компонента відповідає за розповсюдження отриманих алгоритмічних знань на інші задачі. Екологічна компонента призначена для формування інструментальних засобів підтримки методів абстрактної та біологічної

компонент, що складають теорію клонів. В рамках екологічної компоненти пропонуються різні інтерпретації алгоритмічних операцій, дослідження паралелізмів та механізмів виводу алгоритмічних знань [119]. В подальшому можна продовжити дослідження методів та засобів «клонування» знань при побудові інтелектуальних програмних систем безпекового типу в їх розвитку від систем будинків, розрахованих на проживання однієї сім'ї до багатоквартирних будинків, кварталів, мікрорайонів, житлових комплексів та загалом міст.

Висновки до розділу 3

Фреймворк аналізу ситуацій базується на графах знань, що відображають контекст середовища та наміри інтелектуального агента. Прогнозування здійснюється на основі емпіричних шаблонів, оновлюваних у процесі узгодження з реальними даними. Для моделювання часових та просторових змін використовується онтологія GFO, а ситуоїди дозволяють описувати еволюцію ситуацій.

Фреймворк підтримує передбачення з різним рівнем деталізації та аналіз траєкторій змін, що є основою для прийняття рішень. Використання методів та засобів ШІ дозволяє системі навчатися на реальних даних, розпізнавати закономірності та адаптуватися до змін.

Розроблено онтологію для інтелектуальних систем безпеки житлових комплексів, що спрощує передачу даних між системами та покращує управління безпекою. Запропоновано обернено-адитивну семантичну метрику для точнішого аналізу онтологій.

Розділ 4. Розроблення структури та архітектури інтелектуальних систем безпеки з ситуаційною обізнаністю для житлових комплексів

4.1 Структура інтелектуальної системи безпеки житлового комплексу

Інтелектуальні системи безпеки житлових комплексів, засновані на технологіях IoT, з використанням методів та засобів ШІ, забезпечують комплексний захист мешканців, розпізнають загрози та автоматично реагують на небезпеку. Вони координують контроль доступу, відеоспостереження та аварійні системи в режимі реального часу. На відміну від систем для окремих квартир, такі рішення інтегрують управління спільними зонами, енергоспоживанням і безпекою всього комплексу. Використання інтернет-інфраструктури ОСББ зменшує витрати на опрацювання даних, підвищуючи ефективність і швидкість реагування на інциденти. Система також надає централізований контроль для ОСББ та персоналізовані сервіси для мешканців, як-от доступ до відеоспостереження та замовлення послуг через мобільні застосунки. Розроблення системи безпеки житлового комплексу проводиться на інфраструктурі та програмно-технологічній платформі компанії «АСТРА» і забезпечує формування вимог, завдань і технологічних специфікацій. На даний час реалізується етап дослідної експлуатації першої черги зазначеної системи в реальних умовах декількох ОСББ міста Львова. Структура інтегрує інтелектуальних агентів, IoT-пристрої, серверні служби та центральний блок опрацювання даних в єдину адаптивну і масштабовану систему. Архітектура враховує функціональні особливості кожної із підсистем [120].

Обмеження поділяються на технічні, інтеграційні, безпекові та законодавчі. Вони включають пропускну здатність мережі, енергоефективність IoT-пристроїв, сумісність компонентів, шифрування даних та відповідність нормативам (GDPR). Допущення стосуються інфраструктури, поведінки користувачів і системних параметрів, передбачаючи стабільну роботу IoT-пристроїв, наявність

кваліфікованих користувачів та оновлення алгоритмів. Припущення охоплюють технологічні, експлуатаційні й аналітичні аспекти, передбачаючи підтримку стандартів IoT, обчислювальну потужність хмарних сервісів, технічне обслуговування та точність аналізу загроз у 95% випадків. Ці дані використовуються для вдосконалення системи.

Інтелектуальні програмні системи, в яких здійснюються урахування ситуацій є значним поступом у розвитку методів та засобів штучного інтелекту. Розроблення інтелектуальної програмної системи, яка може ідентифікувати та аналізувати ситуації є доволі складною задачею. Такі системи, по суті, мають реалізовувати когнітивні функції, подібні до тих, що притаманні людині. Це передбачає використання контекстних знань, процеси навчання, цілеспрямовану поведінку, процедури прийняття рішень тощо. Інтелектуальна програмна система безпеки повинна володіти якостями саме такої природи, тому вона містить функції ситуаційної обізнаності в умовах невизначеності та непередбачуваності. Вона функціонуватиме в обмеженій зоні, вирішуватиме обмежену кількість завдань і може реалізувати окреслену множину ситуацій для реагування, проте постійно навчатиметься, генеруючи нові ситуації.

Розроблення структури інтелектуальної інформаційної системи безпеки передбачає прийняття рішень щодо оптимального розподілу інтелектуальних сенсорів, обчислювальних потужностей, розташування центрів прийняття рішень. При цьому беруться до уваги вимоги до ресурсів необхідних для виконання різних завдань, потребу реагування в режимі реального масштабу часу. Проведено порівняльний аналіз інформаційних систем, що реалізовані на принципах генеративного ШІ з інтелектуальними системами, які врахують реальні ситуації та сценарії (Таблиця 4.1).

Збір даних здійснюється через сенсори, такі як відеокамери, мікрофони та екологічні давачі, які можуть попередньо опрацьовувати інформацію, фільтрувати шум і виявляти ключові функції (наприклад, рух, звукові шаблони,

Таблиця 4.1 Порівняння характеристик систем, реалізованих на принципах генеративного ШІ з системами ШІ, які врахують реальні ситуації та сценарії

N п/п	Генеративний ШІ	ШІ з урахуванням ситуації
1	Зосереджується насамперед на створенні вмісту (тексту, зображень тощо) на основі сформованих шаблонів на основі великих наборів даних. Ці моделі можуть подавати вражаючі результати, їм часто бракує глибокого розуміння контексту або здатності розумно взаємодіяти з реальним світом	Виходить за рамки генерування контенту, розуміючи контекст реального світу та приймаючи рішення або вживаючи дій на основі цього розуміння. Це означає перехід від покоління пасивного сприйняття до активної взаємодії з навколишнім середовищем.
2	Діють переважно в цифровій сфері, опрацьовуючи та генеруючи повідомлення на основі текстових даних.	Призначені для взаємодії з фізичним світом, інтеграції сенсорних даних і прийняття рішень у реальному масштабі часу. Це включає не лише розуміння тексту, але й інтерпретацію візуальних, звукових і навколишніх даних для планування та реалізації відповідних дій
3	Хоча наділений можливостями створення творчих результатів, йому бракує самостійності у процедурах прийняття рішень або адаптації своєї поведінки відповідно до мінливих обставин	Включає системи, які можуть автономно адаптувати свою поведінку, приймати рішення в реальному масштабі часу та взаємодіяти з навколишнім середовищем змістовним чином, демонструючи суттєво вищий рівень інтелекту
4	Незважаючи на складність, великі мовні моделі в основному мають справу з розпізнаванням мови та шаблонів у структурованих або напівструктурованих даних	В системах цього класу реалізується мультидисциплінарний підхід, який інтегрує сучасні досягнення в галузях робототехніки, сенсорних технологій, периферійних обчислень і методів та засобів штучного інтелекту для створення систем, які здатні діяти автономно в реальних ситуаціях та сценаріях

тощо). Моніторинг у реальному часі дозволяє виявляти аномалії, наприклад, несанкціонований доступ або незвичну поведінку. У разі загрози агенти можуть автоматично виконувати дії, як-от замикання дверей, активація сигналізації чи

перенаправлення камер. Наприклад, при пожежі агент розблокує аварійні виходи та активує пожежогасіння. Аналізуючи поведінкові моделі, агенти виявляють потенційні загрози, контролюють переміщення осіб і координацію ресурсів, таких як відеокамери, дрони чи роботизоване патрулювання. Вони обмінюються інформацією та узгоджують дії для точнішого виявлення загроз. Крім того, агенти слугують посередниками між системою та операторами, подаючи дані у зручному форматі та пропонуючи відповідні рішення.

Використання інтелектуальних агентів у відеоспостереженні підвищує ефективність, масштабованість, адаптивність і швидкість реагування. Ключовим етапом упровадження таких систем є проектування їхньої структури, що визначає розподіл функцій, взаємодію компонентів та інтеграцію інтелектуальних можливостей. Особливістю реалізації проекту такого роду систем є те, що він зорієнтований на розроблення та супровід компанією-провайдером Інтернет послуг, що працює в розлогих будинкових мережах та обслуговування ОСББ. Під житловим комплексом (ЖК) в роботі розуміється утворення, яке виділене із житлового середовища як самодостатня структурна одиниця, що включає взаємопоєднані житлові та нежитлові об’єкти та об’єкти інженерної інфраструктури.

Структура інтелектуальної програмної системи безпеки формується на основі відповідних завдань. До них належать моніторинг стану території у реальному масштабі часу, відеофіксація подій у середовищі розумного житлового комплексу та зберігання відеоматеріалів, виявлення та аналіз підозрілої активності, оповіщення мешканців про виявлені загрози, підтримання двостороннього зв'язку між користувачами і відвідувачами, контроль доступу до об’єктів та приміщень житлового комплексу. Для виконання поставлених завдань необхідним є *к*омплексне покриття території відеоспостереженням, реєстрація та зберігання відео, виявлення руху та проведення аналітики. Разом з тим необхідним є забезпечення дистанційного доступ до приміщень житлового комплексу та можливість управління ними, стійкість до зовнішніх факторів та захист від вандалізму, цілодобовий контроль доступу та термінове реагування на

проблемні ситуації. Аналіз принципів побудови інтелектуальної інформаційної системи безпеки житлового комплексу показав, що вона реалізується як система з ситуаційною обізнаністю, здатна своєчасно розпізнавати загрози та реагувати на них. Ухвалення рішень базується на досвіді минулих ситуацій, що дозволяє системі передбачати можливий розвиток подій та діяти проактивно. Постійне оновлення знань і узгодження відбувається на основі аналізу даних, отриманих з великої кількості різнотипових сенсорів. Система підтримуватиме як локальний аналіз та ухвалення рішень, так і координацію загальної ситуації, забезпечуючи взаємодію всіх компонентів. Вона враховує вимоги до обчислювальних ресурсів і використовує комбінацію туманних, локальних, периферійних і хмарних обчислень для ефективного виконання різних завдань.

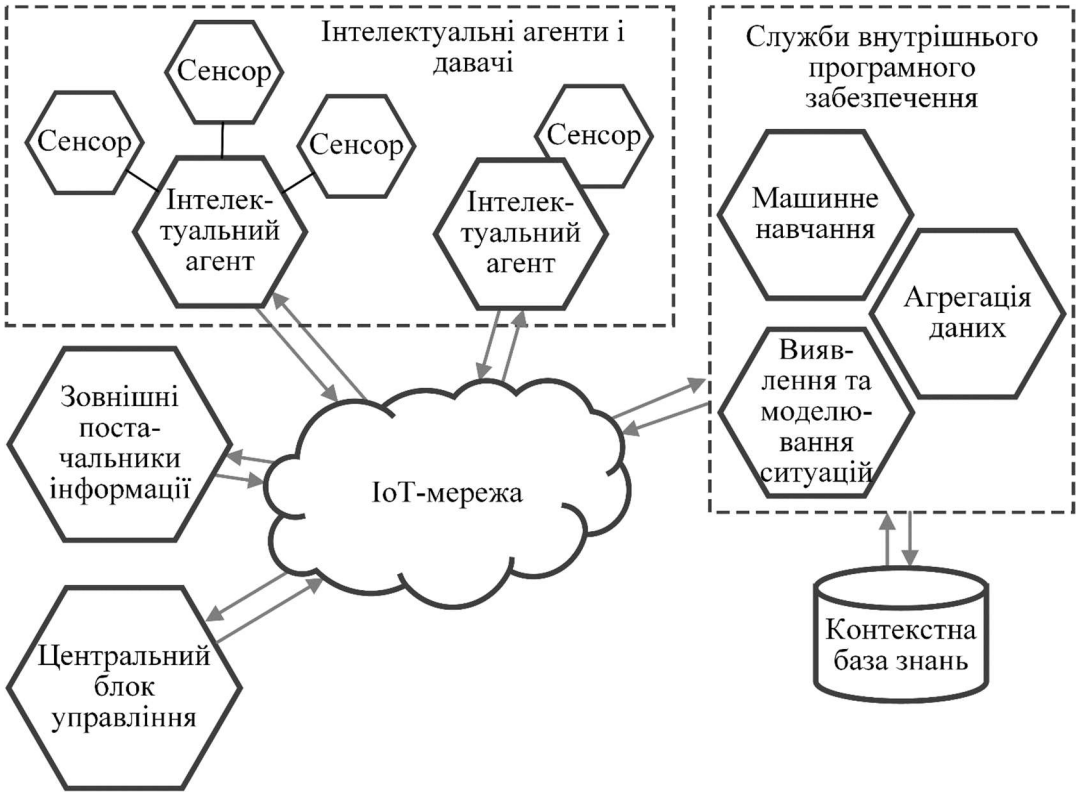


Рис. 4.1 Структура інформаційної системи безпеки з ситуаційною обізнаністю

Перш ніж аналізувати структуру запропонованої інтелектуальної програмної системи безпеки житлового комплексу, сформулюємо означення цього концепту та завдання, які вона повинна виконувати. Зазвичай під структурою розуміють пару, яка складається з множини елементів певної

природи (компонентів), а також заданого на цій множині відношення порядку. Структурою інтелектуальної програмної системи вважатимемо організовану сукупність взаємопов’язаних компонентів, які системно взаємодіючи, реалізують процеси відбору, реєстрування, зберігання, передавання, опрацювання, подання та захисту інформації. В контексті даного дослідження під структурою інтелектуальної ситуаційно-обізнаної інтелектуальної програмної системи безпеки розумітимемо організовану сукупність компонентів та множини взаємозв’язків між ними. Вони забезпечують інтелектуальні процедури моніторингу середовища, аналізу даних і управління безпекою в реальному масштабі часу на основі ситуаційної обізнаності. До її складу входять інтелектуальні агенти, служби внутрішнього програмного забезпечення, центральний блок управління, сенсори та бази знань.

Структура інтелектуальної системи безпеки житлового комплексу (Рис. 4.1) містить такі компоненти: інтелектуальні автономні агенти – орієнтовані на завдання автономні пристрої, інтегровані з інтелектуальними сенсорами, служби внутрішнього програмного забезпечення, реалізовані як хмарні сервіси, які виконують ресурсомісткі обчислення та побудовані відповідно до вимог сервісно-орієнтованої архітектури; центральний блок управління, який формує та аналізує загальну картину безпекової ситуації, використовуючи інформацію, що надається сенсорами та службами, а також знання, що подані в базах знань.

4.2 Особливості використання компонентів структури у інтелектуальних програмних системах безпеки

Інтелектуальні автономні агенти – програмні або апаратні суб’єкти, що здатні автономно діяти в середовищі, приймаючи рішення на основі отриманих даних і виконуючи завдання без втручання людини (Рис. 4.2). Такий агент наділений ознаками штучного інтелекту, що надає йому можливість аналізувати ситуації, навчатися з досвіду, реагувати на зміни в середовищі та співпрацювати з іншими агентами або системами для досягнення поставлених цілей. Інтелектуальні агенти аналізують ситуацію у середовищі на основі даних, отриманих від сенсора чи групи сенсорів, інтерпретуючи їх як параметри об'єктів

з онтології. Іноді він може реалізовувати функції, що пов'язані з автоматизованими діями в складних або динамічних умовах. Зазвичай вони взаємодіють із середовищем і сенсорами та приймають рішення на основі локальних даних.

Концепт «інтелектуальний автономний агент» подано у формі кортежу:

$$IA = (S, E, D, P, F, Act, G, AF) \quad (4.1)$$

де S – множина сенсорів, які використовує агент для отримання інформації з

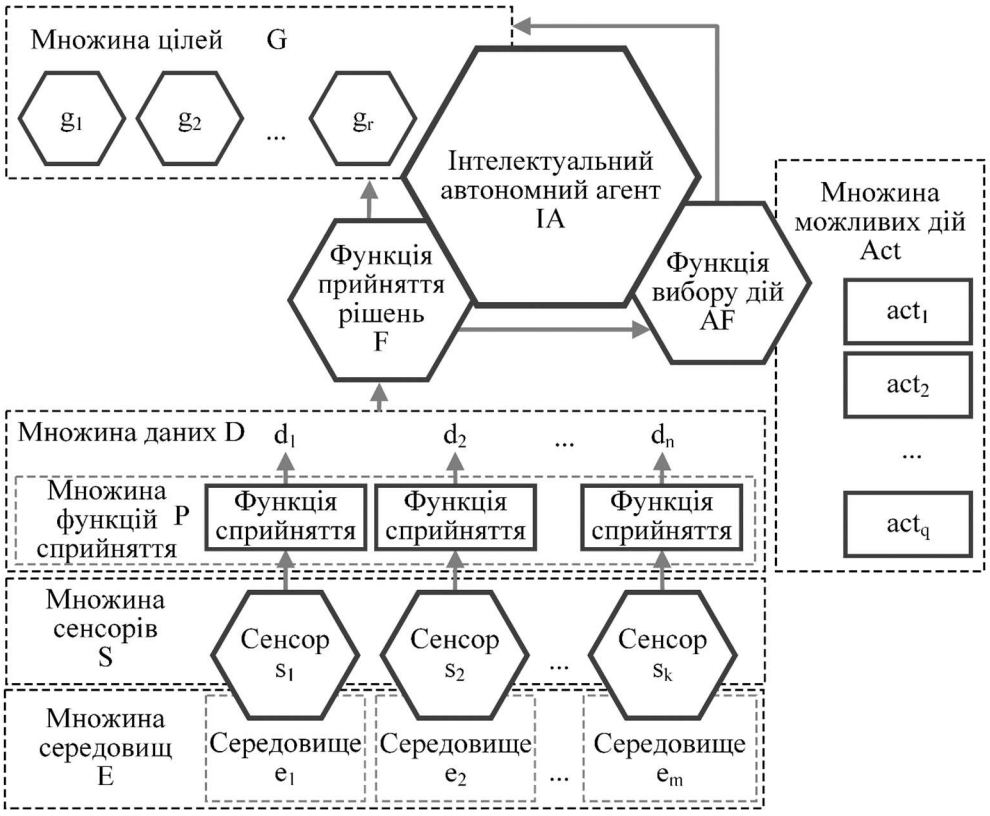


Рис. 4.2 Структура концепту «інтелектуальний автономний агент»

середовища $S=\{s_1, s_2, \ldots, s_k\}$, E – середовище, або множина середовищ, у яких агенти функціонують $E=\{e_1,e_2,\ldots,e_m\}$, D – множина даних, зібраних агентом з середовища для аналізу і прийняття рішень $D=\{d_1, d_2, \ldots, d_n\}$, P – функція сприйняття, яка визначає способи, з використанням яких агент отримує інформацію з середовища. Множина функцій сприйняття $P=\{p_1,p_2,\ldots,p_n\}$ відповідає множині даних D, F – функція прийняття рішень, Act – множина можливих дій, які може виконувати агент. $Act=\{act_1,act_2,\ldots,act_q\}$, G – множина

цілей, яких намагається досягти агент $G=\{g_1, g_2, \ldots, g_r\}$, AF – функція вибору дій, яка визначає, яку дію виконує агент для досягнення мети на основі аналізу даних.

Реалізація функції сприйняття P забезпечує трансформацію конкретних параметрів середовища (E) у дані (D), які можуть бути використані агентом для подальшого аналізу. Таким чином, кожен елемент середовища $e \in E$ відображається певним елементом даних $d \in D$. Формально це подано як $P:E \rightarrow D$, де функція P встановлює явну відповідність між станами середовища та отриманими даними. F — це функція прийняття рішень, яка забезпечує перетворення вхідних даних (D) у конкретні дії (Act). Кожен елемент даних із множини D аналізується шляхом реалізації функції F, і на основі аналізу обирає дію з множини можливих дій Act. Формально це подано як $F:D \rightarrow Act$, де функція F встановлює відповідність між кожним елементом $d \in D$ і дією $act \in Act$. Функція вибору дій AF приймає як вхідні, дані опрацьовані шляхом реалізації функції прийняття рішень F, і визначає конкретну дію з множини можливих дій Act, яку агент має виконати. $AF:D \rightarrow Act$, де D — дані, які використовуються для аналізу, а Act — обрана дія. Сенсори (S) збирають дані (D) про середовище (E). Функція сприйняття (P) забезпечує процес перетворення інформації (Рис. 4.2), отриманої з середовища у дані, які агент може використовувати, трактуючи їх як параметри об’єктів онтології. Функція прийняття рішень (F) реалізує процес аналізу цих даних і визначає оптимальні дії (Act). Функція вибору дій (AF) забезпечує виконання дій для досягнення цілей (G).

Таке подання описує основні елементи автономного інтелектуального агента, його функції та процеси взаємодії із середовищем для досягнення поставлених цілей без втручання людини.

Окремий клас складають програмні агенти, що працюють із хмарними інтелектуальними сервісами, наприклад, для розпізнавання об’єктів, ідентифікації осіб або аналізу інформації, отриманої з Інтернету. Основними перевагами таких систем є зменшення затримок при опрацюванні даних, підвищення швидкості реагування, розподіленість процесів опрацювання даних, збереження конфіденційності та оптимізацію ресурсів.

Автономні агенти виконують обчислення безпосередньо на пристроях, що знижує залежність від хмарних серверів і дозволяє швидко реагувати на локальні події (наприклад, у системах відеоспостереження чи контролю доступу). Вони адаптують використання енергоресурсів відповідно до доступних можливостей, що підвищує рівень енергоефективності. Завдяки використанню методів машинного навчання агенти вдосконалюють моделі прийняття рішень, враховуючи зміни в середовищі та поведінці користувачів.

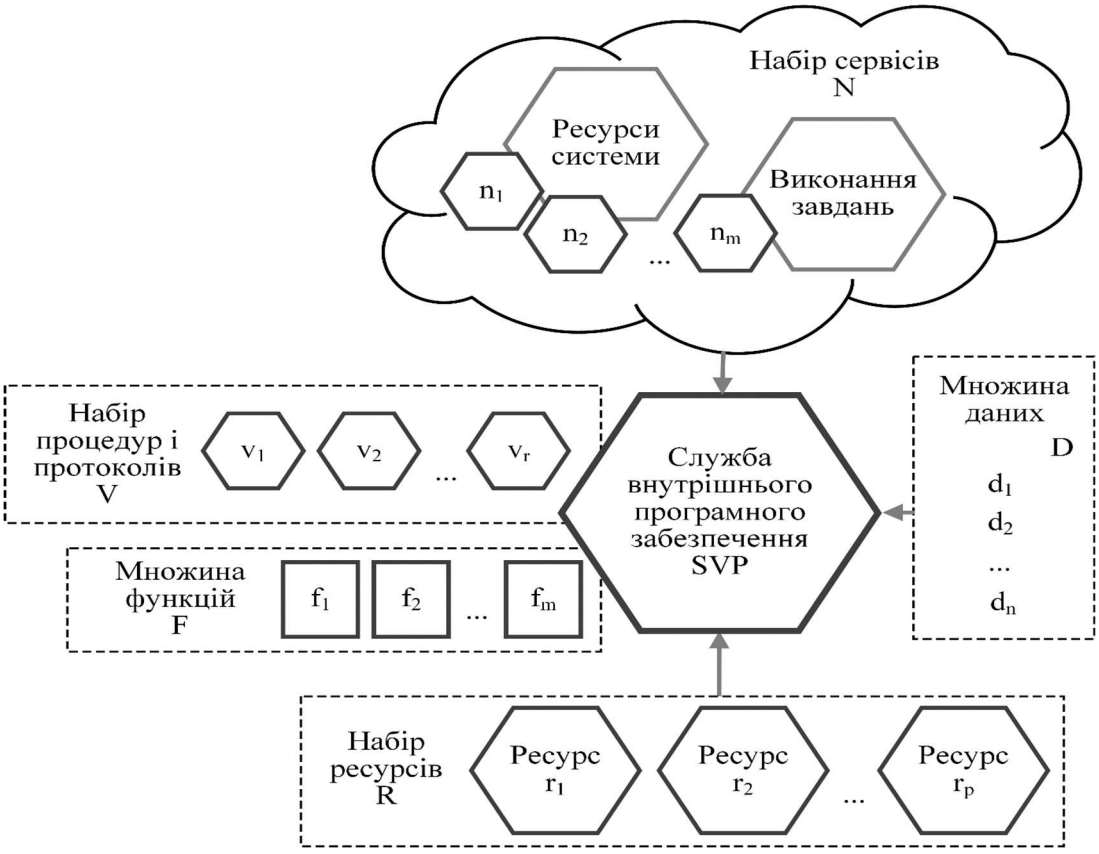


Рис. 4.3 Структура концепту «служба внутрішнього програмного забезпечення»

Програмні сервіси є важливими компонентами інтелектуальної системи безпеки. Їх використання обумовлюється тим, що виконання окремих завдань в системі вимагає значних обчислювальних потужностей, які можуть забезпечити лише виділені центри опрацювання даних. Служба внутрішнього програмного забезпечення як елемент структури інтелектуальної програмної системи безпеки житлового комплексу подається як набір програмних компонентів. Ця служба

відповідає за управління підсистемами відеоспостереження, контролю доступу та комунікацією між елементами системи безпеки. Вона опрацьовує внутрішні запити, керує доступом до ресурсів, забезпечує моніторинг і автоматизацію безпекових процесів та взаємодіє з інтерфейсами користувача для надання необхідних даних мешканцям та адміністраторам. Структура концепту «служби внутрішнього програмного забезпечення» як елемента більш загальної структури інтелектуальної програмної системи безпеки житлового комплексу подана на рис. 4.3, а формальний запис концепту подано кортежем:

$$SVP = (N, F, R, M, V), \quad (4.2)$$

де N – набір сервісів, що реалізовані як хмарні обчислення та надають ресурси для системи, забезпечують виконання завдань, пов'язаних із безпекою N $=\{n_1,n_2,\ldots,n_m\}$, F – множина функцій, що виконуються службою $F=\{f_1,f_2,\ldots,f_m\}$, R – набір ресурсів, які використовуються службою для виконання функцій (обчислювальна потужність, пам'ять тощо). $R=\{r_1,r_2,\ldots,r_p\}$, M – множина даних, що опрацьовуються та зберігаються службою для аналізу безпеки та прийняття рішень $M=\{m_1,m_2,\ldots,m_q\}$, V – набір процедур і протоколів, правил, що забезпечують коректне функціонування служби в рамках інформаційної системи. $V=\{v_1,v_2,\ldots,v_r\}$.

Проаналізуємо процеси взаємодії елементів формальної моделі. Прийом запитів здійснюється службою внутрішнього програмного забезпечення, яка отримує ресурси R з різних підсистем. Кожен ресурс опрацьовується відповідною функцією F, визначеною у відношенні V. Функції використовують або змінюють дані D для виконання певних дій. Використовуючи правила доступу V, служба визначає, які підсистеми або користувачі мають право на виконання певних функцій або отримання даних. В такому поданні відображені структура і процеси, що забезпечують роботу служби внутрішнього програмного забезпечення в інтелектуальній системі безпеки житлового комплексу. Вони включають прийом запитів, їх опрацювання, взаємодію з іншими підсистемами та забезпечення захисту даних. Одним із завдань служби внутрішнього програмного забезпечення є навчання моделі з використанням методів

машинного навчання та реалізація процедур аналізу даних. Служби внутрішнього програмного забезпечення є модульними, компонованими, переважно з використанням хмарних ресурсів та обчислень. Вони забезпечують реалізацію спеціалізованих функцій системи. Служби взаємодіють як з агентами, так і з центральним блоком управління. Програмні сервіси виконують агрегацію та злиття даних; створення моделей машинного навчання; забезпечення комунікації та координації; потокове передавання даних в реальному масштабі часу та їх зберігання; виявлення ситуації та управління сповіщеннями.

