МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кваліфікаційна наукова
праця на правах рукопису

МИКОЛАЙЧУК ВІРА РОМАНІВНА

УДК 004.051

ДИСЕРТАЦІЯ

МЕТОДИКА ОЦІНКИ ЕФЕКТИВНОСТІ РОБОТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ МІСЦЕВОСТІ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Спеціальність 123 «Комп'ютерна інженерія»
Галузь знань 12 «Інформаційні технології»
Подається на здобуття наукового ступеня
доктора філософії

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

Науковий керівник: СТОРЧАК Каміла Павлівна, доктор технічних наук, професор

Електронний підпис
Миколайчук
Віра
Романівна
Ідентифікаційний
код
3495412107
Підписано у Вчасно

Київ – 2023

АНОТАЦІЯ

Миколайчук В.Р. Методика оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії в галузі знань 12 – Інформаційні технології, за спеціальністю 123 – Комп'ютерна інженерія. – Державний університет інформаційно-комунікаційних технологій. – Київ, 2023.

Дисертаційна робота присвячена актуальній науковій задачі розробки методики оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання.

Тематика дисертаційного дослідження відповідає тимчасовому стандарту та фаховим компетентностям освітньо-наукової програми підготовки докторів філософії з комп'ютерної інженерії Державного університету інформаційно-комунікаційних технологій Міністерства освіти і науки України, а саме: фундаментальним науковим дослідженням теоретико-методологічних, науково-методичних та прикладних засад підвищення ефективності інноваційної та виробничої діяльності підприємства, а також вдосконаленню процесу забезпечення впровадження новітніх інформаційних технологій на об'єктах інформаційної діяльності.

Дисертаційне дослідження присвячено аналізу систем моніторингу місцевості (СММ) та їх ефективності у виявленні та ідентифікації рухомих об'єктів. Ці системи відіграють важливу роль у сучасних умовах, коли потреба у високоточному моніторингу територій стає все більш актуальною.

Актуальність дослідження СММ, особливо у контексті виявлення та ідентифікації рухомих об'єктів, можна обґрунтувати наступним чином:

1. Зростання технологічних можливостей: Сучасний світ переживає бурхливий розвиток технологій, що веде до появи нових засобів моніторингу та контролю місцевості. Це створює потребу у вдосконаленні існуючих систем та розробці нових методів аналізу.

2. Безпека та оборона: У контексті глобальної безпеки та оборони здатність швидко виявляти та ідентифікувати рухомі об’єкти може бути критично важливою. Це може стосуватися як військових операцій, так і мирного моніторингу кордонів.

3. Екологічний моніторинг: Зміни клімату, незаконне вирубування лісів - це лише декілька з проблем, які можна виявляти та контролювати за допомогою ефективних СММ.

4. Міське планування та розвиток: У міських умовах СММ можуть використовуватися для моніторингу транспортних потоків, виявлення аварійних ситуацій, контролю за будівництвом та іншими аспектами міського життя.

5. Вдосконалення технологій: Як було вказано у дослідженні, існуючі сенсорні мережі можуть бути неефективними у певних умовах. Це підкреслює потребу у додаткових дослідженнях та розробці нових підходів.

6. Економічний аспект: Ефективні системи моніторингу можуть призвести до зниження витрат, оптимізації ресурсів та підвищення продуктивності у ряді секторів економіки.

Отже, актуальність дослідження систем моніторингу місцевості, зокрема у контексті виявлення та ідентифікації рухомих об’єктів, очевидна. Вони відіграють ключову роль у багатьох аспектах сучасного життя, від оборони до екології, і потребують постійного вдосконалення та адаптації до змінюваних умов.

Аналіз практичних підходів до СММ виявив, що більшість сучасних систем базуються на сенсорних мережах. Ці мережі дозволяють збирати детальну інформацію про місцевість, але мають певні обмеження. Основна проблема полягає у виявленні та ідентифікації рухомих об’єктів, особливо тих, які намагаються уникнути ідентифікації. У таких випадках сенсорні мережі можуть бути не ефективними.

Для підвищення ефективності СММ у виявленні рухомих об’єктів пропонується використовувати роботизовані засоби. Ці засоби можуть включати

в себе різноманітні дрони, роботи-землероби та інші автоматизовані системи, які здатні відстежувати рухомі об’єкти на місцевості та ідентифікувати їх.

Аналіз теоретичних підходів до оцінки ефективності роботизованих СММ показав, що основна проблема полягає у моделюванні динаміки ідентифікації рухомих об’єктів. Це ускладнює процес оцінки, адже потрібно враховувати численні фактори: швидкість руху об’єкта, його розміри, форму, колір та інші характеристики, які можуть впливати на його видимість для сенсорів.