Центральний блок управління – ядро системи, яке відповідає за загальний контроль, прийняття рішень і поглиблений аналіз даних (Рис.4.4). Подамо концепт «Центральний блок управління» кортежем:

$$CBU = (MC, DN, A, W, G, Cmd) \tag{4.3}$$

де MC – множина компонентів центрального блоку, які реалізують основні функції керування системою, формують апаратно-програмну інфраструктуру для

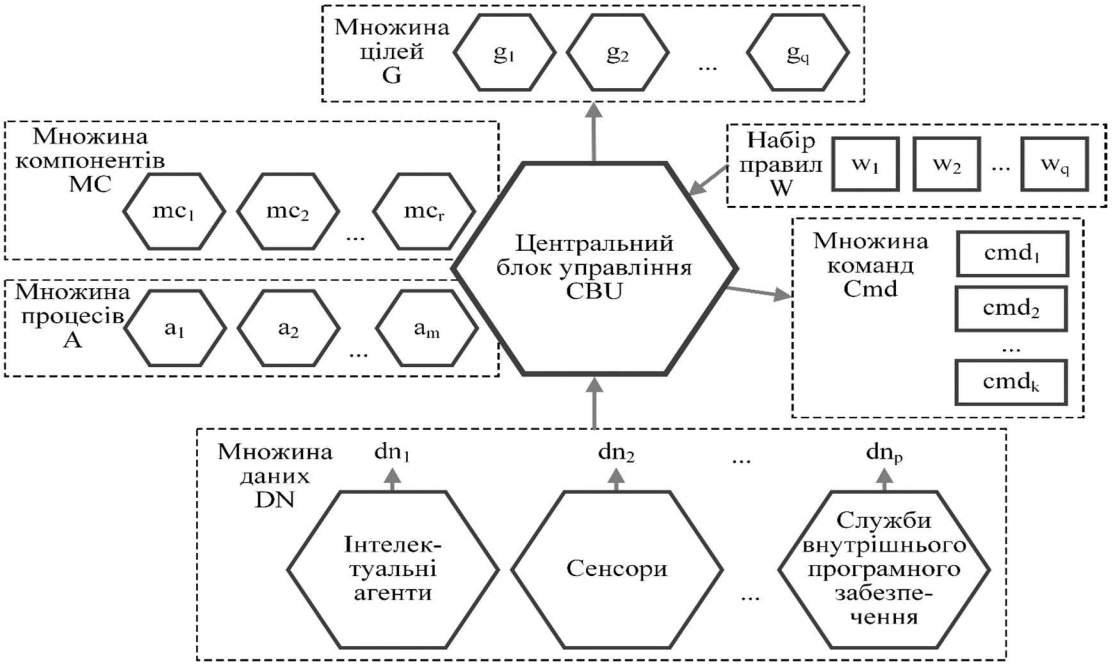


Рис. 4.4 Структура концепту «Центральний блок управління»

виконання процедур управління MC={mc_1,mc_2,…,mc_r}, DN – множина даних, що надходять від сенсорів, інтелектуальних агентів та служб внутрішнього програмного забезпечення, які використовуються для аналізу з метою

формування загальної ситуаційної обізнаності $DN=\{dn_1,dn_2,\ldots,dn_p\}$, A – множина процесів, що виконуються для аналізу ситуації, зокрема опрацювання даних та виявлення аномалій, включаючи глибинний аналіз і виявлення загроз. A асоціює кожну команду з конкретним компонентом або підсистемою, яка повинна виконати дію $A=\{a_1,a_2,\ldots,a_m\}$, W – набір правил, що визначають логіку прийняття рішень на основі аналізу даних $W=\{w_1,w_2,\ldots,w_q\}$, G – множина цілей або стратегій, яких система намагається досягти для забезпечення безпеки, що визначають загальну мету системи, зокрема, забезпечення безпеки мешканців та реагування на потенційні загрози $G=\{g_1,g_2,\ldots,g_q\}$, Cmd – множина команд, що видаються для виконання дій, які визначені як результат аналізу і прийняття рішень $Cmd=\{cmd_1,cmd_2,\ldots,cmd_k\}$.

Взаємодія складових елементів концепту передбачає збір та аналіз даних. Центральний блок управління збирає дані DN з різних підсистем. За допомогою процедур аналізу A, центральний блок аналізує зібрані дані, визначає аномалії, загрози або інші важливі події.

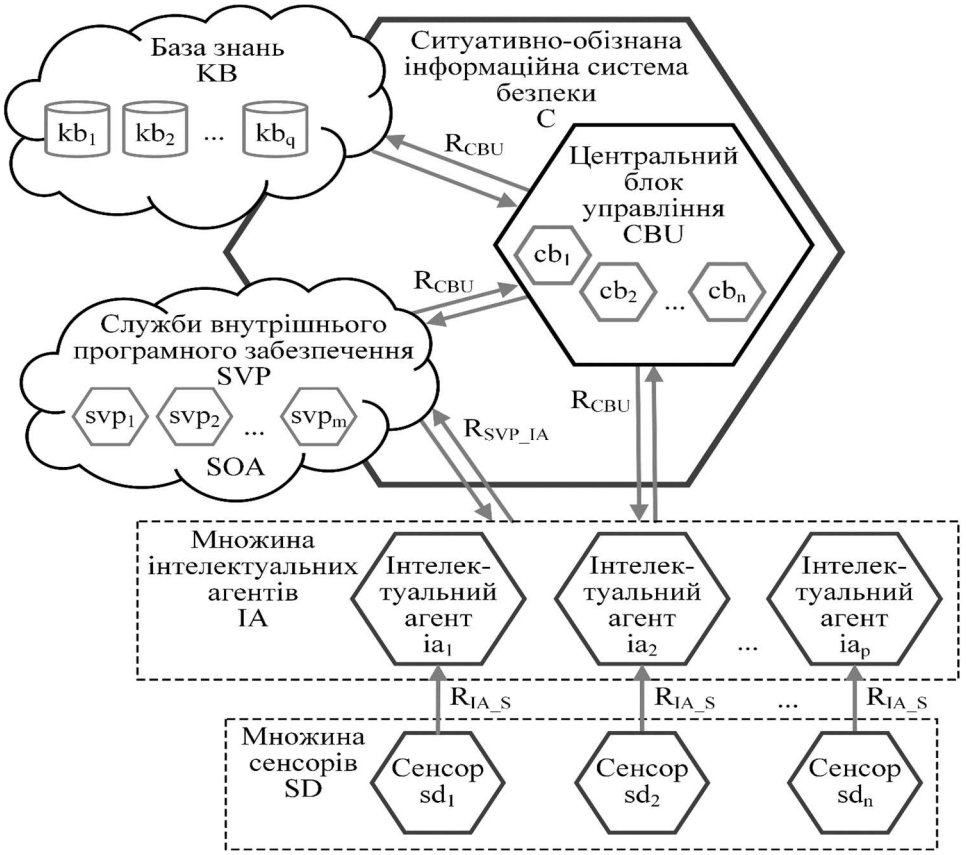


Рис. 4.5 Структура інтелектуальної програмної системи безпеки

На основі аналізу, використовуючи правила W, центральний блок приймає рішення щодо подальших дій. Він формує команди Cmd і надсилає їх до відповідних компонентів або підсистем.

Структуру інтелектуальної ситуаційно-обізнаної програмної системи безпеки подано на (Рис.4.5):

$$C = (IA, SVP, CBU, SD, KB) \quad (4.4)$$

де C – структура ситуаційно-обізнаної інтелектуальної програмної системи безпеки, IA – інтелектуальні агенти, множина автономних пристроїв, що орієнтовані на виконання певних завдань і використовують дані, отримані з сенсорів IA={ia_1,ia_2,…,ia_p}, SVP – служби внутрішнього програмного забезпечення, реалізовані як хмарні сервіси, що виконують ресурсомісткі обчислення відповідно до вимог сервіс-орієнтованої архітектури SVP={svp_1,svp_2,…,svp_m}, CBU – центральний блок управління, що відповідає за формування загальної картини ситуації та аналіз інформації, отриманої від сенсорів, агентів і служб CBU={cb_1,cb_2,…,cb_n}, SD – множина сенсорів, що інтегровані з інтелектуальними агентами та подають дані щодо біжучої ситуацію SD={sd_1,sd_2,…,sd_n}, KB – база знань, яка використовується центральним блоком управління як експертні знання для аналізу ситуації та ухвалення відповідних рішень KB={kb_1,kb_2,…,kb_q}.

Відношення між компонентами інтелектуальної програмної системи безпеки подано таким чином**:** зв’язки між сенсорами та агентами (R_{IA_SD}) забезпечують відбирання даних з сенсорів інтелектуальними агентами:

$$R_{IA\_SD} \subseteq IAxSD \quad (4.5)$$

де IA – інтелектуальний агент, SD – множина сенсорів.

Взаємодія між агентами та множиною сенсорів (R_{IA_SD}) визначається як набір пар, у яких кожен агент (IA) передає дані сенсора (SD) службі для виконання обчислень. Взаємодія між службами та агентами (R_{SVP_IA}) визначається перш за все виконанням службою внутрішнього програмного забезпечення обчислень на основі даних, сформованих відповідними агентами:

$$R_{SVP\_IA} \subseteq SVPxIA \quad (4.6)$$

де SVP – служба внутрішнього програмного забезпечення, IA – інтелектуальний агент.

Взаємодія між службами внутрішнього програмного забезпечення та інтелектуальними агентами (R_{SVP_IA}) визначається як відношення, у якому служба внутрішнього програмного забезпечення (SVP) отримує дані від інтелектуального агента (IA) і виконує відповідні обчислення. Взаємодія між центральним блоком та іншими компонентами (R_{CBU}) передбачає, що Центральний блок управління взаємодіє з інтелектуальними агентами, службами внутрішнього ПЗ та базою знань для реалізації процесів аналізу ситуації:

$$R_{CBU} \subseteq CBUx(IA \cup SVP \cup KB) \quad (4.7)$$

де CBU – центральний блок управління, що забезпечує виконання процедур аналізу даних та прийняття рішень, IA – інтелектуальні агенти, які збирають та передають дані, SVP – служби внутрішнього програмного забезпечення, що виконують обчислення, KB – база знань, яка забезпечує зберігання та доступ до інформації, необхідної для аналізу ситуації. Інтелектуальні агенти (IA) збирають дані з сенсорів (SD) та передають їх службам внутрішнього програмного забезпечення (SVP) для подальшого опрацювання. Служби внутрішнього програмного забезпечення (SVP), реалізовані як хмарні сервіси, виконують ресурсомісткі обчислення, а результати передаються до центрального блоку управління (CBU). Центральний блок управління (CBU) використовує інформацію від агентів, служб, а також знання з бази знань (KB) для формування загальної картини ситуації безпеки та прийняття рішень. Ця модель описує структуру компонентів інтелектуальної програмної системи безпеки, їхні зв’язки, що забезпечують ефективне управління безпековими ситуаціями.

4.3 Архітектура інтелектуальних програмних систем безпеки з ситуаційною обізнаністю

На відміну від індивідуального «розумного» будинку, призначеного для проживання в основному окремих сімей, великі житлові комплекси, зорієнтовані на розташування громад та чисельних різнопланових груп мешканців, що в свою чергу генерує велику кількість специфічних та доволі складних для вирішення

проблем. Це вимагає інсталяції та обслуговування суттєво складніших та надійніших IT інфраструктур. Під архітектурою інтелектуальної програмної системи безпеки розуміємо її концептуальну модель, в якій визначаються компоненти та їх взаємодії, функції, а також методи та засоби їх інтеграції із зовнішнім середовищем. В архітектурі фіксується, як саме компоненти системи взаємодіють між собою, як вони інтегруються з іншими системами і як забезпечується реалізації функцій зручності використання, масштабованості, безпеки та ефективності роботи цілісної інформаційної системи (Рис.4.6).

Рис. 4.6 Функціональні особливості інтелектуальної програмної системи безпеки житлового комплексу

Архітектура інтелектуальної програмної системи безпеки житлового комплексу містить ряд компонентів (Рис.4.7), які забезпечують виконання таких вимог як масштабованість, наявність декількох точок доступу, охоплення відеоспостереженням спільних просторів, опрацювання великих обсягів даних і управління ними, екстерне реагування на проблемні ситуації.

Передбачається, що система може опрацьовувати зростаючий трафік даних, збільшувати кількість одночасних підключень. На відміну від окремого будинку з обмеженими точками входу, житловий комплекс має численні точки доступу, включаючи ворота, вестибюлі, гаражі та окремі блоки та ін. Ефективні

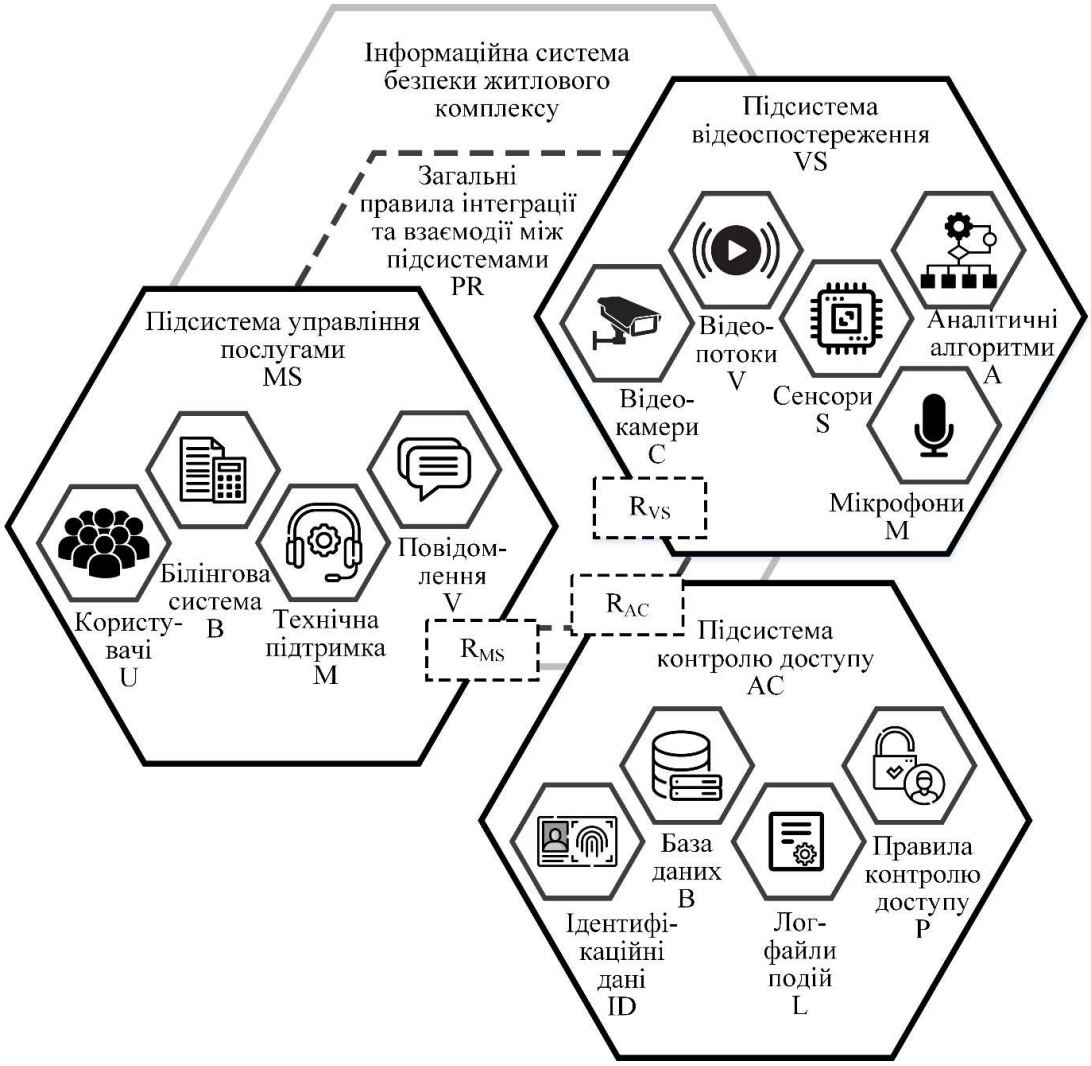


Рис. 4.7 Архітектура інтелектуальної програмної системи безпеки з ситуаційною обізнаністю

безпекові рішення повинні передбачати системно узгоджене керування цими точками та їх контроль. Спільні у використанні приміщення, такі як спортзали, басейни та паркінги, потребують додаткових заходів безпеки. Це включає спостереження та контроль доступу, з метою запобігання несанкціонованому проникненню та забезпеченню безпеки мешканців у зонах загального користування. Керувати даними з численних пристроїв і сенсорів у великій громаді суттєво складніше, аніж у виокремленому будинку, призначеному для проживання однієї сім'ї. Координація реагування на надзвичайні ситуації у великій громаді є суттєво складнішою, що вимагає використання надійних систем зв'язку та дотримання протоколів для забезпечення швидких та ефективних дій у разі виникнення інцидентів. Окрім того, інформаційні системи

безпеки у житлових комплексах повинні бути інтегровані з системами місцевих правоохоронних органів та служб екстреної допомоги.

Архітектуру інтелектуальної програмної системи безпеки житлового комплексу формально подано кортежем множин, що задають окремі групи елементів в підсистемах і множин відношень, що задані на них:

$$IS = (VS, AC, MS, PR) \tag{4.8}$$

де VS (Video Surveillance) – підсистема відеоспостереження

$$VS = (C, M, S, A, V, R_{VS}) \tag{4.9}$$

де C – відеокамери, M – мікрофони, S – сенсори, A – аналітичні алгоритми (виявлення руху, розпізнавання обличь), V – відеопотоки, R_{VS} – правила взаємодії із центральним блоком.

Підсистема відеоспостереження (VS) забезпечує збір даних з допомогою камер, опрацьовує їх з використанням аналітичних алгоритмів, і забезпечує взаємодію з іншими підсистемами через центральний блок.

AC (Access Control) – підсистема контролю доступу:

$$AC = (ID, B, L, P, R_{AC}) \tag{4.10}$$

де ID – ідентифікаційні дані (картки, біометрія), B – база даних, L – лог файли, P – правила контролю доступу, R_{AC} – реакції на події (блокування)..

Підсистема контролю доступу (AC) забезпечує регулювання доступу до приміщень, зберігання журналів подій та можливість автоматичного блокування доступу у разі загрози.

MS (Management Services) – підсистема управління послугами:

$$MS = (U, B, T, N, R_{MS}) \tag{4.11}$$

де U – користувачі, B – білінгова система, T – технічна підтримка, N – повідомлення, R_{MS} – правила керування послугами.

Підсистема управління послугами оператора (MS) реалізує процеси управління користувачами, їхніми передплатами, обслуговуванням і надає інформаційні сервіси, зокрема забезпечує користувачів оперативними повідомленнями.

PR – загальні правила інтеграції та взаємодії між підсистемами:

$$PR = (PR_{VS}, PR_{AC}, PR_{MS}) \quad (4.12)$$

де RP_{VS}, RP_{AC},RP_{MS} – правила інтеграції та взаємодії кожної із підсистем.

Ці правила визначають, як підсистеми взаємодіють одна з одною та з Центральним блоком управління. Формалізм комплексно подає основні компоненти архітектури та визначає, їх взаємодію з метою належної реалізації та підтримки функцій безпеки в житловому комплексі.

4.4 Функціональні особливості підсистем інтелектуальної програмної системи безпеки

Архітектура інтелектуальної програмної системи безпеки житлового комплексу включає три ключові підсистеми, а саме відеоспостереження, контролю доступу та управління послугами оператора.

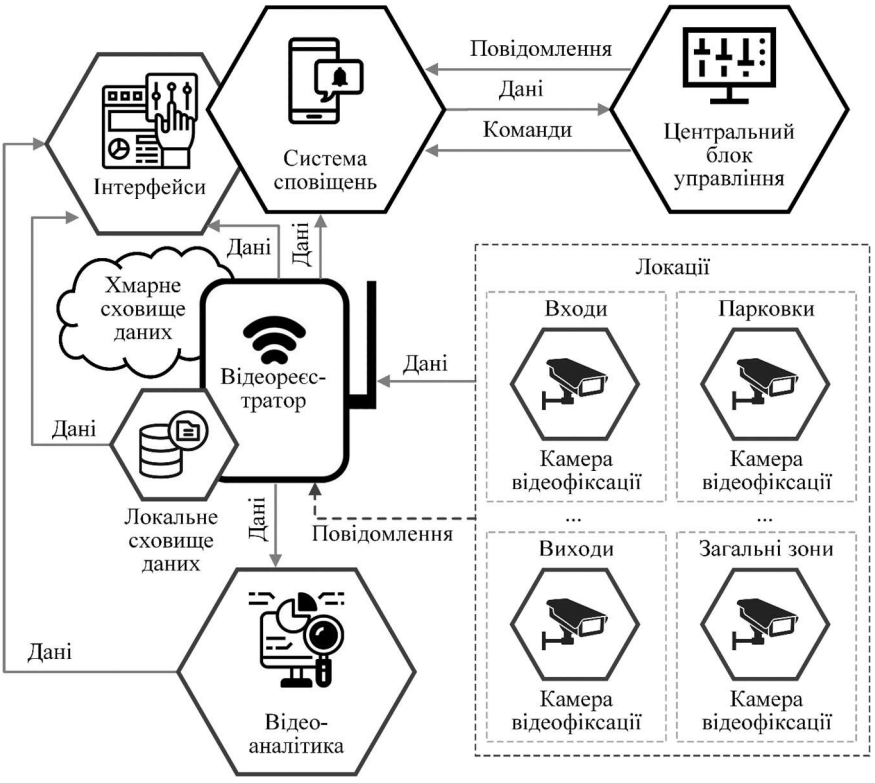


Рис. 4.8 Архітектура підсистеми відеоспостереження

Підсистема відеоспостереження технічно базується на камерах фіксації відеопотоків в реальному масштабі часу, дані яких зберігаються на локальному відеореєстраторі або в хмарному сховищі. Підсистема контролю доступу

включає точки контролю доступу, які реєструють вхід і вихід, відомості про доступ, які передаються в базу даних для подальшого їх аналізу та загального моніторингу.

Підсистема управління послугами оператора забезпечує реалізацію функції інтеграції з сервісами постачальників (інтернет, охорона), управління з допомогою користувацьких інтерфейсів (мобільні або веб-застосунки та ін.). Подамо узагальнений опис основних елементів архітектури підсистеми відеоспостереження (Рис.4.8) та процесів їх функціонування і взаємодії .

Камера відеофіксації відповідає за захоплення відео з різних локацій (входи, виходи, парковки, загальні зони), підключається до відеореєстратора та інтегрується з процедурами відеоаналітики. Відеореєстратор зберігає зображення локально або в хмарі впродовж певного часового періоду та інтегрується з каналом передачі повідомлень від відеокамери.

Система відеоаналітики опрацьовує потокове відео для виявлення руху, розпізнавання обличь та виявлення підозрілих дій та підключена до відеореєстратора для забезпечення доступу до відео матеріалів. Система сповіщень надсилає повідомлення до центральної системи управління або користувача, базуючись на результатах аналітичних процедур та підключена до системи відеоаналітики для отримання результатів аналізу. Компонента «Інтерфейс» забезпечує користувачам доступ до потокового відео та записаних відеоматеріалів у реальному масштабі часу, вона підключена як до відеореєстратора, так і до функції відеоаналітики.

Центральний блок управління інтегрується з іншими підсистемами, такими як система контролю та управління доступом, сигналізація та ін. для швидкого реагування на інциденти. Камери захоплюють відеопотік та передають його на відеореєстратор. Система відеоаналітики отримує відео для опрацювання і виявляє підозрілі дії, а також передає результат у систему сповіщень, яка надсилає повідомлення. Користувачі мають змогу переглядати відео в реальному масштабі часу або попередньо записані та збережені матеріали.

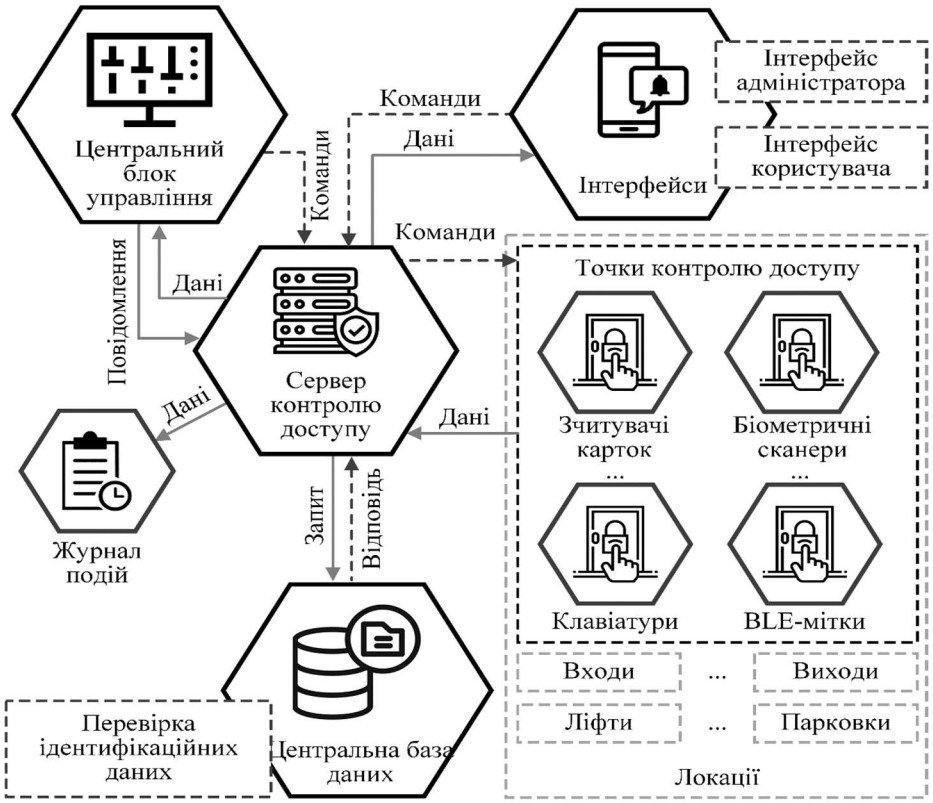


Рис. 4.9 Архітектура підсистеми контролю доступу

Основні елементи архітектури підсистеми контролю доступу наведені на рис.4.9. Точки контролю доступу – це фізичні або електронні пристрої, зокрема, зчитувачі карток, біометричні сканери, клавіатури для введення особистих ідентифікаційних номерів-кодів (PIN-код), мітки Bluetooth з низьким енергоспоживанням (BLE- мітки) тощо. Водночас ці пристрої відповідають за ідентифікацію користувачів перед наданням доступу до конкретної зони (вхід до будівлі, ліфт, паркінг тощо). На сервері контролю доступу розміщується центральна база даних, в якій зберігається інформація про користувачів, їхні ідентифікатори, права доступу та записи про входи/виходи. База даних отримує запити, що надходять з точок контролю доступу, перевіряє ідентифікаційні дані користувача (картка, біометрія, PIN-код) та надає або відхиляє доступ.

З точки контролю доступу передаються дані про ідентифікацію користувачів до сервера контролю доступу, де вони перевіряються. Якщо ідентифікація успішна, сервер надає доступ, а інформація про подію записується в журнал подій, в якому зберігаються записи про всі входи та виходи, включаючи

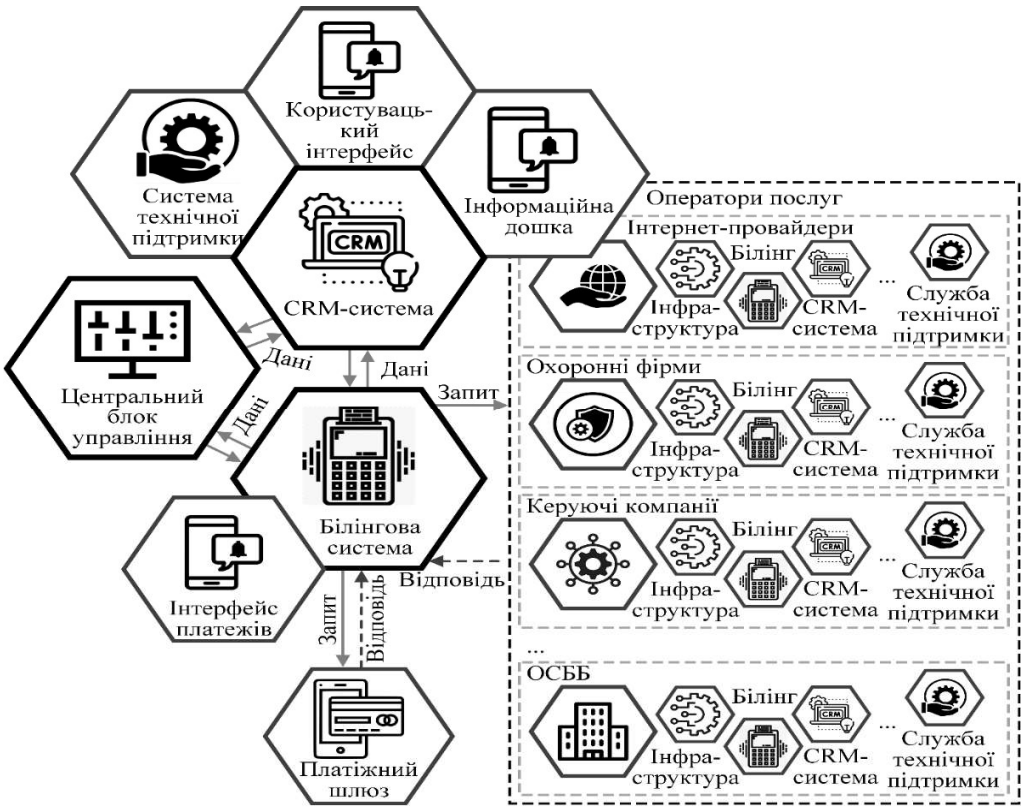


Рис. 4.10 Архітектура підсистеми управління послугами

час, місце та ідентифікацію користувачів. Інтерфейс адміністратора призначений для керування доступом користувачів, налаштування правил доступу, перегляду журналів подій та ін. і він взаємодіє із сервером контролю доступу для управління налаштуваннями та реалізації функції моніторингу. Інтерфейс користувача забезпечує управління доступом через мобільний застосунок або веб-інтерфейс та підключений до сервера контролю доступу для надання користувачам відповідних функцій.

Підсистема управління послугами забезпечує мешканцям взаємодію з сервісами житлового комплексу, такими як інтернет-послуги, охорона, обслуговування житлового комплексу, оплата послуг та інші (Рис.4.10).

Проаналізовано основні елементи архітектури підсистеми управління послугами. Оператори послуг – компанії та організації, які надають послуги мешканцям: інтернет-провайдери, охоронні фірми, компанії з обслуговування домофонів, керуючі компанії, об’єднання співвласників багатоквартирного будинку (ОСББ) і ін. Кожен оператор має свою власну інфраструктуру,

наприклад, білінгові системи, програмне забезпечення, необхідне для систем управління взаємовідносинами з клієнтами (CRM-систем), служби технічної підтримки.

Користувацький інтерфейс включає мобільні застосунки або веб-інтерфейси, з допомогою яких користувачі можуть управляти послугами, оплачувати рахунки, отримувати підтримку, переглядати інформацію про стан надання послуг та отримувати повідомлення від операторів. Зазвичай він інтегрований з білінговою системою, CRM-системою та платіжними шлюзами для надання комплексного сервісу мешканцям.

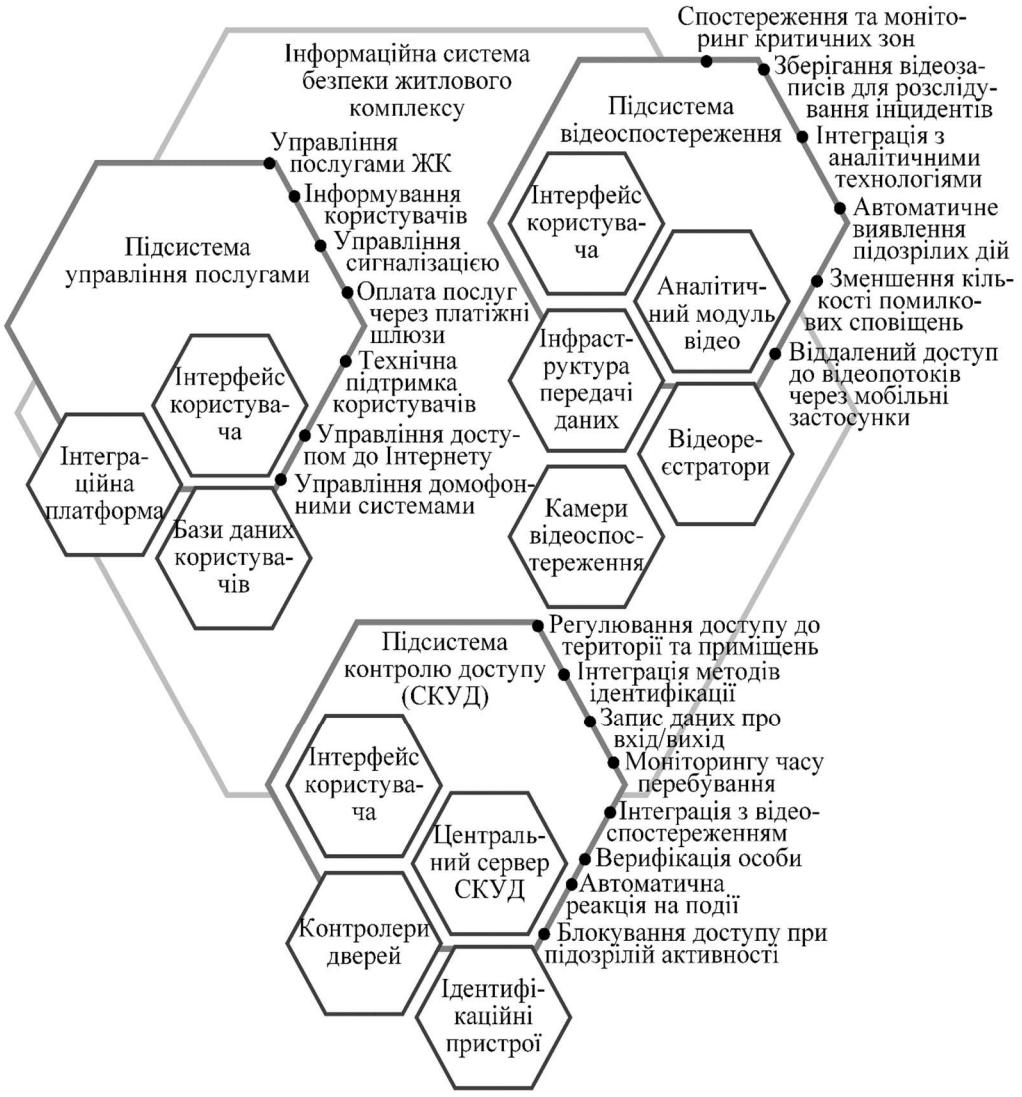


Рис. 4.11 Інтегроване подання компонентів інтелектуальної системи безпеки житлового комплексу

Система технічної підтримки забезпечує підтримку користувачів при вирішенні технічних проблем. Мешканці можуть подавати заявки на технічне обслуговування використовуючи мобільні застосунки або веб-інтерфейс. Підсистеми відеоспостереження, контролю доступу та управління послугами можуть працюють в єдиній IT екосистемі, використовуючи центральний контрольний блок, як диригента. На рис. 4.11 зображене інтегроване подання компонентів системи безпеки.

Підсистеми можуть бути інтегровані в єдину систему сповіщень про інциденти, виявлені підсистемою відеоспостереження, можуть автоматично активувати додаткові функції в підсистемі контролю доступу або викликати служби технічного обслуговування, задіяючи при цьому підсистему управління послугами.

Вхідні дані інтелектуальної системи безпеки можна класифікувати таким чином: дані від IoT-пристроїв (відеопотоки з камер, дані від сенсорів руху, температури, звуку, сигнали від пристроїв контролю доступу (RFID-картки, біометричні пристрої), інформація про стан інфраструктури (наприклад, дверей, вікон)), час реакції системи, пропускна здатність, типи подій (виявлення руху, надання доступу до контрольованих зон, активація сигналу тривоги, зміна статусу пристроїв (ввімкнення/вимкнення).

Успішність виконання дослідження обумовлена розробленням інтелектуальної системи безпеки, що ґрунтується на ситуаційній обізнаності, з використанням інноваційних підходів. В інтелектуальній системі безпеки досягаються високі показники гнучкості, адаптивності, пропускної здатності та інтегрованості підсистем у єдину розлогу IoT-мережу. Вона має ряд суттєвих переваг. Можливість поєднання та гнучкого розподілу міркувань між інтелектуальними агентами та серверними службами сприяє її адаптації до різнопланових обчислювальних потреб при виконанні широкого спектру завдань. Реалізацію концепції ситуаційної обізнаності як на локальному, так і загальносистемному рівнях забезпечує підтримку функцій аналізу ситуації та планування, що дає змогу системі бути проактивною та передбачати розвиток

ситуації та підтримку безперервного навчання на основі відгуків про локалізацію та ефективну реакцію на реальні ситуації.

Висновки до розділу 4

Розроблено структуру інтелектуальної системи безпеки, яка об'єднує ключові компоненти в єдину адаптивну мережу, а її основою є IoT-мережа, яка забезпечує інтеграцію сенсорів і інших пристроїв із центральним блоком управління. Інтелектуальні агенти на основі бази знань реалізують процеси аналізу ситуації, автономність прийняття рішень. Вони дозволяють виконувати локальне опрацювання даних і автономно взаємодіяти з іншими вузлами системи. Запропонована структура враховує можливість динамічного масштабування та оновлення компонентів IoT-мережі. Унікальність розробленої структури полягає в такій інтеграції інтелектуальних агентів із центральним блоком управління, яка забезпечує не лише централізоване, але й децентралізоване управління процесами. Система працює з використанням ситуаційно-орієнтованих концептуальних моделей конкретних ситуацій, що отримані з бази знань з використанням фактору подібності контекстів. Завдяки такому підходу, інтелектуальна системи з ситуаційною обізнаністю здатна моделювати та аналізувати стан середовища у реальному масштабі часу, виявляти ситуації, приймати рішення та діяти відповідно до них. В системі підтримується безперервний процес навчання узгодження прогнозованих даних із даними сенсорів.

Інтелектуальні агенти дозволяють ефективно знижувати навантаження на центральний блок управління, виконуючи первинний аналіз даних та прийняття рішень у режимі реального часу. Ці елементи структури надають можливість поєднувати виконання різних типів завдань в інформаційній системі безпеки з урахуванням ситуації та динамічно розподіляти обчислювальне навантаження між агентами та сервісами. У розробленій інформаційній системі поєднано переваги централізованого й децентралізованого підходів. Це пояснюється потребою зменшити затримки у прийнятті рішень та підвищити стійкість системи до відмов окремих її компонентів.

Запропонована архітектура поєднує підсистеми відеоспостереження, контролю доступу та управління послугами оператора з допомогою єдиного центру опрацювання даних, в ній враховано сумісність підсистем та їхню інтеграцію в IoT-мережу. Така інтеграція підсистем дозволяє повно та системно відображати цілісну картину безпекової ситуації оператору.

Проведено класифікацію елементів підсистем та проаналізовано наявні відношення між ними. Використання аналітичних алгоритмів та хмарних сервісів дозволяє забезпечувати оперативний контроль, моніторинг, аналіз загроз та оперативне реагування на потенційні інциденти.

Розділ 5. Реалізація інтелектуальної системи безпеки «АСТРА. Безпечний ЖК»

5.1 Концептуальне моделювання інтелектуальної системи безпеки

Інтелектуальна система безпеки житлового комплексу — це складна система, що включає багато складних, нових за сутністю, і важливих функцій. Відповідно до закону Галла [121] така система не може бути цілісно побудованою з нуля, а повинна стати результатом еволюції більш простих, практично життєздатних системних реалізацій. Цей підхід підтримується гнучкою методологією розроблення програмного забезпечення [122] та ідеєю мінімально життєздатного продукту, згідно з якою версія продукту повинна містити достатньо функцій, щоб її могли використовувати перші клієнти [123]. Як зазначалося у четвертому розділі перша черга інформаційної системи безпеки житлового кварталу складається із трьох підсистем: відеоспостереження, контролю доступу, управління послугами оператора. Розглянемо процес концептуального моделювання інформаційної системи безпеки житлового комплексу на прикладі підсистеми відеоспостереження, яка реалізує функцію відеоспостереження у всіх критичних зонах житлового комплексу: входах, виходах, ліфтах, паркінгах, прилеглих територіях та громадських зонах. Використання високоякісних відеокамер дозволяє отримувати чіткі зображення, включаючи умови недостатнього освітлення завдяки ІЧ-підсвічуванню. Відеокамери фіксують відео, яке зберігається на локальному відеореєстраторі або хмарному сховищі протягом визначеного періоду. Це забезпечує перегляд записів для розслідування інцидентів або надання доказів у разі необхідності. В підсистемі відеоспостереження імплементовані технології відеоаналітики, такі як виявлення рухомих об’єктів, розпізнавання обличь, автоматичне виявлення підозрілих дій або об’єктів. Така інтеграція сприяє оперативній передачі інформації про інциденти до центрального блоку для прийняття відповідних заходів. Підсистема дозволяє користувачам отримувати доступ до відеопотоків

у режимі реального часу через мобільні застосунки або веб-інтерфейси. Камери відеоспостереження, встановлені на відкритих територіях, зазвичай мають високий ступінь захисту від впливу складних погодних умов та вандалізму (сертифікація IP66, IK10), що забезпечує надійну роботу впродовж тривалого часу. Підсистема відеоспостереження є важливим елементом комплексної безпекової системи житлового комплексу з високим рівнем захисту завдяки поєднанню функцій постійного моніторингу, інтелектуальної аналітики та можливостей інтеграції з іншими компонентами системи безпеки.

Підсистема відеоспостереження є ключовим компонентом системи безпеки житлового комплексу, забезпечуючи моніторинг, аналіз і реагування на загрози в режимі реального часу. Вона складається з ядра та додаткових компонентів, які взаємодіють із ядром через API. Основні компоненти включають сервер відеоспостереження, білінг, СКУД панелей і шлагбаумів, агрегатор смс і застосунок. Сервер містить API для роботи з компонентами системи та записами, live API, компонент опрацювання відео, дві бази даних (для записів і live). Білінг забезпечує взаємодію з базою даних через API. Агрегатор смс використовується для надсилання повідомлень. Застосунок слугує інструментом взаємодії кінцевого користувача із підсистемою відеоспостереження та іншими підсистемами. Інтеграція відеопотоків із сенсорами руху, звуку та контролю доступу дозволяє автоматично ідентифікувати підозрілу активність і передавати тривожні сигнали. Взаємодія з іншими підсистемами забезпечує блокування входів або виклик охорони, а функція контролю реалізується через мобільні та веб-інтерфейси. Концептуальне моделювання за допомогою UML-діаграм сприяє чіткому визначенню архітектури, оптимізації процесів, масштабованості, підвищенню безпеки та ефективному впровадженню комплексної системи безпеки житлового комплексу [124].

Діаграма послідовності відображає взаємодію між об’єктами підсистеми, зосереджуючись на їхній хронологічній послідовності. Вона моделює функціонування підсистеми відеоспостереження у сценаріях, таких як виявлення руху або запит на перегляд відеозапису. У сценарії "Виявлення руху та запис

відео з камери" визначено таких акторів та об’єкти: MotionDetector, який виявляє рух у зоні спостереження; Camera, що починає запис відео; VideoArchive, який зберігає відеозапис; SystemController, який координує взаємодію між компонентами системи; SecurityPersonnel, який отримує сповіщення про появу рухомих об’єктів.

Послідовність подій у цьому сценарії починається з того, що MotionDetector виявляє появу рухомих об’єктів і надсилає сигнал до SystemController. SystemController передає команду відповідній камері (Camera) для початку запису. Camera передає відеопотік до VideoArchive для збереження. Після цього SystemController надсилає сповіщення SecurityPersonnel з деталями події, такими як місце, час та камера. SecurityPersonnel має можливість переглянути відеозапис із VideoArchive. Елементи UML діаграми послідовності включають лінії життєвого циклу для об’єктів MotionDetector, SystemController, Camera, VideoArchive і SecurityPersonnel. Повідомлення включають: detectMotion() - викликається MotionDetector; alertController() - повідомлення від MotionDetector до SystemController; startRecording() - команда від SystemController до Camera; saveVideo() - збереження відео в архів; notifySecurity() - сповіщення охоронця про подію; retrieveVideo() - запит охоронця до VideoArchive для перегляду запису. Діаграма ілюструє динамічну взаємодію між об’єктами в часі, допомагаючи повніше сприймати потоки даних і повідомлень.

Проаналізуємо сценарій використання підсистеми відеоспостереження, візуалізований діаграмою послідовності. Користувач встановлює застосунок на смартфон, відкриває його та вводить номер телефону для авторизації. Генерується код підтвердження авторизації і через агрегатор смс надсилає користувачу. Користувач вводить код підтвердження і активує опцію “Підтвердити”. Далі відбувається звіряння коду та у випадку відсутності розбіжностей видається повідомлення про успішну авторизацію. Після цього користувач у застосунку надсилає запит для встановлення наявності зарезервованого акаунту, що призначений для здійснення відео/аудіо дзвінків.

Спочатку пошук акаунту відбувається в базі даних. Якщо відповідний акаунт знайдено, то відправляється позитивна відповідь користувачу у застосунок. Якщо акаунт не знайдено, але послуга має бути доступною, то резервується акаунт на

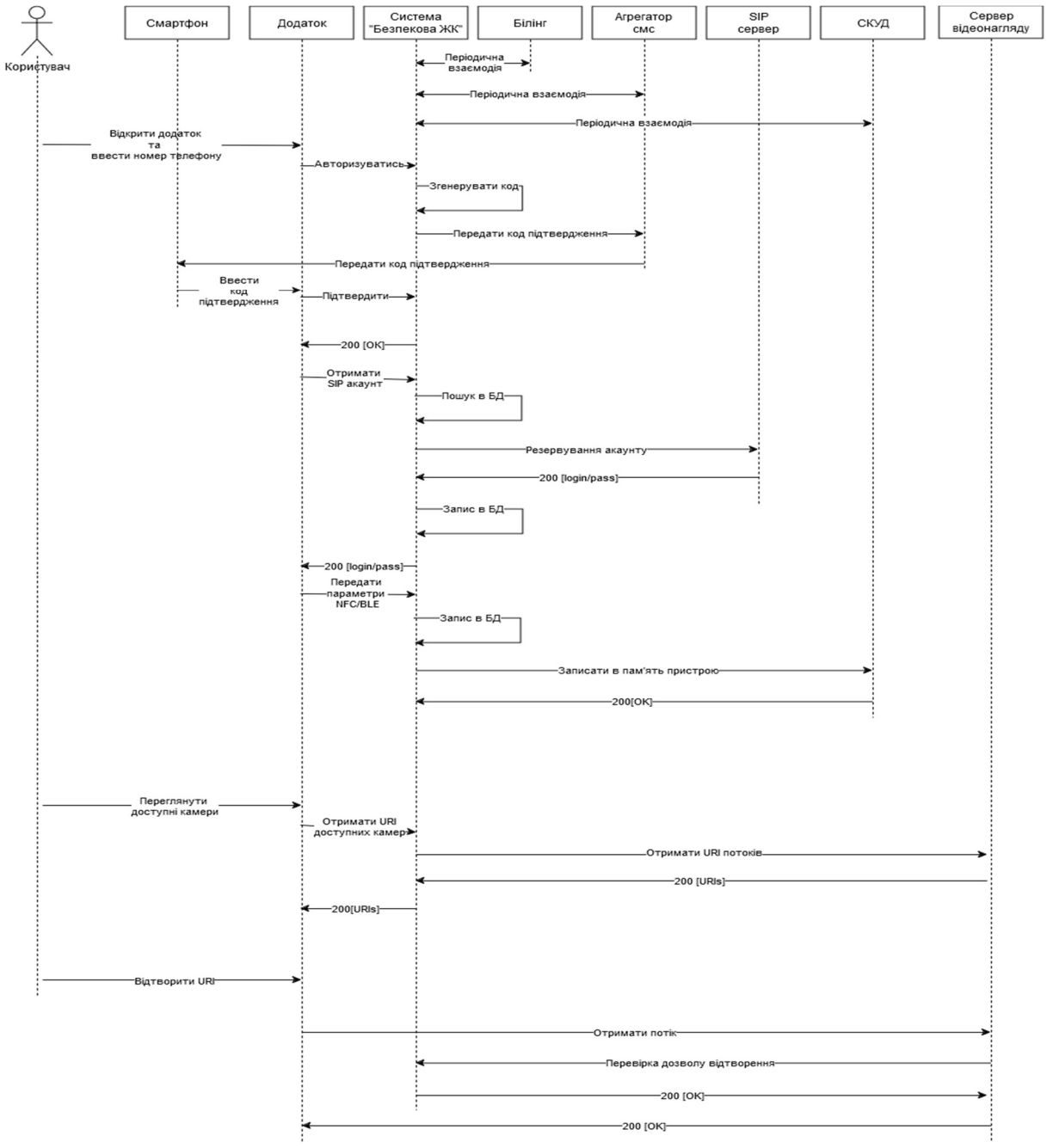


Рис.5.1 UML діаграма послідовності підсистеми відеоспостерження

SIP сервері, записуються дані аканту в базу даних і надсилається повідомлення користувачу у застосунок, який без участі користувача авторизує акаунт для прийому/здійснення дзвінків. Після завершення успішної реєстрації акаунту,

застосунок передає мітку NFC/BLE. Ця мітка записується в базу даних, а також надсилається на пристрої контролю та управління доступом (СКУД).

При потребі користувач може переглядати відомості з відеокамер. При переході у застосунку на вкладку «відеоспостереження», надається можливість відправлення запиту для отримання дозволу на перегляд відеопотоку з певної камери, у відповідь на який отримується підтвердження можливості перегляду відеопотоку.

Діаграма розгортання ілюструє фізичну архітектуру підсистеми відеоспостереження, включаючи розташування апаратних вузлів, програмного забезпечення та їх взаємозв'язки, моделює фізичне розташування компонентів - сенсорів, серверів для опрацювання та аналізу даних, інтерфейсів для доступу користувачів.

Серед апаратних вузлів виділяються камери спостереження, які захоплюють відеопотоки та передають дані до центрального сервера. Камери встановлюються на території житлового комплексу та оснащені вбудованим програмним забезпеченням для опрацювання відео та мережевого підключення. Центральний сервер забезпечує координацію роботи підсистеми, зберігання архівів і управління доступом. Програмне забезпечення сервера включає компоненти управління відеоспостереженням та базу даних для архівування відео і логів подій. Сенсори руху встановлюються у ключових зонах спостереження та забезпечують виявлення рухомих об'єктів та передачі сигналу до камери або сервера. Мобільні пристрої користувачів, на яких встановлені мобільним застосунки, слугують для доступу до відеоархіву, перегляду потоків у реальному масштабі часу та отримання сповіщень. Мережеве обладнання, включаючи маршрутизатори, комутатори та мережеві камери, забезпечує передачу даних між вузлами.

Для забезпечення комунікаційних зв'язків у підсистемі відеоспостереження передбачається підключення камер до центрального сервера через локальну мережу, передачу сповіщень мобільним пристроям через захищені інтернет-з'єднання і підключення сенсорів руху до камер або сервера

через дротові чи бездротові зв’язки. Програмні компоненти включають елементи для керування подіями, які опрацьовують сигнали від сенсорів руху; модуль запису та зберігання відео, що дозволяє формувати базу даних з відеофайлів у визначеному форматі; та модуль сповіщень, який надсилає сигнал тривоги охоронцям або користувачам. Сценарій розгортання описує процеси роботи підсистеми відеоспостереження: камери спостереження передають відеопотік до центрального сервера; сервер опрацьовує сигнали від сенсорів руху, ініціює запис і зберігає відео в архіві; сповіщення про події надходять на мобільні пристрої охоронців чи мешканців; користувачі отримують доступ до архіву чи переглядають відеопотоки в реальному масштабі часу через мобільний застосунок.

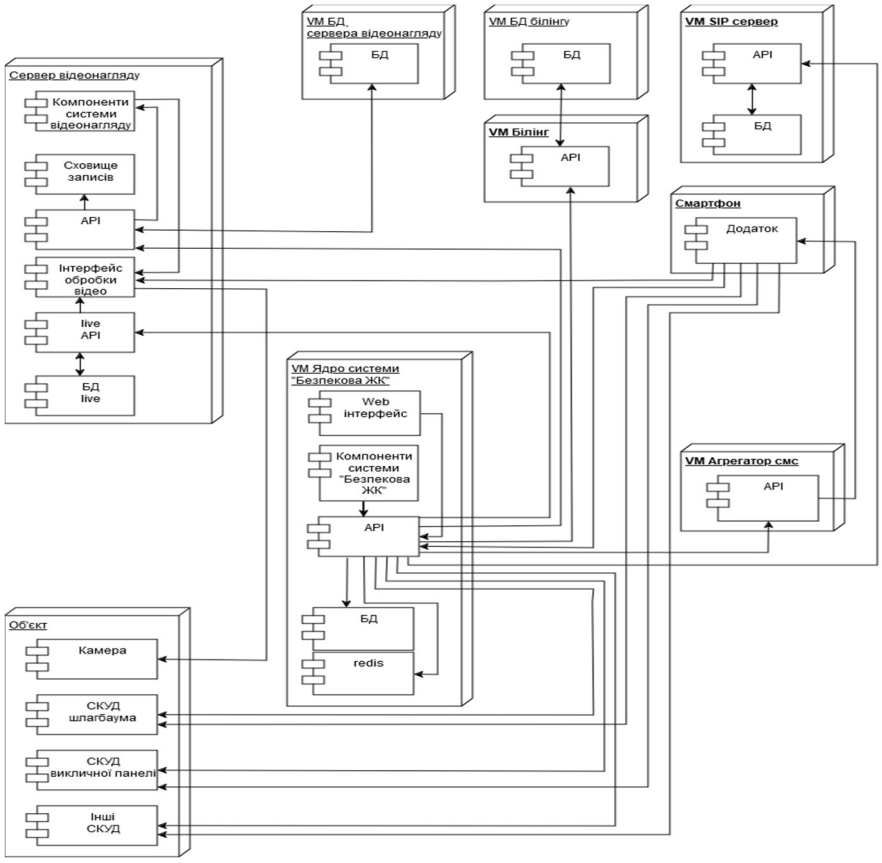


Рис. 5.2 UML діаграма розгортання

Графічно діаграма розгортання містить прямокутні вузли, що представляють фізичне обладнання (камери, сервери, пристрої), компоненти

програмного забезпечення, які працюють на цих вузлах, та лінії зв'язку, що відображають мережеві підключення. Вона ілюструє фізичне розміщення компонентів системи на вузлах і є важливою для проектування апаратної частини системи відеоспостереження. Діаграма розгортання відображає модулі, які підтримують ситуаційну обізнаність та показує їх взаємозв'язок із сенсорами, камерами, компонентами сповіщень і користувачами.

Діаграма компонентів підсистеми відеоспостереження демонструє взаємозв'язки елементів, інтеграцію інтелектуальних пристроїв, програмних компонентів, процедури аналізу даних. Основними компонентами такої підсистеми є IoT-пристрої, які включають давачі та сенсори для збору інформації про навколишнє середовище - давачі руху, температури, вологості або звуку; розумні пристрої, такі як камери відеоспостереження, дверні замки та освітлювальні системи, якими можна централізовано керувати; актори, які виконують дії у відповідь на команди, наприклад, відкривання дверей або включення сигналізації. Діаграма демонструє, як підсистема відеоспостереження інтегрується з іншими підсистемами для забезпечення ситуаційної обізнаності та підвищення рівня безпеки на основі технології IoT.

Діаграма компонентів демонструє структуру програмних і апаратних компонентів підсистеми відеоспостереження, їхні взаємозв'язки та функціональність. Основними складовими є компоненти керування камерами, які відповідають за контроль їх роботи. Вони виконують функцію запуску та зупинки запису, визначення статусу камер і їх взаємодіє з фізичними камерами через API або драйвери.

Відеозаписи зберігаються у архіві, забезпечується їх пошук та автоматичне видалення через визначений проміжок часу. Детектор руху виявляє рухомі об'єкти у зоні спостереження, проводить перевірку активності, в разі виявлення небезпечної ситуації надсилає сигнал тривоги і інтегрується з камерами для активації запису. Система сповіщень інформує користувачів про події, надсилаючи повідомлення та налаштовуючи надсилання сигналу тривоги. Інтерфейс користувача забезпечує доступ до функціоналу підсистеми через веб-

інтерфейс або мобільний застосунок, включаючи перегляд відеопотоку в реальному масштабі часу, доступ до архіву відеозаписів та зміну налаштувань. База даних зберігає інформацію про камери, архіви, події та користувачів, забезпечуючи виконання запитів та збереження даних. Взаємозв'язки між компонентами подають взаємодії камер із модулем детектора руху для активації запису під час виявлення рухомих об'єктів. Процедура керування камерами передбачає передачу відеозаписів до архіву відеоданих. Користувач через користувацький інтерфейс звертається до архіву відеоданих для відображення інформації, а компонента сповіщень надсилає повідомлення користувачам через налаштовані канали. Сценарій роботи системи передбачає, що при виявленні активності детектор руху сповіщає компоненту керування камерами, про необхідність вмикання камери, після чого камера починає запис і передає відеодані до архіву. Компонента сповіщень інформує користувача про подію, а користувач має можливість переглядати записи або відеопотоки у реальному масштабі часу.