Для підвищення ефективності систем моніторингу місцевості необхідно комбінувати сенсорні мережі з роботизованими засобами. Такий підхід дозволить забезпечити високу точність виявлення та ідентифікації рухомих об’єктів, незалежно від їх спроб уникнути ідентифікації. Також необхідно розробити нові методики моделювання динаміки ідентифікації, щоб точніше оцінювати ефективність роботизованих СММ.

Для забезпечення заданих високих вимог до якісних показників продуктивності систем моніторингу, що є частиною організації сучасних підприємств, необхідна розробка методики оцінки ефективності таких систем.

Перший розділ дослідження присвячено аналізу практичних та науково-методологічних підходів до роботизованих систем моніторингу місцевості. Було встановлено, що сенсорні мережі, хоча і є ефективними у багатьох випадках, можуть мати обмеження у виявленні рухомих об'єктів. Роботизовані засоби, у поєднанні з методами машинного навчання, спрямовані на подолання викликів пов’язаних з динамічністю об’єктів ідентифікації, проте потребують подальших наукових досліджень для оптимізації їх роботи. Наприкінці розділу сформовані такі наукові завдання:

– Обґрунтувати показники та критерії оцінки ефективності роботизованих систем моніторингу місцевості (РСММ).

– Розробити модель середовища моніторингу з урахуванням роботизованих засобів та рухомих об'єктів.

– Розробити модель роботизованого засобу (що включає розробку алгоритму формування множини можливих дій, які система може виконувати для ідентифікації об'єкта та модель вибору поведінки).

– Розробити методику оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання, а саме навчання з підкріпленням ((Reinforcement Learning) RL).

У другому розділі проводиться аналіз існуючих теоретичних підходів до оцінки ефективності РСММ, а також досліджуються методи RL та їх застосування для вирішення цієї задачі. Розглядаються основні концепції та алгоритми RL, їх переваги та обмеження. Аналізуються можливості використання RL для оптимізації стратегій моніторингу, адаптації до змінюваних умов та підвищення загальної ефективності РСММ. Цей розділ розкриває теоретичні основи оцінки ефективності РСММ, а також розкриває потенціал застосування методів RL для оптимізації їх роботи.

У третьому розділі розглядаються ключові аспекти розробки методики оцінки ефективності роботизованої системи моніторингу місцевості. Основна увага приділяється обґрунтуванню показників та критеріїв оцінки, моделюванню середовища моніторингу, розробці моделі роботизованого засобу та об’єкту моніторингу та формулюванню методики оцінки ефективності. Цей розділ надає чітке розуміння процесу розробки методики оцінки ефективності РСММ, включаючи визначення ключових показників та критеріїв, моделювання середовища та роботизованого засобу, а також формулювання загальних принципів оцінки.

У четвертому розділі дослідження акцентується на практичному застосуванні розробленої методики оцінки ефективності роботизованої системи моніторингу місцевості. Основна увага приділяється програмній реалізації цієї методики у вигляді Framework, який базується на передових технологіях машинного навчання, зокрема, комп'ютерному зору ((Computer vision) CV) та навчанню з підкріпленням. Цей розділ надає докладний огляд процесу програмної реалізації розробленої методики оцінки ефективності РСММ,

демонструючи її практичну цінність та можливості застосування в реальних умовах.

Для досягнення поставленої мети дисертаційної роботи були вирішені такі завдання:

1. Удосконалено модель середовища моніторингу, яка враховує протидію роботизованих засобів та рухомих об’єктів та за допомогою використання згорткових нейронних мереж аналізує середовище, що дало змогу підвищити адаптивність системи до змінних умов та дало можливість детально відтворити реальні умови, в яких діють роботизовані засоби, включаючи перешкоди, зони видимості та різні типи поверхонь.

2. Удосконалено модель роботизованого засобу, що включає розробку алгоритму формування множини можливих дій, які система може виконувати для ідентифікації об’єкта та модель вибору поведінки, використано техніки глибинного навчання для автоматичного вивчення оптимальних стратегій взаємодії з навколишнім середовищем, що дало можливість враховувати взаємодію з іншими роботизованими засобами та об'єктами моніторингу.

3. Вперше розроблено методику оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання, яка включає розроблену модель навчання з підкріпленням для точного визначення характеристик об'єкта моніторингу та роботизованого засобу, що забезпечило високу адаптивність та ефективність порівняно з існуючими методиками, особливо в складних динамічних умовах.

Дисертація виконувалась в Державному університеті інформаційно-комунікаційних технологій. Обраний напрям досліджень відповідає тематиці науково-дослідних робіт Державного університету інформаційно-комунікаційних технологій.