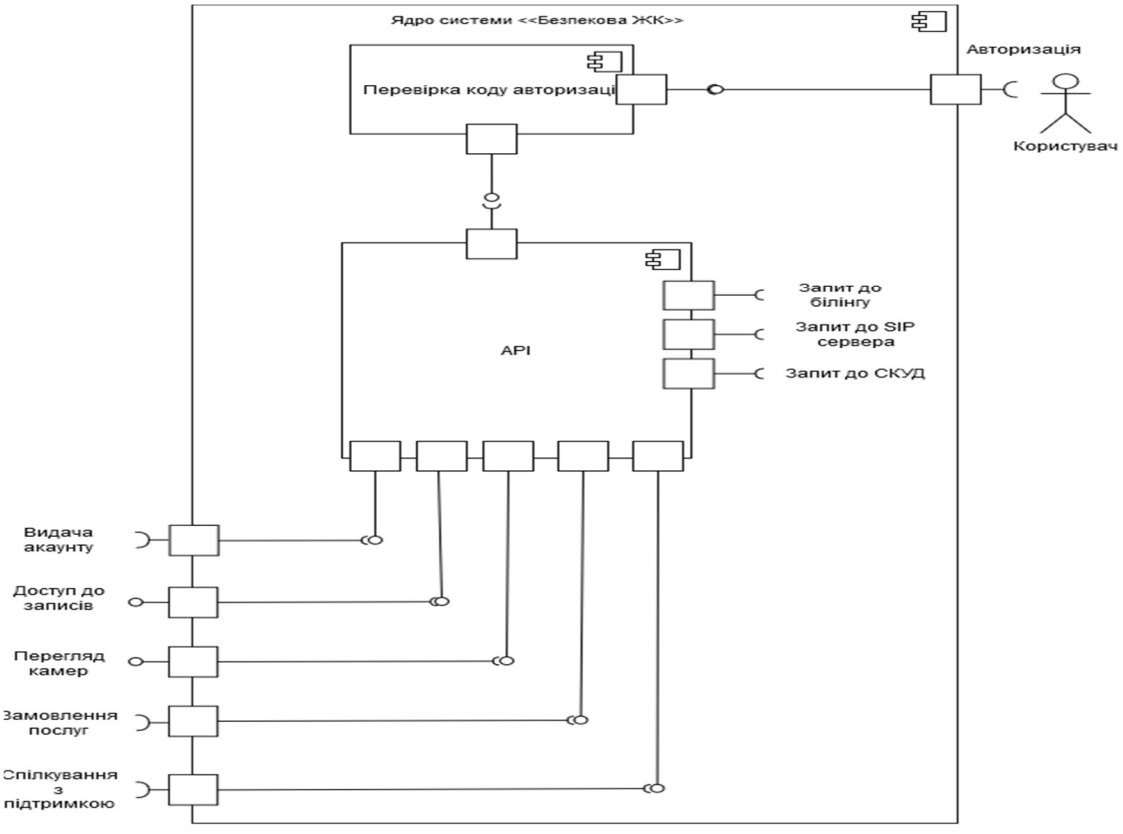


Рис. 5.3 UML діаграма компонентів

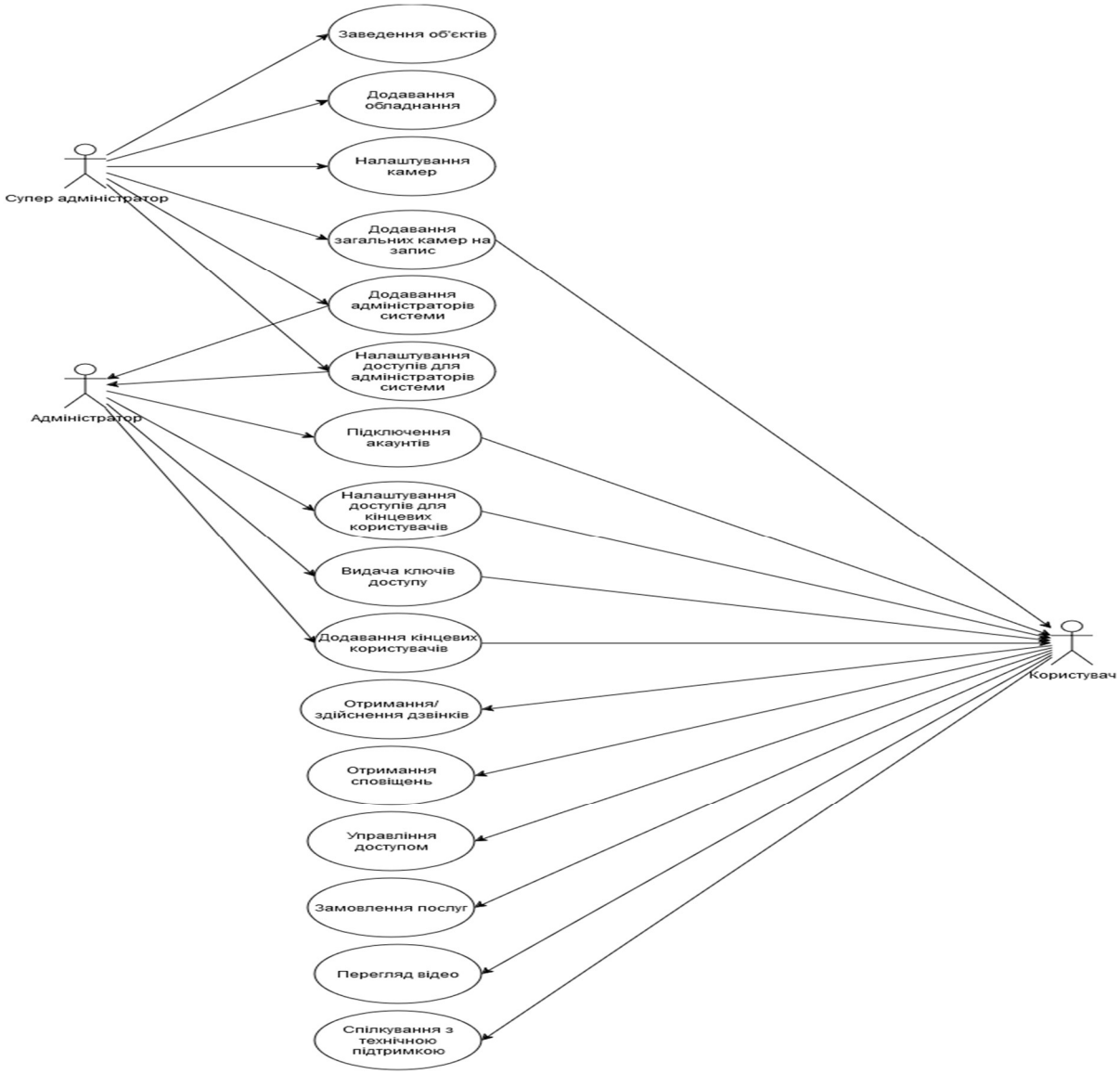


Рис. 5.4 UML діаграма прецедентів підсистеми відеоспостереження

Діаграма прецедентів ілюструє взаємодію користувачів (акторів) із підсистемою відеоспостереження та відображає основні функціональні можливості. Основними акторами є адміністратор системи, який відповідає за налаштування, моніторинг, обслуговування та управління доступом до підсистеми; персонал охорони, який використовує підсистему для моніторингу відеопотоків в реальному масштабі часу та реагування на інциденти; мешканець, що має доступ до перегляду відео з певних камер, зокрема у своєму під'їзді чи біля квартири; та модуль контролю доступу, який інтегрується з підсистемою для автоматизації процесів доступу до території. Актори, взаємодіють із підсистемою через прецеденти, які забезпечують ситуаційну обізнаність. Основні

функціональні можливості підсистеми відображені через ряд прецедентів. Моніторинг відеопотоків в реальному масштабі часу, який доступний адміністраторам і охоронному персоналу, дозволяє здійснювати візуальне спостереження за територією через монітори або мобільний застосунок. Перегляд архівних записів, доступний адміністраторам, охоронцям і мешканцям, дає змогу отримувати доступ до записів подій, що сталися раніше.

Налаштування підсистеми відеоспостереження, яке виконується адміністраторами, охоплює встановлення і конфігурування параметрів відеокамер, таких як якість запису, тривалість зберігання відеоматеріалів, визначення зон детекції руху тощо. Функція детекції руху, що виконується модулем контролю доступу і використовується персоналом охорони, забезпечує автоматичне фіксування рухомих об’єктів в зоні спостереження та надсилання сповіщень. Інтеграція з модулем контролю доступу дозволяє перевіряти відео у разі спрацювання давачів руху, наприклад, під час відкривання дверей. Автоматично відбувається процедура формування звітів про події, яка здійснюється адміністратором, дає змогу генерувати звіти про інциденти, переглядати записи відеокамер або моніторити зони активності. Дистанційний доступ, доступний мешканцям і адміністраторам, забезпечує можливість користування підсистемою з допомогою мобільного застосунку чи веб-інтерфейс. Діаграма прецедентів відображає сценарії використання, які пов’язані з ситуаційною обізнаністю, такі як виявлення аномалій, автоматичне реагування на загрози або прогнозування ризиків.

Діаграма класів підсистеми відеоспостереження моделює її структуру, відображаючи класи, їх атрибути, методи та зв’язки між ними, що дозволяє зрозуміти взаємодію компонентів і функції, які вони виконують. Основними класами є:

Камера (Camera). Атрибути - cameraID (унікальний ідентифікатор камери), location (місце розташування камери), resolution (роздільна здатність, наприклад Full HD, 4K), status (активність камери - увімкнена/вимкнена), recordingMode (режим запису: 24/7 або за детекцією руху). Методи - startRecording() (запуск

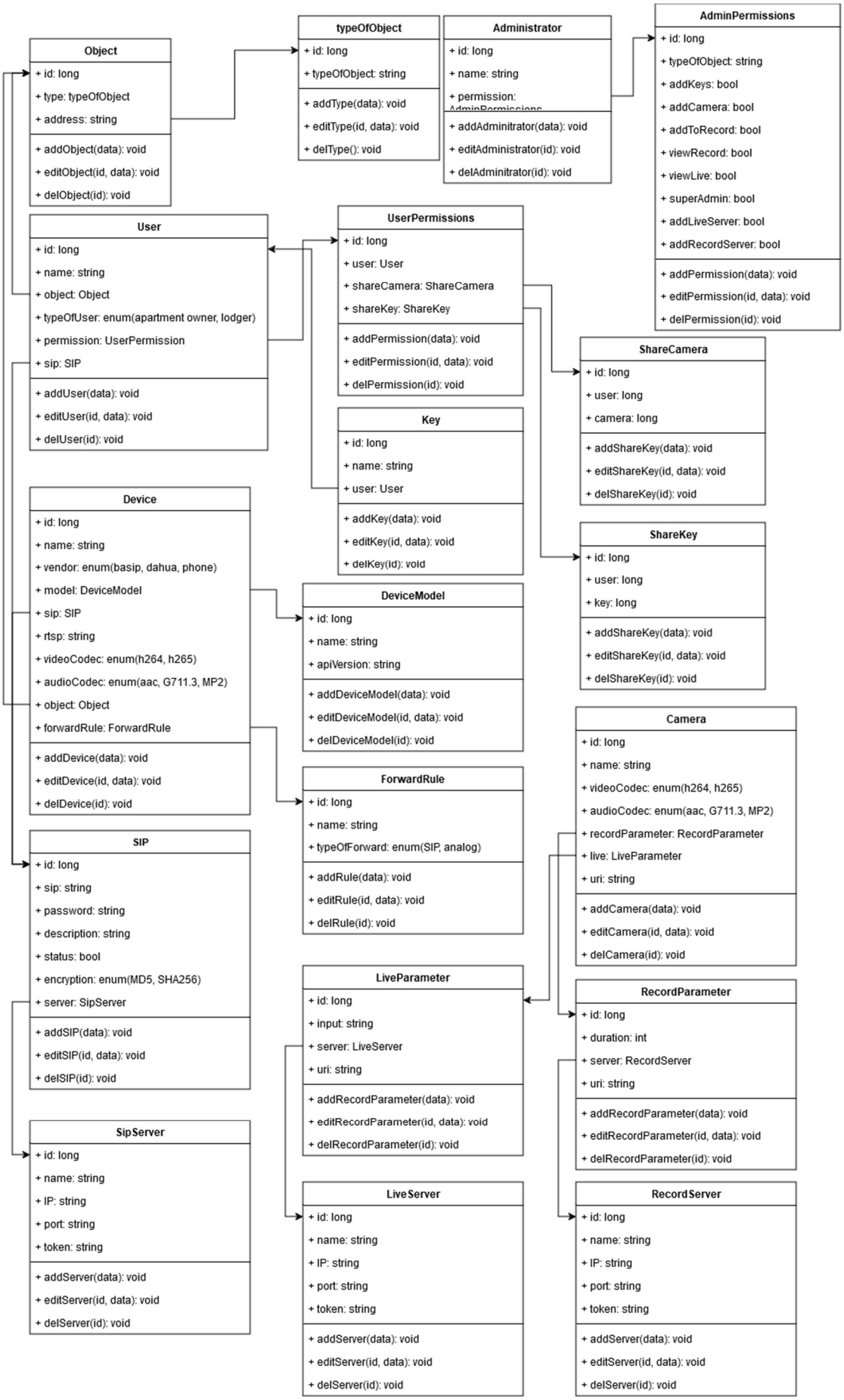


Рис. 5.5 UML діаграма класів підсистеми відеоспостереження

запису), stopRecording() (зупинка запису), adjustSettings(settings) (зміна параметрів).

Архів відеозаписів (VideoArchive). Атрибути - archiveID (ідентифікатор архіву), startTime (час початку запису), endTime (час завершення запису), filePath (шлях до файлу запису). Методи - saveVideo(video) (збереження відео в архів), retrieveVideo(timeRange) (отримання записів за заданим часовим проміжком).

Детектор руху (MotionDetector). Атрибути - detectorID (ідентифікатор сенсора), sensitivity (рівень чутливості), status (стан - увімкнено/вимкнено). Методи - detectMotion() (перевірка наявності руху), sendAlert() (надсилання сповіщення).

Користувач (User). Атрибути - userID (ідентифікатор користувача), role (роль: Адміністратор, Охоронець, Мешканець), accessRights (права доступу, наприклад, доступ до певних камер). Методи - login(credentials) (авторизація), requestVideo(cameraID) (запит відео).

Рис. 5.6 Компоненти підсистеми відеоспостереження.

Система контролю (SystemController). Атрибути - systemID (ідентифікатор системи), status (статус системи - активна/неактивна). Методи - monitorSystem() (контроль стану компонентів), generateReport() (створення звіту про події).

Відношення між класами:

Асоціація - Camera пов’язані із VideoArchive, оскільки камери створюють відеозаписи, які зберігаються в архіві; MotionDetector пов’язаний із Camera для активації запису при виявленні руху; User взаємодіє із SystemController для доступу до функціоналу.

Композиція - SystemController містить список камер (Camera), детекторів руху (MotionDetector) та архівів (VideoArchive).

Наслідування - Administrator, SecurityPersonnel і Resident є підкласами класу User, маючи різні права доступу.

У цьому проекті підсистема відеоспостереження включає ряд ключових компонентів, які забезпечують повний спектр функцій для моніторингу і управління безпекою. На перших етапах реалізації системи були встановлені сучасні камери відеоспостереження, Axis Q6215-LE, які підтримують роздільну здатність до 4K, функції панорамування, нахилу, зуму та нічного бачення. Окрім камер, у загальних зонах будинку були розміщені мікрофони Shure MX396, які забезпечують високоякісний запис аудіо (Рис. 5.6).

Збір даних здійснюється через шлюзи IoT, зокрема, Raspberry Pi 4, які підключені до камер та мікрофонів. Ці пристрої виконують первинне опрацювання даних, включаючи виявлення руху і шуму. Raspberry Pi використовує програмні рішення OpenCV для опрацювання відео і TensorFlow Lite. Це дозволяє скоротити час опрацювання і зменшити обсяги відеоданих, що передаються до хмари. Опрацювання і зберігання відеоданих здійснюється в хмарній інфраструктурі. Віртуальні машини на базі AWS EC2 використовуються для централізованого опрацювання відео та аудіоданих. Спочатку використовуються інстанси типу t3.large, але в разі потреби можливе масштабування до інстансів типу c5.2xlarge для опрацювання більших обсягів даних. Додатково, функції AWS Lambda використовуються для опрацювання

подій, таких як сповіщення про виявлення руху або звуку, що дозволяє автоматично запускати опрацювання даних без потреби в постійному моніторингу з боку користувачів.

Дані з камер і мікрофонів завантажуються в хмарне сховище AWS S3, де зберігаються у форматі, який дозволяє реалізацію швидкого доступу і перегляду архівів. Для зберігання метаданих, таких як налаштування камер, записів подій і користувацьких даних, використовується AWS RDS для PostgreSQL. Для аналітики відео та аудіо даних використовуються сервіси Amazon Rekognition і Amazon Kinesis. Rekognition забезпечує розпізнавання обличчя і аналіз відео, тоді як Kinesis дозволяє опрацьовувати відео- та аудіопотоки в реальному масштабі часу, інтегруючись з Rekognition для додаткового аналізу.

Користувачі можуть взаємодіяти з підсистемою через мобільний застосунок, розроблений за допомогою React Native, який забезпечує доступ до відеопотоків в реальному масштабі часу, отримання сповіщень і налаштування системи. Веб-інтерфейс, створений за допомогою React.js, дозволяє переглядати архіви, аналізувати відеодані і управляти безпекою. Технічні компоненти підсистеми включають декілька ключових елементів, які забезпечують її функціональність і ефективність. Камери спостереження поділяються на купольні, циліндричні, панорамні та PTZ (з поворотом і можливістю збільшення зображення). Вони мають високу роздільну здатністю, наприклад, Full HD або 4K, інфрачервоне підсвічування для роботи в нічний час і захистом стандарту IP66 для використання на відкритому повітрі. Сервер для зберігання даних представлений централізованими або розподіленими відеореєстраторами з підтримкою RAID-масивів, які забезпечують надійність зберігання і захист даних. Програмне забезпечення підсистеми включає модулі управління відео, що дозволяють аналізувати і організовувати доступ до відеозаписів, а також модулі відеоаналітики, які забезпечують інтелектуальні функції, наприклад, пошук за подіями та створення теплових карт руху. Комунікаційна інфраструктура побудована на основі Ethernet-кабелів або бездротових модулів, таких як Wi-Fi або 4G/5G, із підтримкою передачі живлення через Ethernet (Power over Ethernet,

PoE), що спрощує інсталяцію та обслуговування обладнання. Переваги підсистеми відеоспостереження включають профілактику злочинів завдяки постійному моніторингу, що знижує ймовірність протиправних дій, створення доказової бази, оскільки відеозаписи можуть бути використані у випадку розслідування злочинів, забезпечення комфорту та безпеки мешканцям для створення комфортного середовища, а також економічність, завдяки використанню сучасних відеокамер з енергоефективними технологіями, які знижують витрати на обслуговування. Можливі виклики та недоліки імплементації підсистеми включають проблеми приватності, які потребують чіткого регулювання доступу до записів та дотримання норм законодавства; вразливість до атак, що вимагає захисту через шифрування даних та регулярне оновлення програмного забезпечення; а також високу вартість, адже первинна інсталяція та регулярне обслуговування зазвичай є дороговартісним.

Створені UML діаграми позитивно вплинули на процес розроблення підсистеми відеоспостереження, забезпечуючи структурований і зрозумілий підхід до її проектування, впровадження та супроводу. Вони дозволяють сформувати чітке уявлення про структуру підсистеми, її компоненти, функціональність і взаємозв’язки, що допомогло зацікавленим сторонам, включаючи замовників, розробників і користувачів, мати цілісне спільне розуміння мети створення, функціонального наповнення та базових технічних характеристик і параметрів розробленої системи безпеки житлового комплексу.

5.2 Інструменти розроблення бекенду інтелектуальної програмної системи безпеки житлового комплексу

Програмні засоби, використані при розробленні інтелектуальної системи житлового комплексу, детально проаналізовані в роботі [125]. Апаратне забезпечення включає інтелектуальні пристрої та сенсори, мережеве обладнання, серверну інфраструктура, агрегаційні пристрої, клієнтське обладнання.

Для побудови інтелекуальної програмної системи безпеки житлових комплексів використано інформаційно-технологічну платформу Arduino, яка є платформою відкритого коду, яка складається з апаратної частини та програмного

забезпечення для розроблення і завантаження програм на мікроконтролери. Платформа широко використовується для розроблення систем автоматизації та прототипування. Апаратна частина складається з плати Arduino, мікроконтролерів, розширення, давачів і модулів. В розробленій інтелектуальній системі основному використовуються мікроконтролери серії AVR. Платформа надала можливість використання різні типи давачів (температури, вологості, руху, світла тощо) та модулі (Wi-Fi, Bluetooth, GPS). Для розроблення середовища Arduino використано мову програмування IDE, засновану на C/C++. Наявність великої кількість бібліотек спростило процедуру розроблення проекту, оскільки платформа підтримує широкий спектр давачів і модулів.

Крім того, в інтелектуальній системі використано ряд інтелектуальних пристроїв та сенсорів, до яких належать камери відеоспостереження, сенсори руху, біометричні пристрої. Використовуються купольні, циліндричні, панорамні та з функцією панорамування, нахилу і масштабування камери відеоспостереження з роздільною здатністю (Full HD, 4K), підтримкою нічного бачення, IP66 для зовнішнього використання. Інтелектуальна система містить різнотипові сенсори. Сенсори руху забезпечують виявлення активності у зоні спостереження та ініціюють відеозапис або надсилають сигнал тривоги. Екологічні сенсори сприяють моніторингу стану середовища, зокрема вимірюють температуру, вологість, якість повітря. Сенсори доступу інтегруються з дверними магнітними сенсорами або розумними замками для контролю входу/виходу. Біометричні пристрої, зокрема сканери відбитків пальців або системи розпізнавання обличь, здійснюють контроль доступу до житлових приміщень та загальних зон.

Мережеве обладнання включає центральний маршрутизатор, комутатори, мережеві точки доступу. Центральний маршрутизатор виконує функцію маршрутизації між локальними мережами житлового комплексу та провайдером інтернет-послуг. Водночас він забезпечує балансування навантаження та високу якість обслуговування для підтримки стабільності системи. Комутатори використовуються для підключення великої кількості сенсорів, камер і

інтелектуальних пристроїв. Зазвичай це керовані комутатори з підтримкою VLAN, що дозволяє розмежовувати зони відповідальності та гарантувати безпеку. Мережеві точки доступу забезпечують бездротове підключення пристроїв IoT, таких як камери та давачі, та підтримують стандарти Wi-Fi 6 для забезпечення високої швидкості передачі даних. Серверна інфраструктура інтелектуальної системи включає центральний сервер та хмарну інтеграцію. Центральний сервер виконує функції координації роботи всіх компонентів, опрацювання, зберігання та аналізу даних. Він характеризується високою обчислювальною потужністю, підтримкою віртуалізації для запуску декількох сервісів та використанням RAID-масивів для забезпечення надійності даних. Хмарна інтеграція забезпечує використання хмарних сервісів для резервного копіювання та підтримки функції масштабованості. Розроблення концептуальних підходів створення модуля інтелектуальної системи для забезпечення ситуаційної обізнаності враховує необхідність проведення розрахунків вагових коефіцієнтів поставленого завдання для кожного моменту часу, що дозволяє визначати їх пріоритетність [126]. В лінійці агрегаційних пристроїв провідне місце займає центральний хаб IoT, як фізичний пристрій, що виконує функції інтеграції даних, отриманих від різнорідних сенсорів і пристроїв. Він забезпечує опрацювання та маршрутизацію даних від сенсорів і пристроїв до інших компонентів системи, таких як сервери та хмарні сервіси. Центральний хаб обладнаний портами для підключення пристроїв, такими як Ethernet, USB, Zigbee, Z-Wave, вбудованими модулями для бездротових протоколів, зокрема Wi-Fi, Bluetooth, LoRaWAN, а також локальним сховищем даних для тимчасового зберігання інформації. Клас клієнтського обладнання включає різнотипові пристрої мешканців житлового комплексу та охорони.

Для розгортання інтелектуальної системи безпеки з ситуаційною обізнаністю у житловому комплексі використано гібридну інфраструктуру, яка поєднує локальні сервери, хмарні сервіси та розвинену мережеву архітектуру. Залучення потужностей інтернет-провайдера, який обслуговує ОСББ, дозволяє ефективно використовувати вже наявну інфраструктуру, мінімізуючи витрати на

впровадження та підтримку програмної системи. Початкова реалізація інформаційної системи безпеки містить підсистемами контролю доступу та відеоспостереження.

При цьому підсистема контролю доступу включає точки доступу, якими є ворота, двері або шлагбауми, обладнані механізмами керування та ін. Для ідентифікації користувачів використовуються картки або бейджі доступу, які є фізичними маркерами, наприклад, картки RFID, якими користуються мешканці та службовий персонал. Біометричні пристрої, такі як сканери відбитків пальців або системи розпізнавання обличчя, застосовуються для ідентифікації та автентифікації осіб. Панель керування, яка виконує функцію центрального блоку, відповідає за керування доступом, зберігання облікових даних і ведення журналів. Користувачами системи є мешканці або гості, які мають дозвіл або запитують дозвіл на доступ до житлового комплексу, а також охоронці або персонал служби безпеки, які відповідають за ситуаційний моніторинг та прийняття рішень щодо безпеки. В системі також передбачені механізми екстреного реагування для опрацювання сигналів тривоги або порушень безпеки.

Підсистема відеоспостереження складається з камер, розташованих у стратегічних місцях для моніторингу та запису відео. Записане відео зберігається в цифрових або мережевих відеореєстраторах (DVR/NVR). Для активації запису або надсилання сповіщень використовуються давачі руху, які реагують на виявлені рухомі об’єкти. Програмне забезпечення для керування відео (VMS) дозволяє відображати, опрацьовувати та аналізувати відеопотоки. Для перегляду відео в реальному масштабі часу персоналом служби безпеки використовуються монітори, аналіз відео здійснюється за допомогою системи аналітики з використанням методів та засобів штучного інтелекту, що забезпечує виявлення об’єктів, розпізнавання обличь та фіксації аномалії у відео. До складу реалізованої інтелектуальної програмної системи безпеки входять давачі контролю доступу та камери спостереження, бек-енд і фронт-енд частини.

Для розроблення бек-ендової частини інтелектуальної системи безпеки був обраний фреймворк Django, який вирізняється здатністю забезпечувати швидке

розроблення надійних і масштабованих веб-застосунків [127]. Django як високорівневий веб-фреймворком Python, базується на архітектурі «Модель-Вид-Шаблон» (MVT), яка є різновидом класичного шаблону «Модель-Вид-Контролер» (MVC). Django Rest Framework (DRF) надає зручний і гнучкий інтерфейс для опрацювання запитів і керування даними, що робить його зручним для розроблення серверних веб-застосунків.

Основними її компонентами є:

1. Моделі в Django відповідають за керування даними та визначення структури бази даних. Вони представляють сутності програми, такі як користувачі, продукти або ситуації, як класи Python. Кожен клас моделі визначає набір полів, які відповідають стовпцям бази даних. Модель використовує Django ORM (Object-Relational Mapping) для взаємодії з базою даних. ORM дозволяє автоматично перетворювати дані з бази даних в об'єкти Python і навпаки, що значно спрощує роботу з базою даних.

2. Представлення відповідають за опрацювання HTTP-запитів, які надходять від користувачів. У традиційному шаблоні MVC представлення відповідають контролерам. У Django представлення приймають запити, взаємодіють з моделями, виконують необхідну бізнес-логіку та повертають відповідь у формі HTTP-відповіді. Представлення можуть повертати різні типи відповідей, наприклад HTML-сторінки, JSON або навіть переспрямування на інші URL-адреси. Django також підтримує представлення на основі класів, що дозволяє організовувати код більш модульно та повторно використовувати загальну функціональність.

3. Шаблони відповідають за відтворення остаточного HTML, який отримує користувач. Вони містять статичний код HTML, а також динамічні елементи, такі як змінні та теги шаблонів, які використовуються для відображення даних із моделей. Django надає вбудований механізм шаблонів, який дозволив створити повторно використовувані компоненти інтерфейсу. Шаблони можуть бути вкладеними, що дозволяє створювати складні інтерфейси з використанням базових шаблонів для загальних елементів, таких як заголовки або навігаційні меню.

4. Менеджер URL-адрес у Django відповідає за маршрутизацію запитів до відповідних представлень. Він працює за принципом зіставлення регулярних виразів, що дозволяє визначити, які URL-адреси мають опрацьовувати типи. Коли запит надходить у програму Django, менеджер URL-адрес перевіряє його відповідність встановленим правилам і передає опрацювання відповідному представленню.

5. Проміжне програмне забезпечення — це проміжні рівні, на яких опрацьовуються запити та відповіді між сервером і представленнями. Вони можуть змінювати або опрацьовувати дані на різних етапах життєвого циклу запиту. Django підтримує різні типи проміжного ПЗ, такі як авторизація, кешування, опрацювання сеансу та багато іншого.

6. Різні види баз даних, включаючи PostgreSQL, MySQL, SQLite та інші, з якими підтримує роботу Django. Використання ORM дозволяє абстрагуватися від конкретних запитів SQL і працювати з базою даних через зручний інтерфейс Python. Це значно спрощує міграцію даних і підтримку крос-платформних програм.

7. Міграції або керування змінами в схемі бази даних є потужним інструментом, наявних у Django. Щоразу, коли модель змінюється, Django автоматично створює міграцію, яка відображає ці зміни в базі даних. Це дозволяє легко відстежувати та контролювати всі зміни в структурі даних.