Ключові слова: методика оцінки ефективності, машинне навчання, система моніторингу місцевості, сенсорні мережі, рухомі об'єкти, роботизовані засоби, навчання з підкріпленням, комп'ютерний зір, моделювання середовища, Framework, швидкість виявлення, точність ідентифікації.

ABSTRACT

Mykolaichuk V.R. Methodology for assessing the effectiveness of the robotic terrain monitoring system based on machine learning. – Qualifying scientific work as a manuscript.

Dissertation for obtaining the scientific degree of Doctor of Philosophy in the field of knowledge 12 - Information technologies, specialty 123 - Computer engineering. - State University of Information and Communication Technologies. - Kyiv, 2023.

The dissertation is devoted to the actual scientific task of developing a methodology for evaluating the effectiveness of a robotic terrain monitoring system based on machine learning.

The topic of the dissertation study corresponds to the temporary standard and professional competences of the educational and scientific program for training doctors of philosophy in computer engineering of the State University of Information and Communication Technologies of the Ministry of Education and Science of Ukraine, namely: fundamental scientific research of theoretical-methodological, scientific-methodological and applied principles of promotion the efficiency of the enterprise's innovative and production activities, as well as improving the process of ensuring the introduction of the latest information technologies at the objects of information activities.

The dissertation study is devoted to the analysis of terrain monitoring systems (TMS) and their effectiveness in detecting and identifying moving objects. These systems play an important role in modern conditions, when the need for high-precision monitoring of territories is becoming more and more urgent.

The relevance of the TMS study, especially in the context of detection and identification of moving objects, can be justified as follows:

1. Growth of technological capabilities: The modern world is experiencing a rapid development of technologies, which leads to the emergence of new means of

monitoring and controlling the terrain. This creates a need to improve existing systems and develop new methods of analysis.

2. Security and Defense: In the context of global security and defense, the ability to quickly detect and identify moving objects can be critical. This can apply to both military operations and peaceful border monitoring.

3. Environmental monitoring: Climate change, illegal deforestation are just a few of the issues that can be identified and monitored through effective TMSs.

4. Urban planning and development: In urban settings, TMSs can be used to monitor traffic flows, detect emergency situations, control construction and other aspects of urban life.

5. Technology improvements: As the study indicated, existing sensor networks may not be effective in certain conditions. This highlights the need for additional research and development of new approaches.

6. Economic aspect: Effective monitoring systems can lead to cost reduction, resource optimization and productivity improvement in a number of sectors of the economy.

Therefore, the relevance of the study of terrain monitoring systems, in particular in the context of detection and identification of moving objects, is obvious. They play a key role in many aspects of modern life, from defense to ecology, and need constant improvement and adaptation to changing conditions.

Analysis of practical approaches to TMS revealed that most modern systems are based on sensor networks. These networks allow you to collect detailed information about the area, but have certain limitations. The main problem is the detection and identification of moving objects, especially those that try to avoid identification. In such cases, sensor networks may not be effective.

To increase the effectiveness of the TMS in detecting moving objects, it is proposed to use robotic means. These tools can include a variety of drones, agricultural robots, and other automated systems that can track and identify moving objects on the terrain.

Analysis of theoretical approaches to evaluating the effectiveness of robotic TMSs showed that the main problem is modelling the dynamics of identification of moving objects. This complicates the evaluation process, because it is necessary to take into account numerous factors: the speed of movement of the object, its size, shape, color and other characteristics that can affect its visibility for sensors.

To increase the effectiveness of terrain monitoring systems, it is necessary to combine sensor networks with robotic means. This approach will ensure high accuracy of detection and identification of moving objects, regardless of their attempts to avoid identification. It is also necessary to develop new methods for modelling the dynamics of identification in order to more accurately assess the effectiveness of robotic TMSs.

In order to ensure the set high requirements for qualitative performance indicators of monitoring systems, which are part of the organization of modern enterprises, it is necessary to develop a methodology for evaluating the effectiveness of such systems.

The first chapter of the study is devoted to the analysis of practical and scientific-methodological approaches to robotic terrain monitoring systems. It was found that sensor networks, although effective in many cases, may have limitations in detecting moving objects. Robotic means, combined with machine learning methods, are aimed at overcoming the challenges associated with the dynamism of identification objects, but require further scientific research to optimize their work. At the end of the chapter, the following scientific tasks are formed:

– To substantiate the indicators and criteria for evaluating the effectiveness of robotic terrain monitoring systems (RTMS).

– Develop a model of the monitoring environment taking into account robotic means and moving objects.

– Develop a model of a robotic tool (which includes the development of an algorithm for forming a set of possible actions that the system can perform to identify an object and a model of behavior selection)

– To develop a methodology for evaluating the effectiveness of a robotic terrain monitoring system based on machine learning, namely reinforcement learning.