Для зберігання інформації використовувалася система управління базами даних PostgreSQL. PostgreSQL — це потужна об’єктно-реляційна система управління базами даних (СУБД) із відкритим кодом, яка є надійною, масштабованою та гнучкою. Система забезпечує повну підтримку транзакцій ACID (Atomicity, Consistency, Isolation, Durability), що гарантує надійність опрацювання даних. PostgreSQL дозволяє розширювати функціонал, додаючи нові типи даних, оператори та індекси, без необхідності змінювати вихідний код. Вона має потужний механізм опрацювання складних запитів, включаючи підтримку під-запитів, об’єднань, агрегатних функцій і багатовимірних масивів. Окрім звичайних типів даних, PostgreSQL підтримує JSON, XML, hstore (ключ-

значення), геометричні типи та дозволяє створювати власні. Система добре масштабується як вертикально на одній машині, так і горизонтально на кількох, забезпечуючи високу продуктивність при зростанні обсягу даних. PostgreSQL підтримує паралельне опрацювання запитів, що значно скорочує час виконання складних запитів для великих обсягів даних. Високий рівень безпеки досягається завдяки можливості налаштування автентифікації, шифрування та контролю доступу до даних на різних рівнях. Крім того, PostgreSQL забезпечує реплікацію даних для високої доступності та аварійного відновлення, що дозволяє зберігати резервні копії та швидко відновлювати систему у разі непередбачуваних ситуацій. Для забезпечення стабільного та безпечного середовища розгортання використовуються операційна система Debian і веб-сервер Nginx. Debian забезпечує стабільну платформу для запуску серверних компонентів, тоді як Nginx забезпечує ефективну службу запитів HTTP, балансування навантаження та кешування.

В проекті використовується RQ Worker (Redis Queue Worker) для опрацювання фонових завдань. Це інструмент для виконання асинхронних завдань, який працює в тандемі з Redis, базою даних, яка функціонує як черга повідомлень. Для взаємодії між серверною і домофонною системами, зокрема BasIP, використовується бібліотека запитів і BasIP API, який забезпечує доступ до різноманітних функцій домофонних систем через HTTP-запити (отримання інформації про стан пристроїв, управління дзвінками, налаштування параметрів тощо). Версія для системи відеоспостереження (Рис. 5.6) виконує завдання проксі-потоків, запису потоків (за необхідності) та надання посилань на архів і нові записи. Для реалізації цих завдань використовуються основні компоненти - система ZoneMinder для архівації, nginx з модулем rtmp для видачі посилань на прямі трансляції, nginx з модулем ngx_http_mp4_module для видачі посилань на архів, API для взаємодії компонентів системи відеоспостереження з іншими системами, а також ffmpeg і ffprobe.

Для ізоляції процесів і спрощення розгортання рішення відеоспостереження були використані можливості докер-платформи -

згенеровано Dockerfile з інструкціями, за допомогою яких створено образ, що містить необхідні пакети для роботи nginx з модулями rtmp і ngx_http_mp4_module. зібрані. На основі зібраного образу запускається контейнер з розділами переадресації портів і монтування, в які скидаються конфігураційні файли nginx і відповідні модулі, логи, а також папка із записами. Це дає змогу вносити зміни з головної операційної системи без необхідності перекомпоновувати образ. Для роботи з потоками та архівами реалізовано інтерфейс прикладного програмування (API), який забезпечує додавання, оновлення та видалення вхідних потоків, перегляд інформації про потоки, що вже опрацьовуються системою, а також виконання архівних операцій.

Новий вхідний потік додається в результаті виконання такої послідовності операцій:

Крок.1. У відповідну точку API надсилається запит, в якому передається назва та URL вхідного потоку.

Крок.2. API створює сервіс на рівні ядра системи, завданням якого є перенаправлення вхідного потоку (push) на nginx модуль rtmp, який працює в докері. Push реалізовано за допомогою інструменту ffmpeg з відкритим кодом. Перед запуском сервісу метадані потоку визначаються інструментом ffprobe. Метадані дають уявлення про структуру вхідного потоку та дозволяють визначити, чи потрібно аудіо перекодувати в підтримувані кодеки. Основний аудіокодек визначається як AAC. H264 використовується як відеокодек.

Крок.3. Після розроблення enpoint для додавання вхідного потоку URL-адреса з модуля rtmp повертається користувачеві.

Крок.4. При додаванні відеопотоку, обирається відповідну функцію запису (безперервний запис або лише рух).

Крок.5. Вхідний потік можна додати до проксі. Тоді стає доступною можливість діяти відповідно до протоколу rtmp. Також до запису можна додати вхідний потік (додається через протокол rtsp). Якщо є потреба додати потік як для проксі, так і для запису одночасно, вхідний потік на zoneminder може бути

rtmp із виводу модуля rtmp. Тобто не потрібно двічі приймати потік з камери чи панелі. Також API надає можливість встановити глибину запису в днях.

5.3 Створення інтерфейсів інтелектуальної програмної системи безпеки житлового комплексу з використанням методу персон

5.3.1 Класифікація потенційних користувачів системи безпеки житлового комплексу

Для розробки інтерфейсної частини продукту обрана популярна бібліотека React.js, а код написаний на TypeScript, який, як суворо типізоване розширення JavaScript, покращив якість коду та зменшив кількість помилок під час розроблення. У поєднанні з React TypeScript сприяв створенню більш надійних і масштабованих програм. React, як бібліотека для побудови інтерфейсів користувача, забезпечила повторне використання компонентів і віртуальної DOM, що покращило продуктивність програми. Інтеграція з TypeScript робить ці програми ще більш гнучкими та надійними. Для швидкого розроблення та складання проекту використано асемблер Vite, який забезпечив швидке збирання та оновлення модулів у реальному масштабі часу. Завдяки Vite процес розроблення став ефективнішим, а результати швидше відображаються в браузері. У проекті використано можливості DOM-бібліотеки React Router для реалізації односторінкової маршрутизації в стилі програми. Це забезпечило плавну навігацію між різними розділами програми без необхідності перезавантажувати сторінки. Завдяки React Router DOM проект реалізує динамічні маршрути, які легко визначаються. Безперервна навігація забезпечує швидке перемикання між сторінками без перезавантаження, створюючи більш інтерактивний досвід для користувача. Вкладені маршрути дозволяють реалізовувати складні навігаційні структури, а механізми захисту маршруту забезпечив обмежений доступ до певних сторінок. Для керування станом даних у застосунках React використано TanStack Query, який дозволив ефективно кешувати та синхронізувати дані, що значно покращило роботу API.

Для здійснення HTTP-запитів до сервера використовується компонент Axios, який надає гнучкий інтерфейс для взаємодії з бекендом. Для зберігання

стану у зовнішній частині використовується бібліотека Zustand. Вона забезпечує простий і гнучкий підхід до керування станом у програмах React, що спростило

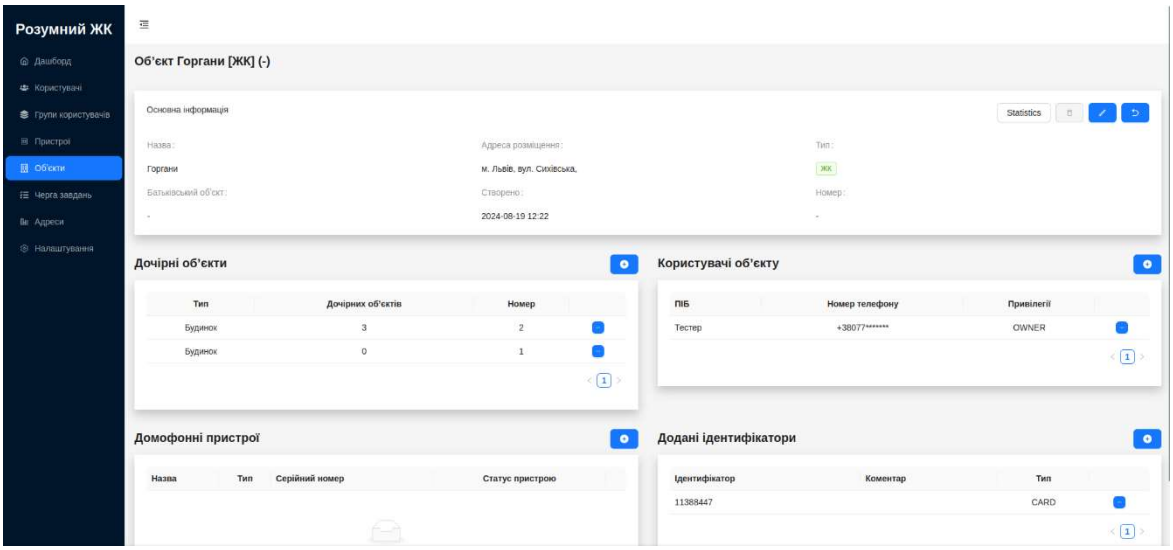


Рис. 5.7 Список зареєстрованих об’єктів

архітектуру та покращило продуктивність програми. Ant Design використовується в проекті як популярна бібліотека компонентів для React, яка забезпечила високоякісні готові до використання компоненти інтерфейсу користувача. Вона дозволила створити сучасний мінімалістичний дизайн із можливістю легкого налаштування тем, а також інтегрується з іншими інструментами для створення ефективних і елегантних веб-застосунків. Використання Ant Design допомогло забезпечити стабільну взаємодію з користувачем і значно прискорило процес розроблення.

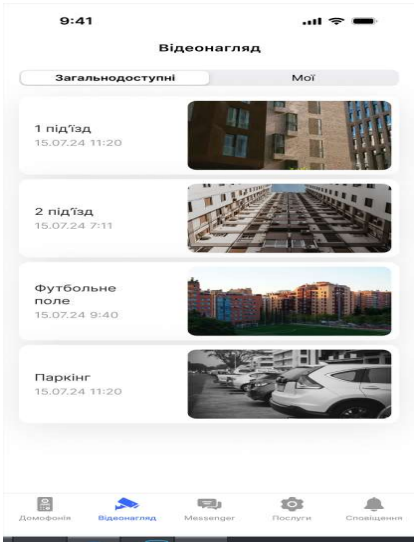


Рис. 5.8 Екран програми мобільного спостереження

На рис.5.7 подано список зареєстрованих об’єктів з відповідними властивостями в інтерфейсі керування програмою. На рис.5.8 – один із екранів у програмі мобільного спостереження, що дозволяє відстежувати всі спостережувані місця. Центральним елементом є інтерфейс користувача, який повинен бути інтуїтивно зрозумілим, зручним у використанні та відповідати специфічним особистісним вимогам найрізноманітніших груп користувачів.

Метод персон є одним з ефективних інструментів проектування користувацьких інтерфейсів, оскільки дозволяє створювати інтерфейси, що точно та в повній мірі відповідають потребам і очікуванням кінцевих користувачів. Цей метод базується на глибокому аналізі різних категорій та груп користувачів та формуванні уявних персонажів (персон), що представляють типові моделі поведінки, мотивації та потреб. Розглянемо особливості створення інтерфейсів інтелектуальних програмних систем безпеки житлових комплексів з використанням методу персон. Зокрема, як саме зазначений метод дозволяє адаптувати інтерфейс під специфічні вимоги різних груп користувачів, таких як мешканці, адміністратори будинків та безпекові служби. Майбутніх користувачів безпекової системи житлових комплексів можна класифікувати за декількома критеріями, враховуючи їхні потреби, роль у її використанні та особливості взаємодії з нею [128], зокрема це стосується інвалідів [129]. Класифікація може проводитись з використанням широкого спектру методів та способів. Такого роду класифікація користувачів загалом сприяє побудові множини «дружніх» та зручних інтерфейсів, зорієнтованих на відповідні цільові групи. Проводячи класифікацію за статусом користувача можна виділити наступні групи протоперсон. Першу групу протоперсон складають окремі мешканці, серед яких доцільно зафіксувати молодих професіоналів, представниками такої групи для прикладу буде Олександр, членів молодої сім'ї представником буде Марія, пенсіонерів, представником яких буде Павло. Другу групу протоперсон складають члени адміністрації будинку, зокрема Анна. Третю групу складають власники квартир, що здаються в оренду, представляти цю групу буде, зокрема, Ігор.

Класифікуючи потенційних користувачів за потребами у галузі безпеки можемо виділити ряд груп протоперсон. До першої належать ті, що переслідують мету особистої безпеки. До другої - належать протоперсони, яким необхідна безпека родини. Третя група протоперсон надає перевагу та презентує потреби в сфері безпеки майна. Четверта група потенційних користувачів артикулює потреби та турботи в контексті безпеки мешканців будинку загалом. (Класифікація за іншими критеріями див. Додаток Г). На основі наведених критеріїв класифікації будуємо узагальнену класифікаційну схему.

Таблиця 5.1 Узагальнена класифікаційна схема

Критерій	Групи протоперсон та специфічні ознаки
Тип користувача	Окремі мешканці, Адміністратори, Власники квартир
Потреби у безпеці	Особиста безпека, Безпека родини, Безпека майна, Безпека мешканців
Рівень технічних знань	Високий, Середній, Низький
Способи взаємодії	Віддалене, Локальне
Мотиваційні фактори	Особистий простір, Родина, Надзвичайні ситуації, Управління будинком, Оренда
Фінансові можливості	Високий, Середній, Низький

Ця класифікація допомагає краще зрозуміти вимоги та очікування користувачів, що, в свою чергу, дозволяє запропонувати адаптивні та ефективні рішення для безпекової системи житлового комплексу. Проте формування таких класифікацій в ручному режимі є доволі трудомістким процесом. Для спрощення процедури побудови моделі класифікації користувачів системи безпеки житлового комплексу пропонується використання методу випадкового лісу.

5.3.2 Метод випадкового лісу в процесах класифікації користувачів системи безпеки житлового комплексу

Метод випадкового лісу є потужним інструментом, який використовується для вирішення задач класифікації та регресійного аналізу. Він базується на множині дерев рішень, де кожне дерево будується на основі випадкової вибірки з навчального набору даних. Рішення, прийняті на основі різних дерев,

об’єднуються для формування остаточного прогнозу. Такий підхід знижує рівень ризику перенавчання та підвищує точність моделі. У даному випадку користувачами системи безпеки можуть бути мешканці, гості, працівники технічного обслуговування, охоронці тощо. Задача полягає у точній класифікації потенційних користувачів на основі наборів певних ознак з метою підвищення рівня їх безпеки та зручності використання повного функціоналу системи безпеки. Для побудови адекватної та повної моделі класифікації можуть бути використані ідентифікаційні дані, певні поведінкові, психологічні, системні та технічні характеристики та параметри. Ідентифікаційні дані включають вік, стать, приналежність до певної групи (мешканці, гості, персонал). Поведінкові характеристики формуються на основі низки параметрів, таких як час входу та виходу з будівлі, частота візитів, місця, які відвідуються в середині будівлі. Формування наборів технічних параметрів передбачає наявність відбитків пальців, даних з камер спостереження, карт- ключів або мобільних застосунків для доступу. Загальними перевагами методу випадкового лісу є висока точність процедур класифікації, можливість опрацювання великої кількості ознак, стійкість до шуму, масштабованість.

Використання потужної множини дерев забезпечує високу точність і стабільність результатів. Метод є менш чутливим до шуму та нерелевантних ознак у даних, що знижує ймовірність формування помилок при класифікації, і може бути застосований до великих наборів даних, що є критично важливим в багатокористувацьких системах безпеки. У системах безпеки житлових комплексів дані класифікації зазвичай використовуються в процесах контролю доступу, аналізу поведінки мешканців та відвідувачів, персоналізації при наданні тих чи інших послуг.

Класифікація користувачів проводиться на основі попередньо визначених правил доступу як то мешканці, наділені правом повного доступу, гостям зазвичай надається право обмеженого доступу. Виявлення аномальних поведінкових патернів може вказувати на потенційні загрози або порушення, надання індивідуалізованих послуг користувачам, таких як автоматичне

відкриття дверей або доступ до певних зон. При цьому варто враховувати, що точність класифікації залежить від якості і кількості навчальних даних. Процеси збору і опрацювання персональних даних потребують дотримання законодавчих норм і захисту приватності.

У житловому комплексі проживають мешканці, які належать до різних груп користувачів, що може слугувати основою формування окремих типів протоперсон, кожна з яких має унікальні потреби, мотивації та проблемні моменти, пов'язані з безпекою. Важливо класифікувати користувачів для персоналізації інтерфейсів системи безпеки, що дозволяє забезпечити більш ефективне їх обслуговування і підвищити рівень безпеки та комфорту. Джерелом даних для класифікації та формування протоперсон є відомості про мешканців, що наявні у керівництва ОСББ. Для ефективної класифікації користувачів (протоперсон) у системі безпеки житлового комплексу необхідно попередньо визначити ключові критерії, які можуть впливати на результати класифікації. Наведемо послідовність кроків процесу визначення особливостей протоперсон.

Крок 1. Збір даних та ідентифікація потенційних критеріїв.

Крок 2. Аналіз даних для кожної протоперсони.

Крок 3. Вибір ключових критеріїв для класифікації.

Крок 4. Використання обраних критеріїв для навчання моделі.

На першому кроці було окреслено критерії та їх складові, які використовувалися для класифікації протоперсон.

Критерій a. Демографічні дані – Вік, Сімейний стан, Наявність дітей.

Критерій b. Професійні характеристики - Рід занять, Спосіб роботи (віддалено/офлайн), Графік роботи.

Критерій c. Житлова ситуація - Тип житла (власне/орендоване), Тип будинку (сучасний, новобудова, радянська забудова).

Критерій d. Технічні потреби та звички - Використання мобільних застосунків, Частота подорожей, Необхідність віддаленого контролю.

Критерій е. Особисті потреби і пріоритети - Фокус на безпеку дітей, Власна безпека, Зручність використання, Вартість послуг із використання системи безпеки.

На другому кроці було проведено аналіз даних щодо кожної протоперсони. Його провено на основі кількох ключових джерел інформації, що дозволило сформувати детальний профіль і чіткіше означити потреби, поведінку та пріоритети кожного сегменту користувачів. Основними етапами такого аналізу є:

- збір демографічних даних, який дозволяє з'ясувати вік, стать, сімейний стан, наявність дітей, рівень доходу тощо. Ці дані дають загальне уявлення про основні характеристики користувачів і допомагають зрозуміти, як їхні потреби можуть змінюватися залежно від життєвих обставин.

- збір психографічних даних, який включає вивчення цінностей, інтересів, стилю життя, а також ставлення до нових технологій чи систем безпеки. Психографічні дані дозволяють зрозуміти мотивацію користувача та його пріоритети при виборі певних продуктів чи послуг.

- визначення поведінкових даних, які характеризують попередню взаємодію з продуктом або сервісом (частота використання, функції, що використовуються найчастіше, тривалість сесій тощо), дозволяє зрозуміти, як користувач взаємодіє з продуктом і які аспекти є важливі для нього.

- аналіз цілей та мотивів користувачів, наприклад, захист житла, забезпечення комфорту чи контроль за дітьми сприяє розумінню основних мотивів та дозволяє вибрати критерії, які безпосередньо пов'язані з вирішенням ключових завдань протоперсони.

- аналіз існуючих рішень на ринку та способів, якими користувачі взаємодіють з конкурентними продуктами. Це допомагає визначити, що подобається або не подобається певним групам користувачів, і, відповідно, які функції можуть бути кориснішими або менш актуальними.

- оцінювання технологічних знань і досвіду користувача, яке полягає у вивченні рівня володіння користувачами інформаційних технологіями, їхнього досвіду використання подібних систем, рівня комфорту при роботі з новими

технологіями. Це допомагає адаптувати параметри системи під категорії користувачів, що є важливим для вибору функцій інтерфейсу, зручності його використання та інших аспектів.

- аналіз сценаріїв використання дозволяє оцінити ситуації, у яких користувачі можуть використовувати продукт чи послугу. Це можуть бути такі функції, як спостереження за будинком під час відпустки, моніторинг місця перебування дітей або аналіз безпекової ситуації вночі. Залежно від таких сценаріїв, можна визначити, які критерії будуть найбільш корисними для кожної протоперсони.

Проведення аналізу на основі вищенаведених даних допомагає сформувати чіткий і детальний профіль протоперсони, що своєю чергою забезпечує вибір релевантних критеріїв для класифікації та персоналізований підхід до кожної категорії користувачів. На основі детального аналізу сформовано протоперсони основних категорій користувачів безпекової системи житлового комплексу.

Протоперсона 1: Олександр, молодий професіонал

Вік: 28 років.

Рід занять: ІТ-спеціаліст, працює віддалено.

Демографічні дані: Неодружений, проживає один.

Житлова ситуація: Орендує квартиру в сучасному багатоквартирному будинку.

Потреби: Безпека особистого простору, зручність використання безпекової системи, можливість віддаленого контролю через мобільний застосунок.

Технічні потреби: Часто подорожує і потребує можливості віддаленого моніторингу та управління безпекою квартири.

Мотивація: Забезпечення безпеки власних речей, зниження тривожності щодо безпеки квартири під час відсутності, віддалений моніторинг безпекової ситуації. (Протоперсона 2-5 див. Додаток Г).

Сформовані протоперсони сприяють розумінню різноманіття потреб та очікувань користувачів безпекової системи житлового комплексу, що є важливим для розроблення ефективних та зручних у використанні проектних рішень.

Третій крок передбачає вибір ключових критеріїв класифікації для успішної реалізації методу випадкового лісу. Він реалізується після аналізу даних кожної протоперсони і дозволяє з'ясувати індивідуальні потреби та особливості, провести контекстуалізацію критеріїв, їх персоналізацію та сегментацію, оптимізацію алгоритмів класифікації. Вибір ключових критеріїв після проведення аналізу даних протоперсон дозволяє підвищити точність класифікації, персоналізувати інтерфейси системи безпеки під потреби різних категорій користувачів і до певної міри уникнути надлишкової складності моделі.

На четвертому кроці обрані критерії використовуються для створення та навчання моделі на основі методу випадкового лісу, що допомагає класифікувати нових користувачів і забезпечити надання їм персоналізованих послуг безпеки, адаптованих до індивідуальних потреб і пріоритетів (Додаток Д. Код 2).

5.3.3 Використання визначених критеріїв для навчання моделі випадкового лісу

При використанні методу випадкового лісу для навчання моделі можна автоматично оцінювати важливість ознак. Згідно прогнозу кожного дерева у випадковому лісі приймається рішення на основі різних наборів ознак, що дозволяє оцінити, наскільки часто і як ефективно кожна ознака використовується для класифікації. Цей підхід дозволяє не тільки визначити важливі ознаки, але й зрозуміти, як вони впливають на процес класифікації, що сприяє покращенню моделі і підвищенню якості прийнятого рішення. На другому кроці модель випадкового лісу будується на основі навчального набору даних, що містить критерії та мітки класів. Модель складається з множини дерев рішень, кожне з яких навчено на випадковій вибірці даних з навчального набору. На наступному кроці навчена модель використовується для класифікації нових користувачів на основі їхніх даних, щоб надавати їм персоналізовані безпекові послуги.

Покроковий сценарій створення та навчання моделі за методом випадкового лісу з використанням визначених критеріїв укрупнено реалізується наступним кроком.

Крок 1. Підготовка даних;

Крок 2. Створення та навчання моделі;

Крок 3. Використання моделі для класифікації нових користувачів.

Виконання наведених кроків дозволило ефективно використовувати метод випадкового лісу для класифікації користувачів безпекової системи житлового комплексу. Реалізація методу випадкового лісу може бути представлена у вигляді такої послідовності дій:

Кожне дерево в лісі обирає підмножину даних для навчання (із заміною) і випадковим чином - підмножину функцій для кожного розбиття. Нехай N – кількість дерев у лісі, n – кількість точок даних у підмножині, m – кількість об'єктів у підмножині. Для кожного дерева k в лісі: вибирається випадкова підмножина даних D_k розміру n та випадкова підмножина функцій F_k розміру m. Дерево рішень будується за допомогою D_k і F_k.

Класифікація проводилась наступним чином:

Для кожного дерева k в лісі:

Класифікація відбувалася з використанням дерева для нового спостереження. Усереднюються показники або застосовується метод регресії для всіх дерев, щоб отримати остаточний результат.

Результат класифікації ŷ для об'єкта x можна подати як:

$$\hat{y} = \frac{1}{N}\sum_{k=1}^{N} N_k(x), \tag{5.1}$$

де $y_{k(x)}$ – результат класифікації від k-дерева.

Підхід гарантує, що сформована модель випадкового лісу може впоратися зі складністю та мінливістю даних користувачів.

$$\hat{y} = \mathrm{mode}(\hat{y}_1, \hat{y}_2, \ldots \hat{y}_k), \tag{5.2}$$

де $\hat{y}_k$ – результат, отриманий з дерева k, а режим (mode) – це функція, яка повертає значення, яке найчастіше зустрічається серед них.

Це є загальною формулою реалізації методу випадкового лісу для класифікації користувачів системи безпеки житлового комплексу. Конкретні параметри, такі як кількість дерев у лісі, критерії вибору ознак тощо, можуть змінюватися залежно від конкретної реалізації та потреб процедур аналізу даних.

Зібрані дані по кожній протоперсоні подаються в структурованому вигляді для формування та навчання моделі. Кожен рядок набору даних відповідає окремому користувачеві (протоперсоні), а кожен стовпець - критерію, який характеризує конкретного користувача. При цьому зазначимо, критерії можуть бути як числовими, так і категорійними (Додаток Е).

Приклад розширеного набору даних для кожної протоперсони, включаючи додаткові записи для підвищення різноманітності та розміру набору даних подано у Додатку Є. Для кожної протоперсони дані розширені шляхом варіації деяких характеристик, таких як вік, тип житла та інше. Розширений набір даних було використано для навчання моделі, побудованої за методом випадкового лісу, що дозволило точніше класифікувати нових користувачів та в подальшому надавати їм персоналізовані послуги з безпеки на основі повнішого набору характеристик.

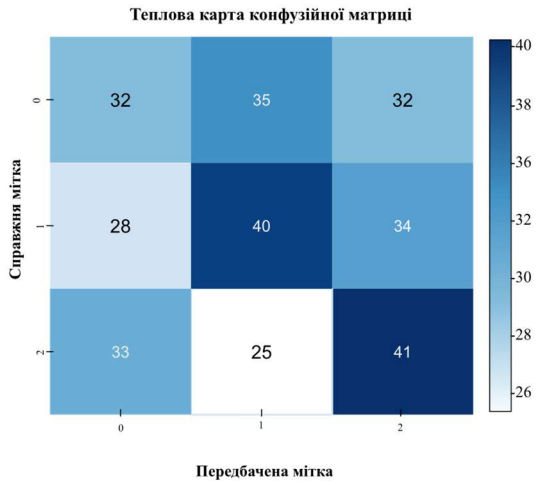


Рис.5.9. Теплова карта конфузійної матриці класифікації

Після навчання моделі нові користувачі можуть бути класифіковані на основі введених даних, що дозволило персоналізувати надання безпекових послуг. Кожен рядок у вибірці відповідає окремому користувачеві (протоперсоні), а кожен стовпець — це критерій, що характеризує цього користувача. Набір включав як числові (наприклад, вік), так і категорійні (наприклад, тип житла) ознаки, які були закодовані чисельно для роботи з

моделлю. Конфузійна матриця, подана на тепловій карті (Рис.5.9), ілюструє точність класифікації кожного класу. Найвпливовішими ознаками для класифікації були: вік (28.17%), рід занять (14.01%), бюджет (13.24%), частота подорожей (11.83%).

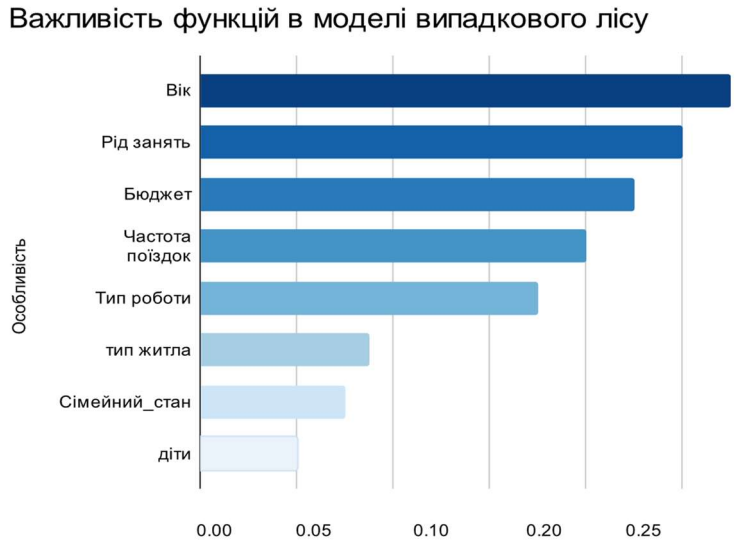


Рис. 5.10. Графік значущості ознак для класифікації

Навчання моделі проводилося таким чином: вхідний набір даних складався з 1000 спостережень, кожне з яких описувало окремого користувача. Дані були поділені на навчальну вибірку, яка становила 70% (700 записів), і тестову вибірку, що складала 30% (300 записів). Навчальна вибірка складалася з 700 спостережень, які включали дані про мешканців, такі як вік, сімейний стан, наявність дітей, рід занять, тип роботи, тип житла, частота подорожей, бюджет, а також мітку класу для класифікації. Ці дані були випадковим чином відібрані з повного набору відомостей про мешканців житлового комплексу, наданих ОСББ, щоб забезпечити репрезентативність для навчання моделі. Всі категорійні ознаки, такі як "Сімейний стан" або "Тип роботи", були закодовані чисельно. Для навчання використовувався метод випадкового лісу з такими параметрами: кількість дерев становила 100, критерієм розбиття за замовчуванням був коефіцієнт Джині, а дані та ознаки вибиралися для кожного дерева випадковим чином. У процесі навчання кожне дерево будувалося на основі випадкової

підмножини даних із заміною і використовувало випадкову підмножину ознак для розбиття у кожному вузлі. При цьому оцінювалися ваги ознак для визначення їхньої важливості в класифікації. Модель будувала ансамбль із 100 дерев рішень на основі навчальної вибірки. Оцінка результатів включала розрахунок таких метрик, як загальна точність, що відображає частку правильних класифікацій, класифікаційний звіт, який містить метрики точності, повноти і F1-міри для кожного класу, а також аналіз конфузійної матриці для оцінки помилок класифікації між класами. Важливість ознак автоматично оцінювалася в процесі навчання, і ознаки з високою важливістю мали більше значення для прийняття рішень моделлю. Цей підхід дозволив ефективно навчити модель, яка здатна точно класифікувати нових користувачів і забезпечувати персоналізовані послуги на основі введених даних.