The second chapter analyses existing theoretical approaches to evaluating the effectiveness of RTMS, and also explores RL methods and their application to solving this problem. The main concepts and algorithms of RL, their advantages and limitations are considered. Possibilities of using RL to optimize monitoring strategies, adapt to changing conditions, and improve the overall efficiency of the RTMS are analyzed. This chapter reveals the theoretical basis for evaluating the effectiveness of RTMS, as well as revealing the potential of using RL methods to optimize their work.

In the third chapter, the key aspects of the development of the methodology for evaluating the effectiveness of the robotic terrain monitoring system are considered. The main attention is paid to the substantiation of indicators and evaluation criteria, modelling of the monitoring environment, development of a model of the robotic tool and the monitoring object, and the formulation of the effectiveness assessment methodology. This chapter provides a clear understanding of the process of developing a methodology for evaluating the effectiveness of RTMS, including defining key indicators and criteria, modelling the environment and the robotic tool, and formulating general evaluation principles.

The fourth chapter of the study focuses on the practical application of the developed methodology for evaluating the effectiveness of the robotic terrain monitoring system. The main focus is on the software implementation of this technique in the form of a Framework, which is based on advanced machine learning technologies, in particular, computer vision and reinforcement learning. This chapter provides a detailed overview of the process of software implementation of the developed methodology for evaluating the effectiveness of RTMS, demonstrating its practical value and the possibilities of application in real conditions.

To achieve the goal of the dissertation, the following tasks were solved:

1. The model of the monitoring environment was improved, which takes into account the opposition of robotic means and moving objects and analyzes the environment with the help of convolutional neural networks, which made it possible to increase the adaptability of the system to changing conditions and made it possible to

reproduce in detail the real conditions in which robotic means operate. including obstacles, areas of visibility and different types of surfaces.

2. The model of the robotic tool was improved, which includes the development of an algorithm for forming a set of possible actions that the system can perform to identify the object and a model of behavior selection, deep learning techniques were used to automatically learn optimal strategies for interacting with the environment, which made it possible to take into account the interaction with other robotic means and monitoring objects.

3. For the first time, a technique for evaluating the effectiveness of a robotic terrain monitoring system based on machine learning was developed, which includes a developed reinforcement learning model for accurately determining the characteristics of the monitoring object and the robotic tool, which ensured high adaptability and efficiency compared to existing methods, especially in complex dynamic conditions

The dissertation was completed at the State University of Information and Communication Technologies. The chosen direction of research corresponds to the topic of research works of the State University of Information and Communication Technologies.

Keywords: performance evaluation method, machine learning, terrain monitoring system, sensor networks, moving objects, robotics, reinforcement learning, computer vision, environment modelling, Framework, detection rate, identification accuracy.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, у яких опубліковані основні наукові результати дисертації:

1. N. Tupitsin та V. Mykolaichuk, “System of UAV failure modeling for external pilot training in emergency situations”, *2019 IEEE 5th Int. Conf. Actual Problems Unmanned Aerial Vehicles Develop. (APUAVD)*, № 5, с. 57–60, 2019.
2. О. В. Полоневич, І. М. Срібна, В. Р. Миколайчук, О. М. Ткаленко та В. В. Шкапа, “Застосування штучного інтелекту для керування проектами”, *Зв’язок*, № 5(146), с. 12–15, 2020.
3. В. Р. Миколайчук, “Використання функціонального поля для оцінювання ефективності роботизованої системи моніторингу місцевості”, *Зв'язок*, т. 149, № 1, с. 62–65, 2021.
4. O. Pliushch, R. Mykolaichuk, V. Mykolaichuk, O. Makhovych, A. Mykolaichuk та O. Kravchenko, “Text information source modeling for learning monitoring system”, *2021 IEEE 3rd Int. Conf. Adv. Trends Inf. Theory (ATIT)*, № 3, с. 154–158, 2021.
5. Миколайчук В. Р., Миколайчук Р. А., Сторчак К. П. Використання засобів машинного навчання для моделювання середовища протидії між роботизованими засобами та об'єктами моніторингу. *Звя'зок*. 2023. № 5(165). С. 31–36.
6. Р. А. Миколайчук, В. Р. Миколайчук, “Використання методів навчання з підкріпленням для розробки моделі роботизованого засобу моніторингу інтелектуальних динамічних об'єктів”, Сучасні інформ. технології у сфері безпеки та оборони, т. 48, № 3, 2023.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

7. “Artificial inteligence in educational environment”, у Полілог культур: освіт. і культурол. аспекти, Чернігів, Україна, 9 квіт. 2019. Чернігів: Нац. ун-т «Черніг. колегіум» ім. Т.Г. Шевченка, 2019, с. 66–67.

8. Р. А. Миколайчук, К. П. Сторчак та В. Р. Миколайчук, “Оцінка ефективності роботизованих систем моніторінгу місцевості”, у Int. Scientific Practical Conf. «Integration new knowl., res. innov. across europe», Київ, Україна, 25 квіт. 2020. Київ: Держ. ун-т телекомунікацій, 2020.

9. В. Р. Миколайчук, “Моніторинг місцевості роботизованою системою з використанням комп’ютерного зору”, у I всеукр. науково-техн. конф. «технол. горизонти: Дослідж. та застосування інформ. технологій для технол. прогресу україни і світу», Київ, Україна, 28 листоп. 2023. Київ: Держ. ун-т інформ.-комунікац. технологій, 2023.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 16
ВСТУП 17
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ ПРАКТИЧНИХ ТА НАУКОВО-МЕТОДОЛОГІЧНИХ ПІДХОДІВ ЩОДО МОНІТОРИНГУ МІСЦЕВОСТІ 22
1.1 Системи моніторингу 22
1.1.1 Структура системи моніторингу місцевості. 24
1.1.2 Сенсорні мережі як складова частина СММ 25
1.1.3 Історичний аналіз розвитку сенсорних мереж 27
1.1.4 Поняття та класифікація об’єктів моніторингу 29
1.1.5 Роботизовані засоби у системах моніторингу місцевості 31
1.2.1 Методика порівняння з реальними даними 33
1.2.2 Методика порівняння з синтетичними даними 34
1.2.3 Методика внутрішньої оцінки 36
1.3. Постановка завдання та мети дослідження 39
1.4 Висновки до розділу 41
РОЗДІЛ 2 АНАЛІЗ ТЕОРЕТИЧНИХ ПІДХОДІВ ДО ВИКОРИСТАННЯ МЕТОДІВ RL ДЛЯ ОЦІНКИ ЕФЕКТИВНОСТІ РСММ 42
2.1. Теоретичні основи оцінки ефективності РСММ з використанням методів машинного навчання 42
2.2. Методи навчання з підкріпленням RL та їх застосування для оцінки ефективності РСММ 48
2.2.1 Введення в Навчання з Підкріпленням 51
2.2.2 Основні Алгоритми і Моделі Навчання з Підкріпленням 55
2.2.3 Методи Оптимізації в Навчанні з Підкріпленням 57
2.3. Моделювання середовища та рухомих об'єктів 62
2.4 Оцінка ефективності за допомогою RL 66
2.5 Висновки до розділу 67

РОЗДІЛ 3 РОЗРОБКА МЕТОДИКИ ОЦІНКИ ЕФЕКТИВНОСТІ РОБОТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ МІСЦЕВОСТІ 69
3.1. Обґрунтування показників та критеріїв оцінки ефективності РСММ 69
3.2. Розробка моделі середовища моніторингу 78
3.3. Розробка моделі роботизованого засобу та моделі переміщення об’єкта пошуку 86
3.4 Розробка методики оцінки ефективності 94
3.5 Висновки до розділу 110
РОЗДІЛ 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА МЕТОДИКИ ОЦІНКИ ЕФЕКТИВНОСТІ СИСТЕМИ МОНІТОРИНГУ НА ОСНОВІ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ 112
4.1. Архітектура фреймворку методики 112
4.2 Вибір технологій, які використовуються у фреймворку 119
4.3. Опис модулів фреймворку 121
4.4. Перевірка адекватності розробленої методики за допомогою контрольованих експериментів та порівняння з існуючими методами 131
4.4. Висновки до розділу 141
ВИСНОВКИ 143
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 146
ДОДАТОК А 158
ДОДАТОК Б 169

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

СММ	– системи моніторингу місцевості
РСММ	– роботизовані системи моніторингу місцевості
RL	– (Reinforcement Learning) навчання з підкріпленням
CV	– (Computer vision) комп'ютерний зір
CNN	– (Convolutional neural network) згорткова нейронна мережа
ГІС	– Геоінформаційні системи
IoT	– (Internet of Things) Інтернет речей
WSN	– (Wireless Sensor Network) бездротові сенсорні мережі
MDPs	– (Markov Decision Processes) процес прийняття рішень Маркова
РЗ	– роботизовані засоби
ОМ	– об’єкт моніторингу
TF-agents	– агенти TensorFlow
API	– (Application Programming Interface) прикладний програмний інтерфейс
IDE	– (Integrated development environment) інтегроване середовище розробки

ВСТУП

Дисертаційне дослідження присвячено аналізу систем моніторингу місцевості та їх ефективності у виявленні та ідентифікації рухомих об’єктів. Ці системи відіграють важливу роль у сучасних умовах, коли потреба у високоточному моніторингу територій стає все більш актуальною.