Точність є однією з ключових метрик оцінки моделі. Значення точності було визначено шляхом порівняння передбачених класів із реальними мітками класів у тестовій вибірці. Навчена модель випадкового лісу була застосована до даних тестової вибірки, щоб отримати передбачені класи для кожного спостереження. Передбачені класи порівнювалися з фактичними мітками класів (реальні значення з тестового набору даних). Загальна точність розраховувалася за формулою (5.3), модель правильно класифікувала 267 із 300 спостережень у тестовій вибірці.

$$\text{Accuracy} = \frac{\text{Кількість правильних передбачень}}{\text{Загальна кількість передбачень}} = \frac{267}{300} \approx 0{,}89 \qquad (5.3)$$

Оскільки було використано навчальну вибірку з 700 спостережень і тестову вибірку з 300 спостережень. Загальна точність моделі склала 89%, що свідчить про високу ефективність методу випадкового лісу. Базуючись на цих результатах сформовано висновок, що модель може використовуватись для ефективної класифікації нових користувачів і надання персоналізованих послуг в системі безпеки житлового комплексу.

Інтелектуальна система безпеки житлового комплексу налаштована на персоналізоване обслуговування користувачів на основі їхніх особистісних характеристик. За допомогою моделі машинного навчання (випадковий ліс)

система класифікує нових користувачів з метою надання їм безпекових послуг, таких як віддалений доступ, налаштування сповіщень та інших.

Класифікація нових користувачів відбувається за наступною послідовністю дій.

Крок 1. Збір даних про нового користувача**.**

Коли новий користувач реєструється або взаємодіє з системою відбувається збір та консолідація відомостей про нього. Це можуть бути дані, зібрані шляхом анкетування або запозичені з інших джерел.

Крок 2. Перетворення даних у формат, придатний для використання.

Перед тим як передати дані, їх потрібно перетворити у прийнятний для моделі формат. Це передбачає реалізацію кроків по кодуванню категорійних даних, таких як сімейний стан, тип роботи, у числовому форматі.

Крок 3. Прогнозування за допомогою моделі

Використання попередньо навченої моделі випадкового лісу для класифікації нового користувача. Модель прогнозує клас, до якого належить новий користувач на основі його особистісних характеристик.

Крок 4. Надання персоналізованих послуг

За результатами процедури класифікації новому користувачеві можуть бути надані певні послуги або доступ до функцій системи безпеки, які відповідають його класу. Користувачу 1, що класифікований як "Молодий професіонал", система пропонує доступ до функції віддаленого моніторингу та керування через мобільний застосунок, оскільки такі користувачі часто подорожують і потребують експериментального доступу до системи. Користувачу 2, який класифікований як "Адміністратор" або "Пенсіонер", система може надати доступ до функції централізованого моніторингу безпеки або функції простого інтерфейсу для спрощення процесу використання. Класифікація нових користувачів з використанням методу випадкового лісу дозволяє адаптувати послуги інтелектуальної системи безпеки до індивідуальних потреб користувачів, підвищуючи рівень їх задоволеності та ефективність

системи в цілому. Це також допомагає зосередити ресурси на найбільш релевантних функціях для кожного сегмента користувачів.

На четвертому кроці проводиться персоналізація послуг на основі прогнозованого класу. При цьому система безпеки може надати персоналізовані налаштування та рекомендації. Наприклад, для молодого професіонала можуть бути запропоновані послуги віддаленого моніторингу, а для пенсіонера — простий і зрозумілий інтерфейс системи безпеки.

Метод випадкового лісу сформований на початковому наборі даних (Табл.5.1) досяг лише нульового рівня точності через надто малий розмір початкового набору даних, який містить декілька прикладів для кожної з категорій. Це в свою чергу призводить до виникнення додаткових проблем з генералізацією моделі. Збільшення кількості даних для тренування моделі (Додаток Д. Табл. Д.3) суттєво покращило результати її застосування. Разом з тим, додавання більшої кількості релевантних критеріїв може сприяти процесу кращого розділення категорій. Це особливо підкреслює важливість використання загалом достатньо об’ємних і різнотипових даних для навчання моделей машинного навчання (Додаток Е). Після розширення набору даних модель випадкового лісу забезпечила більшу точність на тестовому наборі даних. Модель змогла правильно класифікувати майже всі приклади в тестовому наборі, використовуючи визначені критерії. У цьому прикладі дані були розширені шляхом додавання більшої кількості прикладів для кожної категорії користувачів. У реальних умовах це може включати збір додаткових даних або використання технік аугментації даних. Для побудови застосунку, особливо з високою варіабельністю, необхідно використовувати великі набори даних, щоб модель могла адекватно навчатися і узагальнювати інформацію.

5.3.4 Сценарій персоналізації послуг у системі безпеки житлового комплексу

Після класифікації користувачів на основі їхніх характеристик, система безпеки може персоналізувати послуги для кожної категорії користувачів (Додаток Ж). Це дозволяє забезпечити ефективніше використання системи,

підвищити рівень задоволеності користувачів і забезпечити їхні конкретні потреби. На етапі персоналізації послуг для кожної класифікаційної категорії визначаються ключові потреби та пріоритети користувачів. Ці потреби можуть варіюватися від віддаленого доступу і моніторингу до зручності використання і забезпечення додаткових функцій безпеки. На основі потреб і пріоритетів розробляються набори послуг і функції, які можуть бути запропоновані користувачам у кожній з категорій.

Послуги імплементуються в безпековій системі і стають доступними для користувачів на основі проведеної класифікації. Це може включати налаштування особистого профілю користувача, в якому зазначаються доступні функції і налаштування. Після впровадження персоналізованих послуг реалізується функція зворотного зв'язку від користувачів для оцінювання ефективності наданих послуг і задоволеності ними. На основі цих даних система може налаштовуватись або вдосконалюватись, з метою кращого задоволення потреб.

Висновки до розділу 5

Використання IoT пристроїв у підсистемі відеоспостереження забезпечує багатофункціональність і автоматизацію процесів моніторингу. Інтеграція ситуаційного аналізу сприяє швидкому виявленню, прогнозуванню та реагуванню на потенційні загрози. Використання UML-діаграм у процесі концептуального моделювання забезпечило чітке визначення компонентів системи, їхніх функцій та взаємозв’язків, знижуючи ризики на етапах проектування та реалізації. Концептуальне моделювання підтримує можливість масштабування системи шляхом додавання нових IoT-пристроїв та функціональних модулів, дозволяючи адаптувати систему до змінних вимог користувачів та умов середовища. Інтеграція підсистеми відеоспостереження з іншими компонентами системи безпеки забезпечує реалізацію комплексного підходу до захисту об’єктів і територій, роблячи систему здатною вчасно реагувати на критичні ситуації. Запропонована концептуально модель слугувала основою для розроблення високотехнологічної системи відеоспостереження, яка

є однією з ключових компонентів інтелектуальної системи безпеки житлових комплексів або промислових зон, розумних міст.

Ефективна взаємодія між бек-ендом і інтерфейсами має вирішальне значення для успішного розроблення веб-застосунків. Використання Django Rest Framework (DRF) для створення RESTful API спрощує опрацювання запитів із інтерфейсу, а React у поєднанні з TanStack Query та Axios забезпечує безперебійну інтеграцію API. Використання методу персон у процесі розроблення інтерфейсів систем безпеки багатоквартирних будинків дозволяє глибше зрозуміти потреби різноманітних категорій користувачів. Це сприяє формуванню інтуїтивно зрозумілих та персоналізованих рішень, які враховують специфічні вимоги кожної з категорій потенційних користувачів.

Інтерфейси, створені на основі аналізу персон, забезпечують підвищенню користувацької задоволеності. Це досягається шляхом врахування життєвих сценаріїв і звичок користувачів, що дозволяє зробити їх взаємодію з системою більш природною та ефективною.

Використання методу персон при створенні інтерфейсів систем безпеки забезпечує гнучкість процесів розробки продукту, дозволяючи легко адаптувати його під нові потреби користувачів або технологічні зміни. Метод персон дозволяє створювати інтерфейси, які легко масштабуються під інші сегменти ринку або житлові умови. Персоналізація послуг у системі безпеки багатоквартирного будинку забезпечує підвищення ефективності системи, забезпечує кращий користувацький досвід і підвищує рівень безпеки.

Переваги використання методу випадкового лісу можна окреслити як гнучкість і точність, захист від перенавчання, можливість інтерпретації. Модель ефективно працює з великими наборами ознак, забезпечуючи високу точність класифікації. Завдяки використанню багатьох дерев рішень, метод має високу стійкість до перенавчання. Аналіз важливості ознак допомагає краще розуміти вплив різних факторів на класифікацію, що може бути корисним в процесі подальшого вдосконалення системи безпеки.

ВИСНОВКИ

У дисертаційній роботі розв’язано важливу науково-прикладну задачу підвищення ефективності, надійності та адаптивності систем безпеки житлових комплексів шляхом розроблення та інтеграції інтелектуальних методів аналізу, прогнозування загроз і прийняття рішень на основі опрацювання великих обсягів даних, просторово-часових характеристик і поведінкових моделей користувачів. В дисертації одержано такі результати:

Проаналізовано широкий спектр наукових праць в галузі систем безпеки, в яких подані оригінальні методи та засоби їх побудови, розглянуто їхню застосовність в контексті житлових комплексів. Проведений аналіз архітектурних рішень для побудови таких систем.

Використовуючи метод аналізу ієрархій, визначено кращу платформу для реалізації інформаційної системи безпеки житлового комплексу. Проведений SWOT аналіз методологій управління IT проектами сприяв обранню методології DevOps, як базової, що забезпечує ефективну реалізацію функцій безперервної інтеграції, тестування та доставки. Завдяки поєднанню методів аналізу ієрархій та експертного оцінювання сформовано комплекси програмних інструментів методології DevOps та обрано кращий серед них.

Сформовано структуру інформаційної системи безпеки, яка складається з Інтелектуальних агентів та давачів, внутрішнього ПЗ та блоку управління, а також архітектуру, яка містить підсистеми відеоспостереження, контролю доступу і управління послугами та керується центральним блоком.

З використанням ситуоїдного об'єкта GFO розроблено моделі прогнозування та передбачення на основі баз знань. Визначено адитивно-обернену метрику, як один з інструментів обчислення відстаней між концептами онтології.

Досліджено, опрацьовано і структуровано значну кількість безпекових сценаріїв, на основі яких спроектовано відповідні ситуоїди, які подано конструкціями мови OWL та у вигляді патернів

Проведено аналіз результатів опитування мешканців ряду житлових комплексів з метою їх кластеризації з використанням демографічних, психографічних та поведінкових критеріїв. Використовуючи методи персон та випадкового лісу, сформовано класи протоперсон, які представляють відповідні ролі в інформаційній системі безпеки житлового комплексу.

Досліджено основні аспекти та інструментарії методології DevOps та їхню застосовність при розробці інтелектуальної інформаційної системи безпеки житлового комплексу та сформовано доцільні групи інструментів, які використано при розробленні систем.

Розроблено архітектурну модель інформаційної системи безпеки, побудовано UML діаграми, які відображають функціональну складову системи, проаналізовано і використано сучасні інформаційні технології відеоспостереження, цифрової ідентифікації та задач розпізнавання.

Реалізовано підсистеми контролю доступу, відеоспостереження, управління послугами та інтерфейси для користувачів і адміністраторів інформаційної системи безпеки житлового комплексу «АСТРА. Безпечний ЖК», яка враховує потреби громади в безпеці житлового комплексу, об'єднує в єдину інформаційну еко-систему IoT рішення безпеки ЖК (відеоспостереження, контроль доступу, інформаційні послуги), є незалежною від конкретних виробників давачів, враховує вітчизняні реалії сьогодення і проходить етап тестових випробувань.

Імплементація інформаційної системи безпеки ЖК на функціональному полі інтернет-провайдера, додало їй наступних переваг: використання кабельної інфраструктури провайдера на території ЖК для швидкого і раціонального розгортання, використання серверної інфраструктури провайдера, інструментарію методології DevOps, для моніторингу, інтеграції, захисту від кібератак та оперативного масштабування.

База знань інформаційної системи формується із врахуванням вимог подальшої інтеграції та розбудови функцій ситуаційної обізнаності в наступних версіях програмного комплексу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Cook D. J., Augusto J. C., Jakkula V. R. Ambient Intelligence: Technologies, Applications, and Opportunities. *Pervasive and Mobile Computing.* 2009. Т. 5, № 4. С. 277–298. DOI: 10.1016/j.pmcj.2009.04.001.

2. Cisco Annual Internet Report (2018–2023). https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

3. Parks Associates Research at CES 2024: https://www.parksassociates.com/blogs/press-releases/at-ces-2024-parks-associates-announces-new-research-showing-average-number-of-connected-devices-per-us-internet-household-reached-17-in-2023

4. Techjury Report on IoT Devices in 2024 https://techjury.net/blog/how-many-iot-devices-are-there/

5. Burzagli, Laura, Pier Luigi Emiliani, Margherita Antona, and Constantine Stephanidis. “Intelligent Environments for All: A Path towards Technology-Enhanced Human Well-Being.” Universal Access in the Information Society, 2022, 1–20

6. IDEO's Human-Centered Design Toolkit https://www.ideo.com/journal/design-kit-the-human-centered-design-toolkit

7. Universal Design Principles. (https://www.udll.com/resources/principles

8. Human-Centric IoT. (https://www.mdpi.com/1999-5903/11/5/105)

9. Pervasive Computing in Smart Homes. (https://www.springer.com/gp/book/9783319047329)

10. Privacy in Smart Homes. (https://dl.acm.org/doi/10.1145/2660216.2660219)

11. Intelligent Environments: A Manifesto / Augusto J. C., Callaghan V., Cook D., Kameas A., Satoh I. *Human-Centric Computing and Information Sciences*. 2013. Т. 3. С. 1–18. DOI: 10.1186/2192-1962-3-12

12. Cook D. J., Augusto J. C., Jakkula V. R. Ambient Intelligence: Technologies, Applications, and Opportunities. *Pervasive and Mobile Computing*. 2009. Т. 5, № 4. С. 277–298. DOI: 10.1016/j.pmcj.2009.04.001

13. Malekshahi Rad, Mozhgan Amir, Masoud Rahmani, Amir Sahafi, Nooruldeen Nasih Qader. Social Internet of Things: Vision, Challenges, and Trends. *Human-Centric Computing and Information Sciences*. 2020. 10, №. 1 P.52. https://doi.org/10.1186/s13673-020-00254-6

14. Taiwo O., Gabralla L. A., Ezugwu A. E. Smart Home Automation: Taxonomy, Composition, Challenges and Future Direction. *Springer*. 2020. С. 878–894

15. Fanshawe D. G. J. Architectures for Home Systems. *IET*. 1990. С. 3–1.

16. What is information system architecture. https://www.architecturemaker.com/what-is-information-system-architecture/

17. Juan A.C. , Callaghan V., Cook C., Kameas A., Satoh I. Intelligent Environments: A Manifesto. Human-Centric Computing and Information Sciences, 2013, Vol.3, P. 1–18. https://doi.org/10.1186/2192-1962-3-12

18. Samah Hassan, Eassa Ahmed. A Proposed Architecture for Smart Home Systems Based on IoT, Context-Awareness and Cloud Computing. *International Journal of Advanced Computer Science and Applications/* 2022. Vol.13, № 6. https://doi.org/10.14569/IJACSA.2022.0130612

19. Vilas Donode Pooja, Khade Shalu Malhari, Patil Mayuri Suresh, Sarode Sonali Pramod, Rahul Kadam. Intelligent Home Systems for Ubiquitous User Support by Using Rule Based Approach. *JETIR*. 2021 Volume 8, Issue 5. P.273-276.

20. Eirini K., Warriach E., Lazovik A., Aiello M. Coordinating the Web of Services for a Smart Home. *ACM Transactions on the Web*, 2013. Vol. 7, № 2. P. 1–40. https://doi.org/10.1145/2460383.2460389

21. Yang K., Cho S.-B. Towards Sustainable Smart Homes by a Hierarchical Hybrid Architecture of an Intelligent Agent. *Sustainability*. 2016. Т. 8. Вип. 10. С. 1020. DOI: 10.3390/su8101020

22. Hamilton A. C., Grafton S. T. Goal representation in human anterior intraparietal sulcus. *Journal of Neuroscience*. 2006. Т. 26. С. 1133–1137.

23. Hamilton A. C., Grafton S. T. Action outcomes are represented in human inferior frontoparietal cortex. *Cerebral Cortex*. 2008. Т. 18. С. 1160–1168.

24. Implementing Intelligent Technical Systems into Smart Homes by Using Model Based Systems Engineering and Multi-Agent Systems / Michael J., Hillebrand M., Wohlers B., Henke C., Dumitrescu R., Meyer M., Trächtler A. *Proceedings of the 14th International Conference on Renewable Energy and Power Quality (ICREPQ'16)*. Madrid, 2016. C. 320–325. DOI: 10.24084/repqj14.320

25. Mocrii D., Chen Y., Musilek P. IoT-Based Smart Homes: A Review of System Architecture, Software, Communications, Privacy and Security. *Internet of Things*. 2018. T. 1. C. 81–98. DOI: 10.1016/j.iot.2018.08.009

26. Vijayaraja L., Jayakumar N. S., Dhanasekar R., Manibha M. P., Vignesh V., Kesavan R. Sustainable Smart Homes Using IoT for Future Smart Cities, 4th International Conference on Smart Electronics and Communication (ICOSEC*)*. 2023 P.365-369. doi: 10.1109/ICOSEC58147.2023.10276371.

27. Saurabh S., Ra In-Ho, Meng W., Kaur M., Cho G. H. SH-BlockCC: A Secure and Efficient Internet of Things Smart Home Architecture Based on Cloud Computing and Blockchain Technology. *International Journal of Distributed Sensor Networks.* 2019 Vol.15, № 4. P. 1550147719844159. https://doi.org/10.1177/1550147719844159

28. Hassan S. A. Z., Eassa A. M. A Proposed Architecture for Smart Home Systems Based on IoT, Context-Awareness and Cloud Computing. *International Journal of Advanced Computer Science and Applications*. 2022. T. 13. Вип. 6. DOI: 10.14569/IJACSA.2022.0130612

29. Laws of software. https://www.laws-of-software.com/laws/gall/

30. Stolojescu-Crisan C., Crisan C., Butunoi B.-P. An IoT-Based Smart Home Automation System. *Sensors*. 2021. T. 21. Вип. 11. C. 3784. DOI: 10.3390/s21113784

31. Physical Layer Secure Communications Based on Collaborative Beamforming for UAV Networks: A Multi-Objective Optimization Approach / Li J., Kang H., Sun G., Liang S., Liu Y., Zhang Y. *Proceedings of the IEEE International Conference on Communications (ICC)*. IEEE, 2021. C. 1–10.

32. Coordinating the Web of Services for a Smart Home / Kaldeli E., Warriach E. U., Lazovik A., Aiello M. *ACM Transactions on the Web (TWEB)*. 2013. Т. 7. Вип. 2. С. 1–40. DOI: 10.1145/2460383.2460389.

33. Risteska Stojkoska B. L., Trivodaliev K. V. A Review of Internet of Things for Smart Home: Challenges and Solutions. *Journal of Cleaner Production*. 2017. Т. 140. С. 1454–1464.

34. IoT-Cloud Service Optimization in Next Generation Smart Environments / Barcelo M., Correa A., Llorca J., Casademont J., Vicario J. L., Morell A., López-Vicario J. *IEEE Journal on Selected Areas in Communications*. 2016. Т. 34. Вип. 12. С. 4077–4090. DOI: 10.1109/JSAC.2016.2621378

35. Youngblood G. M., Cook D. J. Data Mining for Hierarchical Model Creation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 2007. Т. 37. Вип. 4. С. 561–572. DOI: 10.1109/TSMCC.2007.897565.

36. Chen L. M., Nugent C. D., Wang H. A Knowledge-Driven Approach to Activity Recognition in Smart Homes. *IEEE Transactions on Knowledge and Data Engineering*. 2012. Т. 24. Вип. 6. С. 961–974. DOI: 10.1109/TKDE.2011.48

37. A review of smart homes—present state and future challenges / Chan M., Esteve D., Escriba C., Campo E.. *Computer Methods and Programs in Biomedicine*. 2008. Т. 91. Вип. 1. С. 55–81. DOI: 10.1016/j.cmpb.2008.02.001

38. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 3rd ed. Upper Saddle River, NJ: Prentice Hall Press, 2009. 1152 с. ISBN: 978-0136042594

39. Yang Y., Xu C., Shi H. Smart Contract-Based Distributed Access Control for Smart Home. *Proceedings of the 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*. 2022. С. 454–460. DOI: 10.1109/ICCECE54139.2022.9712746.

40. Intelligent Access Control System / Miaolan Z., Nuo C., Yang Y., Pan C., Zhou X. *Proceedings of the 7th International Conference on Communications, Signal Processing and Systems (CCISP)*. 2022. С. 1–6. DOI: 10.1109/CCISP55629.2022.9974224

41. Cimorelli Belfiore R., Ferrara A. L. Security Analysis of Access Control Policies for Smart Homes. *Proceedings of the International Conference on Cybersecurity and Resilience (ICCR)*. 2023. С. 99–106.

42. Cimorelli Belfiore R., Ferrara A. L. Security Analysis of Access Control Policies for Smart Homes. *Proceedings of the International Conference on Cybersecurity and Resilience (ICCR)*. 2023. С. 99–106.

43. Lee S.-H., Yang C.-S. An Intelligent Home Access Control System Using Deep Neural Network. *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*. 2017. С. 281–282.

44. Yunhui Y., Xu C., Shi H. Smart Contract-Based Distributed Access Control for Smart Home. *2nd International Conference on Consumer Electronics and Computer Engineering* (ICCECE). 2022. P.454–60. https://doi.org/10.1109/ICCECE54139.2022.9712746

45. Miaolan Z., Nuo C., Yang Y., Chenlu P., Xiangjie Z. Intelligent Access Control System. 2022. P. 423-427. 10.1109/CCISP55629.2022.9974224...

46. Krizhevsky A., Sutskever I., Hinton G. ImageNet classification with deep convolutional neural networks. Commun. *ACM.* 2017. Vol.60, №6. P.84–90. https://doi.org/10.1145/3065386

47. Gruber T. R. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*. 1993. Т. 5. Вип. 2. С. 199–220. DOI: 10.1006/knac.1993.1008.

48. Neuhaus F. What Is an Ontology? *ArXiv*. 22 жовтня 2018. URL: https://arxiv.org/abs/1810.08764 (дата звернення: [10.10.2024])

49. Gruber T. Ontology. *Encyclopedia of Database Systems*. Springer, 2009. С. 1963–1965.

50. Ganter B., Wille R. *Formal Concept Analysis*. Springer Berlin Heidelberg, 2013. 284 с.

51. Hofstadter D. Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought. New York: Basic Books, 1995. 528 с.

52. Contextualizing Ontologies / Bouquet P., Giunchiglia F., Van Harmelen F., Serafini L., Stuckenschmidt H. *Journal of Web Semantics*. 2004. Т. 1. Вип. 4. С. 325–343. DOI: 10.1016/j.websem.2004.07.001.

53. Contextual Ontologies / Benslimane D., Arara A., Falquet G., Maamar Z., Thiran P. *Springer*. 2006. С. 168–176.

54. Burov Y., Karpov I. Contextual Concept Meaning Alignment Based on Prototype Theory. *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Systems (COLINS)*. 2023. С. 137–146.

55. C-Owl: Contextualizing Ontologies / Bouquet P., Giunchiglia F., Van Harmelen F., Serafini L., Stuckenschmidt H. *Springer*. 2003. С. 164–179.

56. Жовнір Ю.І., Цейтлін Г.Е., Захарія Л.М., Захарія О.В. Екологічні аспекти подання знань засобами алгебри алгоритміки. *Проблеми програмування*. 2010. № 2–3: Спеціальний випуск. С.369-375

57. Жовнір Ю., Захарія Л., Захарія Ю. Формалізація та породження знань засобами алгебри алгоритміки. *Комп'ютерні науки та інженерія*: матеріали наукової конференції молодих вчених, Львів, 25-27 листопада 2010 р., Львів, 2010. С. 118-120

58. Cyc's knowledge base, https://www.cyc.com/archives/service/cyc-knowledge-base, last accessed: 2024/10/24

59. Wordnet. A lexical database for English, https://wordnet.princeton.edu/, last accessed: 2024/10/24

60. Guarino N. Formal Ontology in Information Systems. *Proceedings of the First International Conference (FOIS'98), June 6–8, Trento, Italy*. Trento: IOS Press, 1998. Т. 46.

61. Guizzardi G., Wagner G. Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages. *Theory and Applications of Ontology: Computer Applications*. Springer, Dordrecht, 2010. С. 175–196.

62. Herre H. General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling. *Theory and Applications of Ontology: Computer Applications*. Springer, Dordrecht, 2010. С. 297–345.

63. The wonderweb library of foundational ontologies / Masolo C., Borgo S., Gangemi A., Guarino N., Oltramari A., Schneider L. *Technical report*, ISTC-CNR 2002. URL: http://wonderweb.semanticweb.org

64. Wohed P. Conceptual Patterns for Reuse in Information Systems Analysis. *Proceedings of the 12th International Conference on Advanced Information Systems Engineering (CAiSE 2000), Stockholm, Sweden, June 5–9*. Springer, 2000. С. 157–175

65. Gangemi A., Presutti V. Ontology Design Patterns. *Handbook on Ontologies*. Springer, 2009. С. 221–243.

66. de Almeida Falbo R., Barcellos M. P., Nardi J. C., Guizzardi G. Organizing Ontology Design Patterns as Ontology Pattern Languages. *Proceedings of the 10th International Conference on The Semantic Web: Semantics and Big Data (ESWC 2013), Montpellier, France, May 26–30*. Springer, 2013. С. 61–75.

67. Souza E., Falbo R., Vijaykumar N. L. Using Ontology Patterns for Building a Reference Software Testing Ontology. *Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW 2013)*. IEEE, 2013. С. 21–30.

68. Zambon E., Guizzardi G. Formal Definition of a General Ontology Pattern Language Using a Graph Grammar. *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2017. С. 1–10.

69. Hirt Q., Shimizu C., Hitzler P. Extensions to the Ontology Design Pattern Representation Language. *Proceedings of the Workshop on Ontology Design and Patterns (WOP@ISWC)*. 2019. URL: https://www.semanticscholar.org/paper/Extensions-to-the-Ontology-Design-Pattern-Language-Hirt-Shimizu/68392a85faf15d91310c3e21daef14a706bb4099 (дата звернення: 20.07.2024)

70. Krieg-Brückner B., Mossakowski T., Codescu M. Generic Ontology Design Patterns: Roles and Change Over Time. *Advances in Pattern-Based Ontology Engineering*. Amsterdam: IOS Press, 2021. С. 25–47.

71. Zagorulko Y. A., Borovikova O. I. Using a System of Heterogeneous Ontology Design Patterns to Develop Ontologies of Scientific Subject Domains. *Programming and Computer Software*. 2020. Т. 46. С. 273–280. DOI: 10.1134/S0361768820040064.

72. Жовнір Ю., Буров Є. Еволюція архітектурних рішень для розумних будинків. *Computer Systems and Information Technologies*, 2024, Вип.3, С.74–85. https://doi.org/10.31891/csit-2024-3-10

73. Towards an Ontology for Technical Security Standards / Illescas J., Ehrlinger L., Denk G., Buchgeher G. *Proceedings of the 28th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2023)*. IEEE, 2023. С. 1–8.

74. Shoaib Farooq M., Talha Waseem M. Developing and Building Ontologies in Cyber Security. *arXiv preprint arXiv:2306.00377*. 2023. URL: https://arxiv.org/abs/2306.00377 (дата звернення: [20.07.2024]).

75. Preuveneers D., Joosen W. An Ontology-Based Cybersecurity Framework for AI-Enabled Systems and Applications. *Future Internet*. 2024. Т. 16. Вип. 3. С. 69. DOI: 10.3390/fi16030069.

76. Babayeva G., Maennel K., Maennel O. M. Building an Ontology for Cyber Defence Exercises. *Proceedings of the IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2022. С. 423–432.

77. Fenton, N. E., Pfleeger S. L. Software Metrics: A Rigorous and Practical Approach. 2nd Edition. Berlin: International Thompson Computer Press, 1997. 195 с.

78. Wilson, R. J. Introduction to Graph Theory. Longman, 2010. – 201 с.

79. Naylor C. Build your own PC expert system. London: Sigma Press, 1983. – 167 с.

80. Tesler H. S. Metrics and norms in the hierarchy of categorical semantics and functions. *Mathematical Machines and Systems*. 2005. №2. P. 65-68.

81. Velichko V. Yu. Solving analytical problems in discrete environments by methods of derivation by analogy: Ph.D. thesis. Kyiv: Institute of Cybernetics, 2004. P.112-114.

82. Lytvyn V. V. Knowledge bases of intelligent decision-making support systems. Lviv: Lviv Polytechnic Publishing House, 2011. P.23-25.

83. Dosyn D. G., Lytvyn V. V., Nikolskyi Yu. V., Pasichnik V. V. Intelligent systems based on ontologies. Lviv: Civilization, 2009. P.123-126.

84. Lytvyn V. V. A method of entering a metric to determine the distance between text documents. *Visnyk of the National Lviv Polytechnic University. Information Systems and Networks*. 2008. №621. P. 162–171.

85. Жовнір Ю., Грибовський О. Порівняльний аналіз програмно-апаратних інструментальних засобів для створення безпекової системи багатоквартирного будинку. *Herald of Khmelnytskyi National University. Technical Sciences*. 2024. Вип.*339*(4). С.344-358. https://doi.org/10.31891/2307-5732-2024-339-4-54

86. Методологія розроблення та супроводу інформаційних систем, базованих на технології інтернету речей / Жовнір Ю. І., Грибовський О. М., Орлов М. В., Дуда О. М., Кунанець Н. Е. *Управління розвитком складних систем* 2024. Вип.60. С. 56-71. https://doi.org/10.32347/2412-9933.2024.60.56-70

87. Жовнір Ю. І., Кунанець Н. Е., Захарія О. В., Орлов М. В. Використання методологій devops та devsecops у IT проєктах. Proceedings *the 4th International scientific and practical conference "Science in the modern world: innovations and challenges"* (December 19-21, 2024) Toronto, Canada. Toronto: Perfect Publishing, 2024. P.228-233. URL: https://sci-conf.com.ua/iv-mizhnarodna-naukovo-praktichna-konferentsiya-sciencein-the-modern-world-innovations-and-challenges-19-21-12-2024-toronto-kanada-arhiv/.

88. Орлов М. В., Жовнір Ю.І., Грибовський О.М., Дуда О.М. Від концепції до реальності: роль DevOps в екосистемах IoT *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2024. Том 35 (74), № 6, Ч.2. С.163-171.

89. Vaskiv R., Veretennikova N., Nebesnyi R., Bilovus H., Zhovnir Y. Formation of an IT Project Team by Analogy with a Flock *IEEE 19th International Conference on Computer Science and Information Technologies* (CSIT). 2024.

90. Pereira I. M., de Senna Carneiro T. G., Figueiredo E. Understanding the Context of IoT Software Systems in DevOps. *arXiv*. 2021. doi: 10.48550/arXiv.2104.10147.

91. Maayan G. D. A DevOps Guide to IoT Technology. URL: https://devops.com/a-devops-guide-to-iot-technology (дата звернення: 08.11.2024).

92. Apprecode. DevOps in the Creative Industries: Streamlining Content Creation Workflows. URL: https://apprecode.com/blog/devops-in-iot-accelerating-innovation-in-the-internet-of-things (дата звернення: 08.11.2024).

93. Michalowski M. Using DevOps Practices to Enhance IoT Security. URL: https://www.iotforall.com/using-devops-practices-to-enhance-iot-security (дата звернення: 08.11.2024).

94. Softprom. AWS CloudTrail: Monitoring and Logging. New Demo! URL: https://softprom.com/ua/aws-cloudtrail-monitoring-ta-vedennya-jurnaliv-loguvannya-nove-demo (дата звернення: 08.11.2024).

95. Орлов М. В., Дуда О. М., Жовнір Ю. І., Грибовський О. М. Інструменти методології DevOps в інформаційних системах на основі технологій IoT *Комп'ютерно-інтегровані технології: освіта, наука, виробництво.* 2024. Вип. 57. С.128-139. https://doi.org/10.36910/6775-2524-0560-2024-57-15

96. Жовнір Ю. І., Кунанець Н. Е., Захарія О. В. Вимоги до інформаційних систем безпеки житлового кварталу. Proceedings *the 5th International scientific and practical conference "Current trends in scientific research development" (December 12-14, 2024)*. Boston, USA. Boston: BoScience Publisher, 2024. P. 298–303. URL: https://sci-conf.com.ua/v-mizhnarodna-naukovo-praktichna-konferentsiya-current-trendsin-scientific-research-development-12-14-12-2024-boston-ssha-arhiv/.

97. Wang M., Han C. The Design of Intelligent Residence Property Management Information System (IPMIS) Based on E-Business. *Proceedings of the*

CRIOCM 2006 International Symposium on Advancement of Construction Management and Real Estate. 2006. С. 1–9.

98. Cimorelli Belfiore R., Ferrara A. L. Security Analysis of Access Control Policies for Smart Homes. *Proceedings of the 28th ACM Symposium on Access Control Models and Technologies (SACMAT 2023)*. 2023. С. 99–106.

99. Pech M., Vrchota J., Bednář J. Predictive Maintenance and Intelligent Sensors in Smart Factory. *Sensors*. 2021. Т. 21. Вип. 4. С. 1470. DOI: 10.3390/s21041470.

100. UFO community portal. https://ontouml.org/ufo/

101. Herre H. General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling. *Theory and Applications of Ontology: Computer Applications*. Dordrecht: Springer, 2010. С. 297–345.

102. Loebe F., Burek P., Herre H. GFO: The General Formal Ontology. *Applied Ontology*. 2022. Т. 17. Вип. 1. С. 71–106. DOI: 10.3233/AO-220264

103. Niles I., Pease A. Towards a Standard Upper Ontology. *Proceedings of the Second International Conference on Formal Ontology in Information Systems (FOIS 2001)*. New York: ACM Press, 2001. С. 2–9.

104. Burov Y., Zhovnir Y., Zakharia O. Designing the ontology for intelligent security system of residential community *Scientific journal of the Ternopil Ivan Puluj National Technical University,* 2024, vol 116, no 4, P. 111-124. https://doi.org/10.33108/visnyk_tntu2024.04.111.

105. Burek P., Loebe F., Herre H. Towards GFO 2.0: Architecture, Modules and Applications. *Proceedings of the International Conference on Ontologies and Semantic Technologies*. Amsterdam: IOS Press, 2020. С. 32–45.

106. Degen W., Heller B., Herre H., Smith B. GOL: A General Ontological Language. *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS 2001)*. October 17–19, 2001, Ogunquit, Maine, USA, Ogunquit: ACM Press, 2001. С. 34–46.

107. Baumann, Ringo, Frank Loebe, and Heinrich Herre. Towards an Ontology of Space for GFO. Amsterdam: IOS Press, 2016. P. 53–66.

108. Жовнір Ю. І., Кунанець Н. Е., Захарія О. В., Орлов М. В. Планування та прогнозування ситуації в інтелектуальній інформаційній системі безпеки. Proceedings *II International scientific and practical conference «Future of science: innovations and perspectives»*, (December 23-25, 2024). Stockholm, Sweden. Stockholm: Perfect Publishing, 2024. С.163-169. https://sci-conf.com.ua/wp-content/uploads/2024/12/FUTURE-OF-SCIENCE-INNOVATIONS-AND-PERSPECTIVES-23-25.12.24.pdf.

109. Baumann R., Loebe F., Herre H. Towards an Ontology of Space for GFO. *Proceedings of the International Conference on Ontologies and Semantic Technologies*. Amsterdam: IOS Press, 2016. С. 53–66.

110. Burov Y. Goal-Driven Situation Awareness Process Based on Predictive Modeling. *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2024), April 12–13, Lviv, Ukraine*. 2024. Т. 2. С. 157–168.

111. Kelly D. Determining Factors That Affect Long-Term Evolution in Scientific Application Software. *Journal of Systems and Software*. 2009. Т. 82. Вип. 5. С. 851–861. DOI: 10.1016/j.jss.2008.11.846.

112. Khorikov V. Short-Term vs Long-Term Perspective in Software Development. *Enterprise Craftsmanship*. URL: https://enterprisecraftsmanship.com/posts/short-term-vs-long-term-perspective/ (дата звернення: 19.09.2024).

113. Towards an Ontology for Technical Security Standards / Illescas J., Ehrlinger L., Denk G., Buchgeher G. *Proceedings of the 28th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2023)*. IEEE, 2023. С. 1–8.

114. Shoaib Farooq M., Talha Waseem M. Developing and Building Ontologies in Cyber Security. *arXiv*. 2023. arXiv-2306. URL: https://arxiv.org/abs/2306.00377 (дата звернення: [10.09.2024]).

115. Preuveneers D., Joosen W. An Ontology-Based Cybersecurity Framework for AI-Enabled Systems and Applications. *Future Internet*. 2024. Т. 16. Вип. 3. С. 69. DOI: 10.3390/fi16030069.

116. Loebe F., Burek P., Herre H. GFO: The General Formal Ontology. *Applied Ontology*. 2022. Т. 17. Вип. 1. С. 71–106. doi: 10.3233/AO-220264

117. Григорович А., Григорович В., Жовнір Ю., Грибовський О. Формування обернено-адитивної семантичної метрики для аналізу онтологій безпекових систем багатоквартирних будинків. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2024. Вип. 56. С.12-30. https://doi.org/10.36910/6775-2524-0560-2024-56-02

118. Fenton N. E., Pfleeger S. L. Software Metrics: A Rigorous and Practical Approach. 2nd ed. Berlin: International Thompson Computer Press, 1997. 195 с.

119. Екологічні аспекти подання знань засобами алгебри алгоритміки / Г.Е. Цейтлін, Л.М. Захарія, О.В. Захарія, Ю.І. Жовнір. *Проблеми програмування*. 2010. №2-3: Спеціальний випуск. С.369-375.

120. Zhovnir Y., Kunanets N., Burov Y., Duda O., Pasichnyk V. Development of the structure and architecture of situational awareness security information systems for residential complexes *Eastern-European Journal of Enterprise Technologies,* 2025, №1(133)

121. Gall J. *Systemantics: How Systems Really Work and How They Fail*. Quadrangle New York: Times Book Co., 1975. 111 с.

122. *Agile Manifesto*. URL: https://agilemanifesto.org (дата звернення: [10.09.2024]).

123. Lean Start-Up as a Strategy for the Development and Management of Dynamic Entrepreneurships / Villalobos Rodríguez G., Vargas Montero M., Rodriguez Ramirez J., Araya-Castillo L. A. *Dimensión Empresarial*. 2018. Т. 16. Вип. 2. С. 193–208

124. Кунанець Н., Жовнір Ю., Веремеєнко А., Пущак С. Концептуальне моделювання системи відеоспостереження з ситуаційною обізнаністю *Herald of Khmelnytskyi National University. Technical Sciences.* 2025. №1. С.189-202.

125. Burov E., Zhovnir Y., Zakhariya O. The Vision and Implementation of Intelligent Security System. *Herald of Khmelnytskyi National University. Technical Sciences*. 2024. Vol. 341, No. 5. P. 497–509. https://doi.org/10.31891/2307-5732-2024-341-5-72

126. Vladov S., Avkurova Zh., Lytvyn V., Zhovnir Yu. Analytical Neural Network System for the Helicopter Turboshaft Engines Operating Modes Classification. *International Journal of Computing*. 2024. Vol. 23, No. 3. P. 342–359. https://doi.org/10.31891/csit-2024-3-10

127. Жовнір Ю. І., Кунанець Н. Е.,Захарія О. В., Пасічник С. О. Формування бекенду та фронтенду інформаційної системи безпеки з ситуаційною обізнаністю Proceedings I International scientific and practical conference «European congress of scientific discovery», (December 29-31, 2024). Madrid. Spain. Madrid: Perfect Publishing, 2024, С.244-252. https://sci-conf.com.ua/wp-content/uploads/2024/12/EUROPEAN-CONGRESS-OF-SCIENTIFIC-DISCOVERY-29-31.12.2024.pdf

128. Жовнір Ю., Грибовський О., Пасічник С., Бобик І. Створення інтерфейсів безпекових систем багатоквартирних будинків з використанням методу Персон *Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі*, 2024, Випуск 16, С. 145 – 166.- https://doi.org/10.23939/sisn2024.16.145

129. Жовнір Ю., Ваків М., Захарія Л. Віртуальна аудиторія як система електронного навчання інвалідів *Комп'ютерні науки та інженерія*: матеріали наукової конференції молодих вчених, 25-27 листопада 2010 р., Львів. Львів, 2010. С.126-128

Додатки

Додаток А Акти про впровадження результатів дисертаційної роботи

АКТ
ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ ЖОВНІРА ЮРІЯ ІВАНОВИЧА
ЗА ТЕМОЮ «МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНОЇ ПРОГРАМНОЇ СИСТЕМИ БЕЗПЕКИ ЖИТЛОВИХ КОМПЛЕКСІВ»

м. Чернівці
«05» 12 2024 р.

Ми, що нижче підписалися, представники кафедри комп'ютерних систем і технологій ПВНЗ «Буковинський університет» у складі:

- Завідувача кафедри Артеменко О.І.
- Професора кафедри Зайця В.М.
- Доцента кафедри Гаця Б.М.

склали цей акт про те, що результати дисертаційного дослідження Жовніра Юрія Івановича впроваджено в навчальний процес кафедри.

Розроблені дисертантом алгоритми аналізу загроз та управління доступом у інтелектуальних програмних системах використані у лекційних матеріалах дисциплін «Технології захисту інформації», «Інтелектуальний аналіз даних» та «Прикладні аспекти систем штучного інтелекту».

Отримані результати включені до методичних рекомендацій для проведення практичних та лабораторних занять. У лабораторних роботах застосовано розроблені дисертантом моделі IoT-мереж для управління безпекою житлових комплексів, практичні завдання з аналізу поведінкових патернів та прогнозування загроз.

Впровадження результатів дослідження сприяє підвищенню рівня підготовки студентів, формуванню практичних навичок у сфері побудови інтелектуальних програмних систем безпеки.

Завідувач кафедри комп'ютерних систем і технологій ПВНЗ «Буковинський університет», к.т.н., доцент — Ольга АРТЕМЕНКО

Професор кафедри комп'ютерних систем і технологій ПВНЗ «Буковинський університет», д.т.н., професор — Василь ЗАЯЦЬ

Доцент кафедри комп'ютерних систем і технологій ПВНЗ «Буковинський університет», к.т.н., доцент — Богдан ГАЦЬ

АКТ
ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ ЖОВНІРА ЮРІЯ ІВАНОВИЧА
ЗА ТЕМОЮ «МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНОЇ ПРОГРАМНОЇ СИСТЕМИ БЕЗПЕКИ ЖИТЛОВИХ КОМПЛЕКСІВ»

м. Луцьк
«27» 11 2024 р.

Ми, що нижче підписалися, представники факультету інформаційних технологій і математики Волинського національного університету імені Лесі Українки у складі:

- Декана факультету, к.пед.н, доцента Світлани ЯЦЮК
- Заступника декана, доктора філософії з інформаційних систем та технологій Валентини ЮНЧИК
- Завідувача кафедри комп'ютерних наук та кібербезпеки, к.ф.-м.н., доцента Тетяни ГРИШАНОВИЧ

склали цей акт про те, що результати дисертаційного дослідження на тему: «Методи та засоби побудови інтелектуальної програмної системи безпеки житлових комплексів» Жовніра Юрія Івановича впроваджено в навчальний процес факультету.

Результати дослідження були впроваджені в освітній процес факультету інформаційних технологій і математики Волинського національного університету імені Лесі Українки зокрема шляхом використання розроблених дисертантом алгоритмів виявлення загроз та управління доступом в інтелектуальних інформаційних системах безпеки житлових комплексів. Зазначені результати використовуються при викладанні навчальних дисциплін:

Паралельні та розподілені обчислення

Нечіткі моделі та методи аналізу даних

Безпека інформаційно-комунікаційних систем

Інтелектуальний аналіз даних

Вони також включені до методичних рекомендацій для проведення практичних і лабораторних занять. Розроблені моделі використовуються в навчальних кейсах для моделювання роботи систем безпеки на основі IoT. У

рамках лабораторних робіт реалізовано моделі розподіленого опрацювання даних та розпізнавання загроз. Прототип інтелектуальної системи безпеки використовується для практичного моделювання роботи систем контролю доступу.

Впровадження цих результатів сприяє підвищенню рівня підготовки студентів, розвитку практичних навичок у сфері математичного та програмного забезпечення інтелектуальних систем.

Декан факультету інформаційних технологій
і математики Волинського
національного університету
імені Лесі Українки,
к. пед.н, доцент **Світлана ЯЦЮК**

Заступник декана факультету інформаційних
технологій і математики Волинського
національного університету імені Лесі Українки,
доктор філософії з інформаційних систем
та технологій **Валентина ЮНЧИК**

Завідувач кафедри комп'ютерних наук
та кібербезпеки Волинського
національного університету імені Лесі Українки,
к.ф.-м.н., доцент **Тетяна ГРИШАНОВИЧ**

"ЗАТВЕРДЖУЮ"
Проректор з науково-педагогічної роботи та інформатизації Національного університету "Львівська політехніка"
_____ д.т.н., професор Павло ЖЕЖНИЧ

АКТ
про використання результатів дисертації аспіранта кафедри інформаційних систем та мереж Жовніра Юрія Івановича "Методи та засоби побудови інтелектуальної програмної системи безпеки житлових комплексів", поданої на здобуття наукового ступеня кандидата технічних наук за спеціальністю 01.05.03 «Математичне та програмне забезпечення обчислювальних машин та систем» у навчальному процесі кафедри інформаційних систем та мереж Національного університету "Львівська політехніка"

Ми, що нижче підписалися, голова Науково -методичної ради Інституту комп'ютерних наук та інформаційних технологій, к.т.н., доцент Шестакевич Т.В., завідувач кафедри інформаційних систем та мереж, д.т.н., професор Литвин В.В.; заступник завідувача кафедри інформаційних систем та мереж, гарант ОПП «Інформаційні системи та технології» другого (магістерського) рівня вищої освіти, д.н.с.к., професор Кунанець Н.Е., гарант ОНП «Інформаційні системи та технології» третього (PhD) рівня вищої освіти, д.т.н., професор Буров Є. В. цим актом підтверджуємо, що результати дисертаційних дослідження аспіранта кафедри інформаційних систем та мереж Жовніра Ю.І. використано у навчальному процесі кафедри інформаційних систем та мереж Національного університету «Львівська політехніка» в частині:

- методики вибору програмно-алгоритмічних засобів IT платформ на основі методу аналізу ієрархій Сааті для побудови інтелектуальних інформаційних систем, базованих на технологіях інтернету речей та методологіях проєктування і супроводу програмних продуктів DevOps та DevSecOps;

- методики формування доменно-орієнтованих онтологій на основі концептів базової 4d онтології GFO для інтелектуальних інформаційних систем безпеки з ситуаційною обізнаністю, а також пакетів патернів типових ситуацій для систем безпеки житлових комплексів;

- методики побудови користувацьких інтерфейсів для інтелектуальних інформаційних систем безпеки житлових комплексів на основі інтеграції методів персон та випадкового лісу, що дозволяє формувати інтерфейси для зазначених систем з урахуванням особливих персональних характеристик потенційних користувачів з огляду на їх вікові, професійні, сімейні та інші характеристики.

Зазначені методики імплементовані у матеріали лекцій та цикли лабораторних робіт в межах викладання дисциплін:

«Розподілені бази даних та знань» для студентів другого (магістерського) рівня освіти за освітньо-науковою програмою «Системний аналіз»;

«Хмарні технології» для студентів першого (бакалаврського) рівня освіти за освітньо-професійною програмою «Розподілені інформаційні системи та технології (DevOps & Data Engineering)»;

«Проєктний аналіз» » для студентів другого (магістерського) рівня освіти за освітньо-професійною програмою «Управління IT проєктами»;

«Управління проєктами з розвитку складних систем» для студентів третього (PhD) рівня освіти за освітньо-науковою програмою "Інформаційні системи та технології».

Голова Науково -методичної ради
Інституту комп'ютерних наук та
інформаційних технологій,
к.т.н., доцент Тетяна ШЕСТАКЕВИЧ

Завідувач кафедри інформаційних систем
та мереж, д.т.н., професор Василь ЛИТВИН

Заступник завідувача кафедри
інформаційних систем та мереж,
гарант ОПП «Інформаційні системи
та технології» другого (магістерського) рівня
вищої освіти, д.н.с.к., професор Наталія КУНАНЕЦЬ

Гарант ОНП «Інформаційні системи та
технології» третього (PhD) рівня вищої освіти,
д.т.н., професор Євген БУРОВ

АКТ
впровадження результатів дисертаційного дослідження
Жовніра Юрія Івановича
за темою «Методи та засоби побудови інтелектуальної програмної системи безпеки житлових комплексів», поданої на здобуття наукового ступеня кандидата технічних наук за спеціальністю
01.05.03 «Математичне та програмне забезпечення обчислювальних машин та систем»

м. Тернопіль
«29» 11 2024 р.

Цим актом підтверджується використання результатів дисертаційного дослідження аспіранта Національного університету «Львівська політехніка» Жовніра Юрія Івановича в освітньому процесі та наукових дослідженнях кафедри комп'ютерних наук та науково-дослідної лабораторії «Розумне місто Тернопіль» факультету комп'ютерно-інформаційних систем та програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя, в частині:

- методики проблемно-орієнтованого SWOT аналізу для вибору методології розроблення та супроводу IT проектів інтелектуальних інформаційних систем безпеки житлових комплексів з використанням технологій Інтернету речей;
- методики експертного оцінювання на основі методу аналізу ієрархій Сааті функціональних інструментальних комплексів DevOps та DevSecOps для інтелектуальних програмних систем безпеки з повним покриттям вимог процесів розгортання, масштабування, оновлення та розширення можливостей програмного продукту;
- методики формування проблемно-орієнтованої 4 d онтології з використанням конструктивів GFO для інтелектуальної інформаційної системи безпеки з ситуаційною обізнаністю щодо безпекових прецендентів та розвитку нетипових ситуацій;
- двоетапної методики побудови користувацьких інтерфейсів на основі методу персон та випадкового лісу для інтелектуальних інформаційних систем безпеки житлових комплексів, кварталів, мікрорайонів та територіальних громад з врахуванням специфічних соціокомунікаційних, вікових, психологічних, релігійних та звичаєвих особливостей мешканців.

Зазначені матеріали використовуються в межах викладання дисциплін:

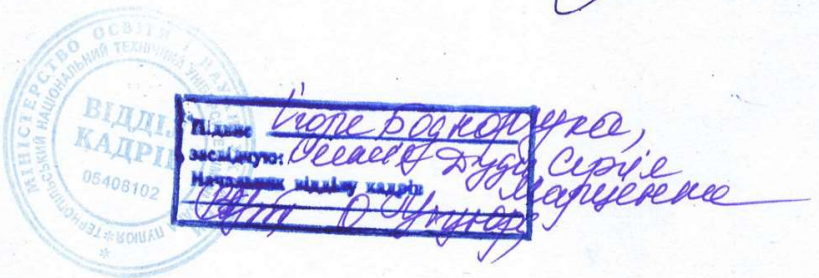
- "Електронні громада, місто та регіон".
- "Інформаційні системи розподіленої та паралельної обробки даних".
- "Інтелектуальні системи аналізу консолідованої інформації".

для студентів другого (магістерського) рівня освіти за освітньо-професійною та освітньо-науковою програмами 122 "Комп'ютерні науки", освітньо-професійними програмами 281 "Публічне управління та адміністрування", 126 "Інформаційні системи та технології», 124 "Системний аналіз".

Завідувач кафедри комп'ютерних наук
Тернопільського національного
технічного університету імені Івана Пулюя,
к.т.н., доцент Ігор БОНДАРЧУК

Керівник науково-дослідної лабораторії
«Розумне місто Тернопіль» Тернопільського
національного технічного університету
імені Івана Пулюя, к.т.н., доцент Олексій ДУДА

Помічник ректора з цифрової трансформації
Тернопільського національного
технічного університету імені Івана Пулюя,
к.т.н., доцент Сергій МАРЦЕНКО



АКТ
впровадження результатів дисертаційного дослідження
Жовніра Юрія Івановича
за темою «Методи та засоби побудови інтелектуальної програмної системи безпеки житлових комплексів», поданої на здобуття наукового ступеня кандидата технічних наук за спеціальністю
01.05.03 «Математичне та програмне забезпечення обчислювальних машин та систем»

Цим актом підтверджується використання результатів дисертаційного дослідження аспіранта Національного університету «Львівська політехніка» Жовніра Юрія Івановича в освітньому процесі та наукових дослідженнях кафедр інформатики та фізико-математичних дисциплін і програмного забезпечення систем факультету інформаційних технологій Державного вищого навчального закладу «Ужгородський національний університет», зокрема:

- методики оцінювання IT-платформ на основі технологій IoT та вибору методології розроблення та супроводу інтелектуальних програмних безпекових систем житлових комплексів;

- методики побудови доменно-орієнтованої онтології на основі базової 4 d онтології GFO, що дозволяє створювати онтології інтелектуальних інформаційних систем з ситуаційною обізнаністю;

- методики формування обернено-адитивної метрики оцінювання відстаней між концептами проблемно-орієнтованих онтологій у процесах їх тиражування та адаптування до швидкозмінних ситуацій у безпекових системах;

- інформаційні матеріали, в яких подано рекомендації з побудови структур та архітектур інтелектуальних інформаційних систем з ситуаційною обізнаністю для житлових комплексів, кварталів та районів у розумних містах, громадах та регіонах.

Зазначені матеріали використовуються при викладанні дисциплін:

- Інформаційні системи та технології в управлінні;
- Проектування баз даних та сховищ даних;
- Інноваційні інформаційні технології

для студентів спеціальностей 121 (F2) «Інженерія програмного забезпечення» та 126 (F6) «Інформаційні системи та технологій» (першого та другого освітніх рівнів вищої освіти).

Водночас результати дисертаційної роботи Жовніра Ю.І. використовуються в наукових дослідженнях аспірантів (третього освітнього-наукового рівня вищої освіти кафедр інформатики та фізико-математичних дисциплін і програмного забезпечення систем), а також в навчально-науковій лабораторії «Розумний регіон «Центр Європи»» кафедри інформатики та фізико-математичних дисциплін ДВНЗ «Ужгородський національний університет».

Декан факультету інформаційних технологій
ДВНЗ «Ужгородський національний університет»,
д.т.н., професор Ігор ПОВХАН

Завідувач кафедри інформатики та фізико-математичних дисциплін ДВНЗ «Ужгородський національний університет»,
к.т.н., доцент Василь КУТ

Завідувач кафедри програмного забезпечення систем
ДВНЗ «Ужгородський національний університет»,
к.ф.-м. н., доцент Юрій БІЛАК

м. Ужгород

«25» 11 2024 р.

АКТ
впровадження в компанії «АСТРА-ЛЬВІВ» результатів дисертаційної роботи Жовніра Юрія Івановича за темою «Методи та засоби побудови інтелектуальних програмних систем безпеки житлових комплексів», поданої на здобуття ступеня кандидата технічних наук за спеціальністю 01.05.03 – математичне та програмне забезпечення обчислювальних машин та систем

Даним актом підтверджуємо, що компанії «АСТРА-ЛЬВІВ» Жовніром Юрієм Івановичем передано для впровадження та подальшого використання наступні результати його дисертаційних досліджень:

1. Методику вибору IT платформ для реалізації IoT мережі інтелектуальної системи безпеки житлових комплексів.

2. Методику вибору та формування комплексів інструментальних програмних засобів для реалізації методології розроблення та супроводу (DevOps) інтелектуальної програмної системи безпеки житлових комплексів.

3. Методику побудови проблемно-орієнтованої онтології на основі базової отології GFO з використанням її конструктивів, якими є конфігуроїди, топоїди, хроноїди та ситуоїди.

4. Структурні та архітектурні рішення побудови інтелектуальної програмної системи безпеки житлових комплексів з використанням IT інфраструктури екосистеми фірми «АСТРА-ЛЬВІВ», яка є регіональним провайдером інтернет послуг.

5. Методику побудови бек-енду інтелектуальної програмної системи безпеки житлового комплексу на основі концептуальної моделі, сформованої у вигляді комплексу UML-діаграм відповідних типів.