Актуальність дослідження систем моніторингу місцевості (СММ), особливо у контексті виявлення та ідентифікації рухомих об’єктів, можна обґрунтувати наступним чином:

1. *Зростання технологічних можливостей*: Сучасний світ переживає бурхливий розвиток технологій, що веде до появи нових засобів моніторингу та контролю місцевості. Це створює потребу у вдосконаленні існуючих систем та розробці нових методів аналізу.

2. *Безпека та оборона*: У контексті глобальної безпеки та оборони здатність швидко виявляти та ідентифікувати рухомі об’єкти може бути критично важливою. Це може стосуватися як військових операцій, так і мирного моніторингу кордонів.

3. *Екологічний моніторинг*: Зміни клімату, незаконне вирубування лісів - це лише декілька з проблем, які можна виявляти та контролювати за допомогою ефективних систем моніторингу місцевості.

4. *Міське планування та розвиток*: У міських умовах системи моніторингу місцевості можуть використовуватися для моніторингу транспортних потоків, виявлення аварійних ситуацій, контролю за будівництвом та іншими аспектами міського життя.

5. *Вдосконалення технологій*: Як було вказано у дослідженні, існуючі сенсорні мережі можуть бути неефективними у певних умовах. Це підкреслює потребу у додаткових дослідженнях та розробці нових підходів.

6. *Економічний аспект*: Ефективні системи моніторингу можуть призвести до зниження витрат, оптимізації ресурсів та підвищення продуктивності у ряді секторів економіки.

Отже, актуальність дослідження систем моніторингу місцевості, зокрема у контексті виявлення та ідентифікації рухомих об’єктів, очевидна. Вони відіграють ключову роль у багатьох аспектах сучасного життя, від оборони до екології, і потребують постійного вдосконалення та адаптації до змінюваних умов.

Аналіз практичних підходів до систем моніторингу місцевості виявив, що більшість сучасних систем базуються на сенсорних мережах. Ці мережі дозволяють збирати детальну інформацію про місцевість, але мають певні обмеження. Основна проблема полягає у виявленні та ідентифікації рухомих об’єктів, особливо тих, які намагаються уникнути ідентифікації. У таких випадках сенсорні мережі можуть бути не ефективними.

Для підвищення ефективності систем моніторингу місцевості у виявленні рухомих об’єктів пропонується використовувати роботизовані засоби. Ці засоби можуть включати в себе різноманітні дрони, роботи-землероби та інші автоматизовані системи, які здатні відстежувати рухомі об’єкти на місцевості та ідентифікувати їх.

Аналіз теоретичних підходів до оцінки ефективності роботизованих систем моніторингу місцевості (РСММ) показав, що основна проблема полягає у моделюванні динаміки ідентифікації рухомих об’єктів. Це ускладнює процес оцінки, адже потрібно враховувати численні фактори: швидкість руху об’єкта, його розміри, форму, колір та інші характеристики, які можуть впливати на його видимість для сенсорів.

Для підвищення ефективності систем моніторингу місцевості необхідно комбінувати сенсорні мережі з роботизованими засобами. Такий підхід дозволить забезпечити високу точність виявлення та ідентифікації рухомих об’єктів, незалежно від їх спроб уникнути ідентифікації. Також необхідно розробити нові методики моделювання динаміки ідентифікації, щоб точніше оцінювати ефективність роботизованих СММ.

Об’єкт дослідження - процес розробки роботизованої системи моніторингу місцевості.

Предмет дослідження - методи та алгоритми оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання.

Мета дослідження полягає у розробці та впроваджені методики оцінки ефективності роботизованої системи моніторингу місцевості, яка базується на принципах машинного навчання, з використанням моделювання за допомогою навчання з підкріпленням.

Наукові завдання:

1. Обґрунтувати показники та критерії оцінки ефективності РСММ.
2. Розробити модель середовища моніторингу з урахуванням роботизованих засобів та рухомих об'єктів.
3. Розробити модель роботизованого засобу (що включає розробку алгоритму формування множини можливих дій, які система може виконувати для ідентифікації об'єкта та модель вибору поведінки).
4. Розробити методику оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання, а саме навчання з підкріпленням.

Методи дослідження: аналіз, моделювання, експериментальні дослідження, методи машинного навчання, RL моделювання.

Наукова новизна: розроблена методика дозволяє отримати об’єктивну оцінку ефективності роботизованої системи моніторингу місцевості, враховуючи особливості машинного навчання та RL моделювання, що дає можливість оптимізувати стратегії моніторингу, з урахуванням адаптації до змінюваних умов середовища та інтелектуальної динамічності об’єктів моніторингу, що сприяє підвищенню загальної ефективності РСММ.