6. Методику побудови інтерфейсів інтелектуальної програмної системи безпеки ЖК на основі методу персон та методу випадкового лісу, що дозволяє формувати максимально персоналізовані інтерфейси такого класу систем з врахуванням особливостей груп потенційних користувачів.

7. Програмний продукт з комплектом технічної документації, в якому реалізовані базові функції інтелектуальної програмної системи безпеки житлового комплексу, з метою проведення тестових випробувань та дослідної експлуатації першої черги зазначеної системи.

На даний час успішно проводиться широкий комплекс впроваджувальних робіт та натурних експериментів в ряді житлових комплексів у місті Львові.

Заступник директора компанії з технічних питань **Роман Нургалієв**

Головний інженер компанії **Андрій Костко**

Начальник відділу обслуговування мережі **Олександр Малоок**

Додаток Б

+ означає, що компонент присутній за замовчуванням, але не детально, – він не розглядається взагалі.

Таблиця Д.1. Порівняльний аналіз архітектур

Стаття	Реалізація фізичного рівня	Проміжне програмне забезпечення, обчислення, управління даними	Застосування	Функції штучного інтелекту	Контекст	Аналіз і прогнозування цілей користувачів
Fanshawe, 1990	Забезпечення ефективного підключення.	-	-	-	-	-
Donode, 2021	+	в програмно-му забезпе - ченні агента	Чат-бот на смартфоні	-	-	-
Kaldeli, 2013	+	Сервісно-орієнтовані рішення	+	Движок на основі правил для вибору складу послуги	Усвідомлення контексту, що забезпечується штучним інтелектом, вибір доступних послуг, відповідно до контексту.	Цілі представлені явними командами
Yang, 2016	+	+	+	Моделювання цілей	-	Декомпозиція намірів користувача
Michael, 2016	+	-	+	Самостійна оптимізація розумного приладу	Діяти відповідно до ситуації	-
Mocrii, 2018	+	Обчислення здійснюються в хмарі	+	-	-	-
Singh, 2019	+	Блокчейн і хмара для безпеки та обчислень	+	-	-	-

Додаток В

Візуалізація експертного оцінювання платформ

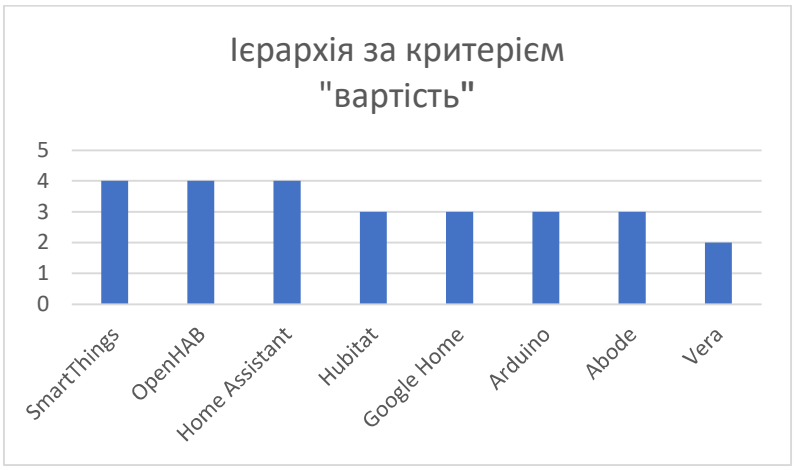


Рис. Д.1 Ієрархія за критерієм "вартість "

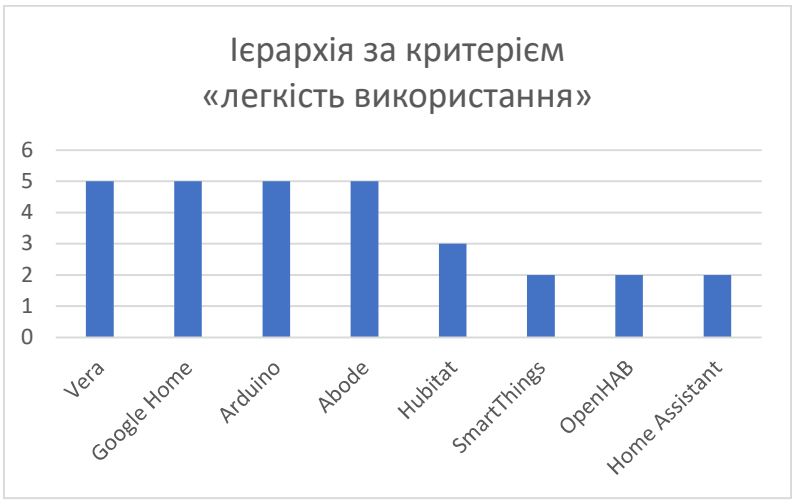


Рис.Д.2. Ієрархія за критерієм «легкість використання»

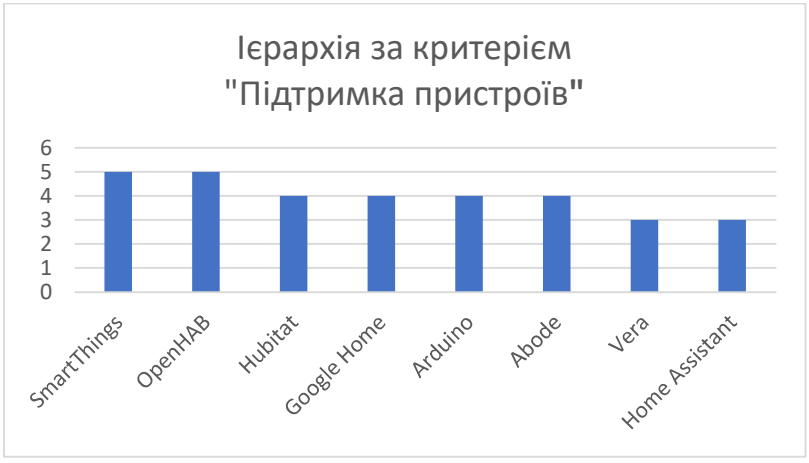


Рис Д.3. Ієрархія за критерієм «підтримка пристроїв»

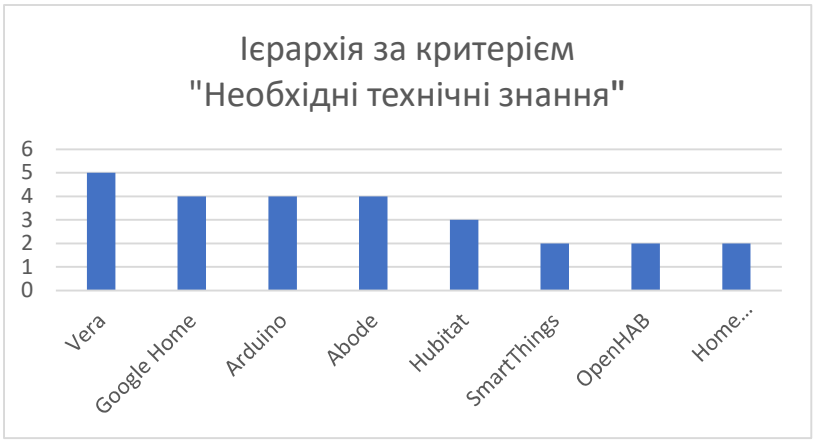


Рис. Д.4. Ієрархія за критерієм «необхідні технічні знання»

Додаток Г

Таблиця Д.2 Функції та завдання систем управління безпекою

Область	Завдання	Опис завдання
Контроль доступу	Аутентифікація та авторизація	Створення стійких фреймворків для автентифікації особистості мешканців і гостей. Це може включати біометричні технології, облікові дані RFID та мобільну доступність.
	Управління відвідувачами	Сприяння ефективній процедурі реєстрації гостей, надання тимчасового доступу та нагляд за поведінкою відвідувачів.
	Безпека воріт та в'їзду	Управління точками входу та виходу, що охоплюють ворота та фойє, за допомогою персоналу охорони, автоматизованих шлагбаумів та механізмів спостереження
	Доступ до спільного простору	Регулювання доступу до комунальних зручностей, таких як фітнес-центри, басейни та

		парки, для запобігання несанкціонованому проникненню та гарантування безпеки мешканців.
Спостереження та моніторинг	Відеоспостереження	Створення та підтримка мережі камер відеоспостереження замкнутого телебачення у комунальних зонах, точках входу та кордонах для спостереження за діяльністю та протидія протиправній поведінці
	Виявлення вторгнень	Розміщення давачів і механізмів оповіщення для виявлення несанкціонованого доступу або порушень безпеки
	Віддалений моніторинг	Сприяння персоналу служби безпеки процедурі нагляду за передачею даних і сповіщеннями в реальному часі з централізованого об'єкта моніторингу або віддалено за допомогою мобільних апаратів

	Зберігання та управління даними	Забезпечення надійного збереження записів і журналів спостереження для подальшого використання та дотримання правил зберігання даних
Кібербезпека	Мережева безпека	Захист мережевої інфраструктури спільноти від кіберзагроз за допомогою використання брандмауерів, систем виявлення/запобігання вторгнень та безпечних протоколів зв'язку.
	Захист даних	Забезпечення конфіденційності та цілісності особистої інформації мешканців, зібраної різноманітними інтелектуальними пристроями та системами
	Реагування на інциденти	Формулювання та реалізація стратегій реагування на інциденти кібербезпеки, що охоплюють виявлення, локалізацію, ліквідацію та відновлення.

	Регулярні аудити безпеки	Проведення періодичних оцінок безпеки та аудитів для виявлення та усунення вразливостей у системі.
Управління реагуванням на надзвичайні ситуації	Системи сигналізації	Встановлення пристроїв виявлення пожежі, пристроїв виявлення диму та систем сповіщення про надзвичайні ситуації для гарантування оперативних сповіщень та реагування.
	Планування евакуації	Формулювання та відпрацювання протоколів евакуації для різних надзвичайних ситуацій, таких як пожежі, стихійні лиха або ризики безпеці.
	Інтеграція з місцевими органами влади	Створення керівних принципів для оперативної комунікації та співпраці з місцевими підрозділами реагування на надзвичайні ситуації та правоохоронними органами.
	Сповіщення в реальному часі	Запуск систем, які можуть надсилати сповіщення в режимі

		реального часу мешканцям та керівництву під час криз.
Екологічний моніторинг та контроль	Моніторинг якості повітря	Встановлення давачів для оцінювання якості атмосфери і рівня забруднення, тим самим гарантуючи здорове середовище проживання.
	Моніторинг якості води	Забезпечення цілісності та обсягів водопостачання шляхом спостереження та адміністрування систем очищення води.
	Контроль температури і вологості	Виконання інтелектуальних систем опалення, вентиляції та кондиціонування повітря, які моделюють температуру та рівень вологості відповідно до умов навколишнього середовища та заповнюваності.
	Управління енергією	Контроль та оптимізація споживання енергії для зменшення витрат та впливу на навколишнє середовище.

Технічне обслуговування та модернізація	Обслуговування системи	Регулярна оцінка та збереження систем безпеки для визначення їх експлуатаційної ефективності.
	Оновлення програмного забезпечення	Впровадження виправлень і вдосконалень захисного програмного забезпечення та мікропрограм для захисту від нових загроз.
	Оновлення апаратного забезпечення	Заміна застарілих апаратних компонентів для забезпечення ефективності та надійності систем безпеки.
	Управління постачальниками	Співпраця з постачальниками обладнання та послуг для підтримки та вдосконалення систем безпеки та апаратів.

Таблиця Д.1 Приклад специфікації ситуоїда для спроби несанкціонованого доступу

Опис ситуоїда
Назва: Спроба несанкціонованого доступу Контекст: Ситуація, коли неавторизована особа намагається отримати доступ до об’єкта, що охороняється. Тимчасові межі: Інцидент стався між 2:00 та 2:15 10 липня 2024 року.

Просторові межі**:** Фізичні межі інциденту обмежуються головним входом до об'єкта, що охороняється, та найближчою прилеглою територією.
Залучені суб’єкти
1. Зловмисник - невідома особа, яка намагається отримати доступ на об’єкт. 2. Охоронець -черговий охоронець, який стежить за входом. 3. Система контролю доступу - система, яка керує входом на об’єкт, включаючи зчитувачі ключ-карт і дверні замки. 4. Камери відеоспостереження – камери, що охоплюють головний вхід і прилеглу територію. 5. Система сигналізації - система, яка запускає сигнал тривоги в разі виявлення несанкціонованого доступу.
Відношення та взаємодії
Взаємодія зловмисника з системою контролю доступу - зловмисник намагається використати підроблену або вкрадену картку-ключ, щоб обійти систему контролю доступу. Взаємодія системи контролю доступу та камер відеоспостереження - система контролю доступу реєструє спробу несанкціонованого доступу, а камери відеоспостереження фіксують відеозапис зловмисника. Взаємодія охоронної сигналізації з системою охоронної сигналізації - система контролю доступу виявляє спробу несанкціонованого доступу та спрацьовує сигналізація. Взаємодія охоронця та зловмисника - при спрацьовуванні сигналізації охоронець реагує наближенням до входу та спробою затримати зловмисника
Структуроване подання ситуоїда
Цей ситуоїд інкапсулює весь інцидент безпеки, включаючи: - Просторову область (головна вхідна зона об’єкта). - Часовий період (з 2:00 до 2:15 10 липня 2024 року). - Задіяних осіб (зловмисник, охоронець, система контролю доступу, камери відеоспостереження, сигналізація).

- Взаємозв’язки між цими сутностями, такі як взаємодія зловмисника з системою контролю доступу, ведення протоколу події системою, камери відеоспостереження, які фіксують інцидент, і реакція охоронця.
Призначення ситуоїда
Аналіз - цей ситуоїд може використовуватися для аналізу інциденту безпеки, що дозволяє системі або персоналу служби безпеки зрозуміти послідовність подій, виявити слабкі місця в системі безпеки та покращити майбутні реакції. Навчання - може бути використано в навчальних сценаріях для інтелектуальної системи для відпрацювання реакцій на подібні спроби несанкціонованого доступу. Документація - ситуоїд служить офіційним записом інциденту, на який можна посилатися у звітах або розслідуваннях.

Рис. Д.5 Онтологія концепту «Охоронні системи»

Додаток Ґ

Класифікація протоперсон за різними критеріями

Обравши за критерій класифікації - способи взаємодії з безпековою системою можемо виділити такі групи протоперсон. До першої будуть відноситись протоперсони, що володіють навичками віддаленого управління та моніторингу стану безпеки з використанням комплексної безпекової інформаційної системи багатоквартирного будинку (Олександр, Ігор, Анна). Другу групу складають протоперсони, що володіють навичками локального управління та моніторингу (Марія, Павло).

Класифікація за критерієм «основні мотиваційні фактори» дозволяє виділити такі групи протоперсон. Першу групу складають протоперсони, що надають перевагу безпеці особистого життєвого простору (Олександр). Мету щодо забезпечення безпеки родини переслідує друга група протоперсон (Марія). До третьої групи належать протоперсони, для яких важливим є забезпечення особистої безпеки та оперативне реагування на надзвичайні ситуації (Павло). До четвертої групи відносяться прихильники ефективного управління безпекою будинку та мешканців загалом - Анна. Безпека орендованого майна та підвищення привабливості його для орендарів турбує протоперсон п’ятої групи, представником якої є - Ігор.

Класифікуючи протоперсони за рівнем фінансових можливостей доцільним видається виділення трьох груп. Перша має високий рівень фінансових можливостей (Ігор, Анна). Друга група протоперсон має середній рівень фінансових можливостей (Олександр, Марія). До третьої групи належать протоперсони із низький рівень фінансових можливостей (Павло).

Класифікації протоперсон за віком (потенційних користувачів або осіб, що можуть взаємодіяти із інтелектуальною системою безпеки житлових комплексів), сприяє кращому розумінню потреб користувачів різних вікових груп і налаштування з врахуванням цього фактору відповідних функцій системи безпеки. Першу групу складають діти до 12 років. Наступну групу складають підлітки - 13–18 років. Третю групу складає молодь 19–35 років. Четверту групу

складають дорослі 36–60 років. П’ята група формується з літніх людей віку 60+ років, які мають специфічні потреби та вимоги до реалізації функцій безпеки та спрощеного доступу до певних систем, зокрема це стосується інвалідів [124]. Класифікація протоперсон за віком допомагає налаштувати безпекову систему відповідно до потреб кожної вікової групи. Для дітей та підлітків акцент робиться на контролі доступу та моніторингу, для дорослих – на персоналізації доступу та захисту власності, а для літніх людей – на спрощеному використанні та особливій увазі до особистої безпеки.

Ім’я·Олександр	Вік·28·років
Сімейний·стан:·Неодружений,·живе·один	
Рід·занять:·IT-спеціаліст,·працює·віддалено	
Житлова·ситуація:·Живе··у·однокімнатній·квартирі·в·сучасному·багатоквартирному·будинку	
Потреби:·Безпека·особистого·простору,·зручність·використання·системи,·можливість·віддаленого·контролю·через·мобільний·додаток	
Мотивація·використання·застосунку:·¶ Забезпечення·безпеки·власних·речей,·зниження·тривожності·щодо·безпеки·квартири·під·час·відсутності,··віддалений·моніторинг	
Проблемні·аспекти:··Часто·подорожує·і·потребує·можливості·віддаленого·моніторингу·та·управління·безпекою·квартири	

Рис. Д.6 Протоперсона 1

Протоперсона 2: Марія, молода мати.

Вік: 32 роки.

Рід занять: Маркетолог, працює частково віддалено.

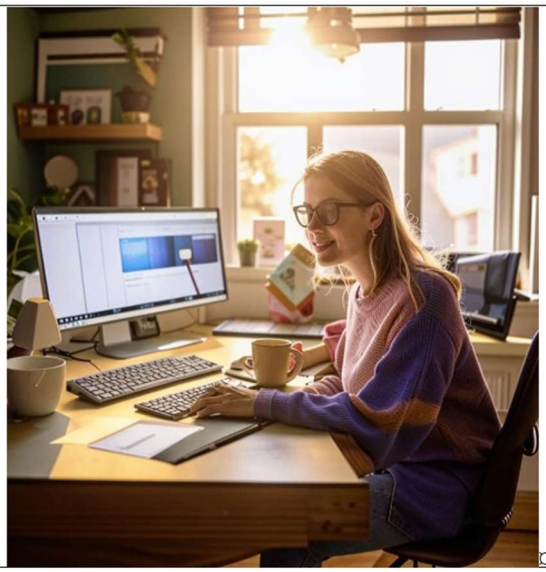
Демографічні дані: Заміжня, двоє дітей (5 і 7 років).

Житлова ситуація: Власна квартира в новобудові.

Потреби: Безпека дітей, контроль за доступом до квартири, можливість отримувати сповіщення про стан безпеки.

Технічні потреби: Потребує швидкого і простого доступу до інформації про стан безпеки, можливість налаштування системи для захисту дітей, сповіщення про стан безпеки.

Мотивація: Забезпечення безпеки дітей, спокій щодо безпеки домівки під час відсутності вдома.



Ім'я: Марія	Вік: 32 роки
Сімейний стан: Заміжня, двоє дітей (5 і 7 років)	
Рід занять: Маркетолог, працює частково віддалено	
Житлова ситуація: Власна трикімнатна квартира в новобудові	
Потреби: Безпека дітей, контроль за доступом до квартири, можливість отримувати сповіщення про стан безпеки	
Мотивація використання застосунку: Забезпечення безпеки дітей, спокій щодо безпеки домівки під час відсутності вдома	
Проблемні аспекти: Потребує швидкого і простого доступу до інформації про стан безпеки, можливість налаштування системи для захисту дітей	

Рис. Д.7 Протоперсона 2

Протоперсона 3: Павло, пенсіонер.

Вік: 67 років.

Рід занять: Пенсіонер, колишній інженер.

Демографічні дані: Вдівець, живе один.

Житлова ситуація: Власна квартира в багатоквартирному будинку.

Потреби: Проста в користуванні система безпеки, контроль за доступом до квартири, можливість сповіщення у випадку надзвичайних ситуацій (пожежа, витік газу та ін.).

Технічні потреби: Потребує простого і інтуїтивного інтерфейсу, має обмежений бюджет на придбання і обслуговування системи.

Мотивація: Забезпечення власної безпеки, зниження ризиків надзвичайних ситуацій, забезпечення швидкого реагування у випадку небезпеки.

Ім'я: Павло	Вік: 67 років
Сімейний стан: Вдівець, живе один	
Рід занять: Пенсіонер	
Житлова ситуація: Власна однокімнатна квартира в багатоквартирному будинку	
Потреби: Проста в користуванні система безпеки, контроль за доступом до квартири, можливість сповіщення у випадку надзвичайних ситуацій (пожежа, витік газу)	
Мотивація використання застосунку: Забезпечення власної безпеки, зниження ризиків надзвичайних ситуацій, забезпечення швидкого реагування у випадку небезпеки	
Проблемні аспекти: Потребує простого і інтуїтивного інтерфейсу, має обмежений бюджет на придбання і обслуговування системи	

Рис. Д.8 Протоперсона 3

Протоперсона 4: Анна, член адміністрації житлового комплексу.

Вік: 45 років.

Рід занять: член адміністрації житлового комплексу.

Демографічні дані: Заміжня, двоє дітей підліткового віку.

Житлова ситуація: Живе в окремому приватному будинку, працює в багатоквартирному житловому комплексі.

Потреби: Можливість централізованого контролю і управління безпекою будинку, моніторинг спільних зон (вестибюлі, ліфти, коридори), швидке реагування на інциденти.

Технічні потреби: Потребує надійної і легкої в управлінні системи, яка дозволяє швидко отримувати інформацію про стан безпеки всіх зон будинку.

Мотивація: Забезпечення безпеки мешканців, підвищення ефективності управління будинком, зниження кількості інцидентів.

Протоперсона 5: Ігор, власник квартири, яка здається в оренду

Вік: 38 років

Рід занять: Підприємець, займається орендою нерухомості

Демографічні дані: Одружений, одна дитина

Житлова ситуація: Володіє декількома квартирами в різних будинках, які здає в оренду

Потреби: Контроль за станом безпеки квартир, можливість моніторингу та управління віддалено, сповіщення про інциденти

Технічні потреби: Потребує надійної системи для контролю за кількома об'єктами одночасно, бажає мінімізувати час і витрати на обслуговування системи

Мотивація: Забезпечення безпеки майна, зниження ризиків псування власності, підвищення привабливості орендованих квартир для клієнтів

Додаток Д

Код 1.

Код, поданий мовою Python, для оцінювання важливості ознак з використанням методу випадкового лісу:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier

# Приклад даних
data = [
    [28, 0, 0, 0, 0, 1, 2, 1, 0],  # Молодий професіонал
    [32, 1, 2, 1, 1, 0, 0, 2, 1],  # Молода мати
    [67, 2, 0, 2, 2, 0, 0, 0, 2],  # Пенсіонер
    [45, 1, 2, 3, 3, 1, 0, 1, 3],  # Адміністратор
    [38, 1, 1, 4, 3, 0, 1, 2, 4],  # Власник квартир
]

# Перетворення даних у numpy array для зручності роботи
data = np.array(data)

# Розділення даних на критерії та мітки
X = data[:, :-1]  # Критерії
y = data[:, -1]   # Мітки

# Навчання моделі випадкового лісу
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X, y)

# Оцінка важливості ознак
importance = model.feature_importances_
```

```
# Візуалізація важливості ознак
features = ['Вік', 'Сімейний стан', 'Наявність дітей', 'Рід занять', 'Тип роботи', 'Тип житла', 'Частота подорожей', 'Бюджет']
plt.barh(features, importance)
plt.xlabel('Важливість ознаки')
plt.ylabel('Ознаки')
plt.title('Важливість ознак у моделі випадкового лісу')
plt.show()
```

Код 2.

Приклад коду мовою Python моделі, яка складається з множини дерев рішень, кожне з яких навчено на випадковій вибірці даних з навчального набору

```
From sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Приклад даних (спрощений)
data = [
    [28, 0, 0, 0, 0, 1, 2, 1, 0],  # Молодий професіонал
    [32, 1, 2, 1, 1, 0, 0, 2, 1],  # Молода мати
    [67, 2, 0, 2, 2, 0, 0, 0, 2],  # Пенсіонер
    [45, 1, 2, 3, 3, 1, 0, 1, 3],  # Адміністратор
    [38, 1, 1, 4, 3, 0, 1, 2, 4],  # Власник квартир
]
# Розділення даних на фічі та метки
X = [row[:-1] for row in data]  # Критерії
y = [row[-1] for row in data]   # Мітки
# Розділення на навчальний і тестовий набір
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Створення і навчання моделі випадкового лісу
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
# Прогнозування на тестовому наборі
y_pred = model.predict(X_test)
# Оцінка точності моделі
accuracy = accuracy_score(y_test, y_pred)
print(f"Точність моделі: {accuracy * 100:.2f}%")
```

Проводиться класифікація нових користувачів. Наводимо код мовою Python**.**

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Приклад даних (спрощений)
data = [
    [28, 0, 0, 0, 0, 1, 2, 1, 0],  # Молодий професіонал
    [32, 1, 2, 1, 1, 0, 0, 2, 1],  # Молода мати
    [67, 2, 0, 2, 2, 0, 0, 0, 2],  # Пенсіонер
    [45, 1, 2, 3, 3, 1, 0, 1, 3],  # Адміністратор
    [38, 1, 1, 4, 3, 0, 1, 2, 4],  # Власник квартир
]
# Розділення даних на критерії та мітки
X = [row[:-1] for row in data]  # Критерії
y = [row[-1] for row in data]   # Мітки
```

Додаток Е

Таблиця Д.4 Початкові дані

Вік	Сімей-н	Наявність	Рід занять	Тип	Тип житла	Частота -жей	Бюджет	Класифікація
28	Не од-й	Немає	IT-ість	Віддалено	Оренда	Висока	Середній	Молодий о-нал
32	Заміжня	Двоє	Маркетолог	Частково	Власне	Низька	Високий	Молода мати
67	Вдівець	Немає	Пенсіонер	Не працює	Власне	Низька	Низький	Пенсіонер
45	Заміжня	Двоє	Адміністра-	Офлайн	Оренда	Низька	Середній	Адміністратор
38	Одру-	Одна	Підприємец	Офлайн	Власне	Середня	Високий	Власник

Додаток Є

Характеристики критеріїв подамо наступним чином:

Вік (числова)

Сімейний стан (категорійна, закодована чисельно)

Наявність дітей (категорійна, закодована чисельно)

Рід занять (категорійна, закодована чисельно)

Тип роботи (категорійна, закодована чисельно)

Тип житла (категорійна, закодована чисельно)

Частота подорожей (категорійна, закодована чисельно)

Бюджет (категорійна, закодована чисельно)

Класифікація (категорійна, закодована чисельно, мітка класу)

Таблиця Д.5 Приклад розширеного набору даних

Вік	Сімейний стан	Наявність дітей	Рід занять	Тип роботи	Тип житла	Частота подорожей	Бюджет	Класифікація
28	Не одружений	Немає	IT-спеціаліст	Віддалено	Орендоване	Висока	Середній	0 (Молодий професіонал)
29	Не Одружений	Немає	IT-спеціаліст	Віддалено	Орендоване	Висока	Середній	0 (Молодий професіонал)
30	Не Одружений	Немає	IT-спеціаліст	Віддалено	Власне	Висока	Середній	0 (Молодий професіонал)
32	Заміжня	Двоє	Маркето-лог	Частково	Власне	Низька	Високий	1 (Молода мати)
33	Заміжня	Двоє	Маркето-лог	Частково	Власне	Низька	Високий	1 (Молода мати)
34	Заміжня	Двоє	Маркето-лог	Частково	Орендо-ване	Низька	Високий	1 (Молода мати)
67	Вдівець	Немає	Пенсіонер	Не працює	Власне	Низька	Низький	2(Пенсіонер)
68	Вдівець	Немає	Пенсіонер	Не працює	Власне	Низька	Низький	2 (Пенсіонер)
69	Вдівець	Немає	Пенсіонер	Не працює	Орендо-ване	Низька	Низький	2 (Пенсіонер)
45	Заміжня	Двоє	Адміністратор	Офлайн	Орендо-ване	Низька	Середній	3 (Адміністратор)
46	Заміжня	Двоє	Адміністратор	Офлайн	Орендо-ване	Низька	Середній	3 (Адміністратор)
47	Заміжня	Двоє	Адміністратор	Офлайн	Власне	Низька	Середній	3 (Адміністратор)
38	Одружений	Одна	Підприє-мець	Офлайн	Власне	Середня	Високий	4 (Власник квартир)
39	Одружений	Одна	Підприє-мець	Офлайн	Власне	Середня	Високий	4 (Власник квартир)
40	Одружений	Одна	Підприє-мець	Офлайн	Орендо-ване	Середня	Високий	4 (Власник квартир)

Додаток Ж

Код процедури класифікації нового користувача:

```
# Дані нового користувача
new_user = [30, 0, 0, 1, 1, 1, 2, 1]  # Наприклад, молодий професіонал
# Прогнозування класу для нового користувача
predicted_class = model.predict([new_user])
print(f"Користувач належить до класу: {predicted_class[0]}")
```