Для досягнення поставленої мети дисертаційної роботи були вирішені такі завдання:

1. Удосконалено модель середовища моніторингу, яка враховує протидію роботизованих засобів та рухомих об’єктів та за допомогою використання згорткових нейронних мереж аналізує середовище, що дало змогу підвищити адаптивність системи до змінних умов та дало можливість детально відтворити

реальні умови, в яких діють роботизовані засоби, включаючи перешкоди, зони видимості та різні типи поверхонь.

2. Удосконалено модель роботизованого засобу, що включає розробку алгоритму формування множини можливих дій, які система може виконувати для ідентифікації об’єкта та модель вибору поведінки, використано техніки глибинного навчання для автоматичного вивчення оптимальних стратегій взаємодії з навколишнім середовищем, що дало можливість враховувати взаємодію з іншими роботизованими засобами та об'єктами моніторингу.

3. Вперше розроблено методику оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання, яка включає розроблену модель навчання з підкріпленням для точного визначення характеристик об'єкта моніторингу та роботизованого засобу, що забезпечило високу адаптивність та ефективність порівняно з існуючими методиками, особливо в складних динамічних умовах.

Особистий внесок здобувача. Основні положення та результати дисертаційної роботи отримані авторкою самостійно. Авторка виконала усі теоретичні та практичні дослідження, що становлять основу дисертаційної роботи.

В опублікованих роботах у співавторстві, згідно списку опублікованих праць за темою дисертації (с.12-13), здобувачці належать такі результати: [1] – проаналізовано та виявлено основні показники надійності роботизованих засобів, а саме БПЛА (безпілотних літальних апаратів); [2] – проаналізовано теоретичні підходи до використання засобів штучного інтелекту, а саме методів машинного навчання; [4] – запропоновані методи підвищення ефективності системи моніторингу; [5] – розроблено модель середовища протидії між роботизованими засобами та об'єктами моніторингу; [6] – розроблено модель роботизованого засобу для моніторингу інтелектуальних динамічних об'єктів.

Апробація матеріалів дисертації

Основні результати дисертаційної роботи доповідалися і обговорювалися на 3 науково-технічних та науково-практичних конференціях:

Полілог культур: освітній і культурологічний аспекти Всеукраїнська науково-практична конференція молодих науковців та студентів 9 квітня 2019 р., м. Чернігів.

І науково-практична конференція «Integration of new knowledge, research and innovation across Europe», 23-25 квітня 2020 р., м. Київ

І Всеукраїнська науково-технічна конференція «технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу», 28 листопада 2023 р., м. Київ.

Структура та обсяг дисертації : дисертація складається з анотації, змісту, переліку умовних скорочень вступу, чотирьох розділів, загальних висновків, списку використаних джерел та додатків і має 130 сторінок основного тексту, 5 рисунків та таблиць, 15 сторінок додатків. Список використаних джерел містить 110 найменувань і займає 12 сторінок. Загальний обсяг дисертаційної роботи – 172 сторінка.

Практична цінність: результати дослідження можуть бути використані при проектуванні та впровадженні роботизованих систем моніторингу місцевості в різних галузях промисловості.

Застосування результатів роботи. Результати наукових досліджень були використані на кафедрі інженерії програмного забезпечення автоматизованих систем навчально-наукового інституту інформаційних технологій Державного університету інформаційно-комунікаційних технологій під час виконання науково-дослідної роботи на тему «Дослідження обробки траєкторної інформації в вимірювально-обчислювальних системах» (ДУТ, м.Київ), впроваджені у виробничий процес на підприємстві ТОВ «ХУАВЕЙ Україна», та в навчальному процесі Державного університету інформаційно-комунікаційних технологій (Київ).

РОЗДІЛ 1
АНАЛІЗ ІСНУЮЧИХ ПРАКТИЧНИХ ТА НАУКОВО-МЕТОДОЛОГІЧНИХ ПІДХОДІВ ЩОДО МОНІТОРИНГУ МІСЦЕВОСТІ

1.1 Системи моніторингу

Моніторинг місцевості можна охарактеризувати як комплексну систему спостереження, що збирає інформацію для подальшого аналізу та прогнозування змін стану досліджуваного середовища. Моніторинг зазвичай зосереджується на процесах, наприклад, у разі спостереженні за територією, що охороняється, завданням моніторингу є — визначити коли й де відбуваються дії, хто їх здійснює, а в разі моніторингу довкілля — спостерігати за станом біосфери, визначати ступінь антропогенного впливу на довкілля, визначати чинники та джерела впливу.

Спостереження за об’єктами і явищами дає змогу оцінити поточну ситуацію й передбачити подальші зміни, спрощуючи процес прийняття рішень для запобігання небажаних станів або ситуацій. Моніторинг застосовується в різноманітних сферах людської діяльності, таких як: картографія, геодезія, транспорт, агрокультура, нафтова промисловість та багато інших [97, 25].

Так у [87] досліджується проблема точності цифрових моделей місцевості для моніторингу залізничних колій та прилеглої території. Запропоновано технічне рішення, яке включало аерофотозйомку та лазерне сканування з використанням безпілотних літальних апаратів. Цей метод дозволяє підвищити точність визначення траєкторії польоту та отримання координат точок на земній поверхні. Результатом була створена точна цифрова модель, придатна для діагностики стану залізничної колії. Це ефективне технічне рішення для поліпшення точності цифрових моделей для моніторингу залізничних колій з використанням безпілотних літальних апаратів.

Однією з галузей моніторингу є — безпека лісового господарства. У роботі [109] запропоновано вдосконалені методи моніторингу безпеки лісового господарства за допомогою технології комп'ютерного зору (CV) та згорткової

нейронної мережі (CNN). Використано CV для обробки топографічних карт та прогнозування поширення бездротового сигналу. Проведені експерименти показали високу точність та ефективність прогнозування дифракційних втрат і кореляційної відстані. Роботизовані системи можуть додатково оптимізувати моніторинг, дозволяючи переміщати камери в реальному часі до потенційних зон пожеж залежно від зібраної інформації. Ці результати вказують на високий потенціал використання CV та CNN у вдосконаленні систем моніторингу безпеки лісового господарства.

Ефективний моніторинг місцевості базується на точних і надійних методах визначення розташування об'єктів у просторі та часі. Нижче розглянуті теоретичні основи та підходи до визначення ефективності у сфері моніторингу місцевості:

1. Геодезичні та геоінформаційні методи:
   1) Глобальні навігаційні спутникові системи (GNSS) [52, 94, 110]: Використання супутникових систем, таких як GPS, Galileo та BeiDou для точного визначення географічного положення об'єктів на землі. Вплив розташування супутників, множинні відбиття сигналу та інші аспекти на точність моніторингу.
   2) Лідар та радар: Використання лазерних (лідар) [70] та радіохвильових (радар) систем [27] для отримання точних геопросторових slamданих про поверхню землі та об'єкти на ній. Врахування висоти, густини та інших параметрів для визначення ефективності.
2. Техніки комп'ютерного зору та обробки зображень
   1) Системи визначення розташування об'єктів ((Simultaneous localization and mapping) SLAM) [35]: Техніки, які дозволяють обчислювати розташування та орієнтацію об'єктів в реальному часі на основі аналізу відеозображень та інших сенсорів.
   2) Відстеження об'єктів [18]: Використання алгоритмів відстеження об'єктів у відео потоці для моніторингу їхнього руху та розташування в просторі.
3. Моделювання та аналіз даних

1) Геоінформаційні системи (ГІС) [6]: Використання ГІС для збору, зберігання, аналізу та візуалізації географічних даних, що дозволяє вивчати зміни в місцевості з часом.
2) Математичне моделювання [50]: Використання математичних моделей для прогнозування та аналізу динаміки змін в місцевості на основі наявних даних.

4. Застосування штучного інтелекту [8] та машинного навчання

1) Класифікація та розпізнавання зразків [16]: Використання алгоритмів машинного навчання для класифікації та розпізнавання об'єктів на зображеннях, що полегшує автоматизований аналіз місцевості.
2) Прогнозування з використанням нейронних мереж [30]: використання нейронних мереж для прогнозування змін в місцевості на основі історичних даних та сучасних спостережень.

Ці теоретичні основи та підходи стають основою для розвитку нових методів та технологій у сфері моніторингу місцевості. Ефективний моніторинг дозволяє не лише збирати дані, а й аналізувати їх з точністю та надійністю, що є ключем до успішного використання цих даних у важливих сферах, таких як дослідження клімату, цивільний захист, розробка транспортних систем та багато інших.

1.1.1 Структура системи моніторингу місцевості.

СММ можуть мати складну структуру, яка включає в себе різноманітні компоненти та технології для збору, обробки та аналізу даних. Ось загальна структура такої системи:

1. Сенсори та спостереження: сенсори (температурні, вологості, руху, звуку, акустичні сенсори тощо);камери та відеокамери (відслідковують рух об’єктів та реєструють візуальні дані); Лідар [71], Радар та інші дистанційні спостереження (вимірюють відстані, висоти та створюють 3D-моделі місцевості) [58].

2. Збір та трансмісія даних: безпровідна технологія (Wi-Fi, Bluetooth, LoRa, NB-IoT (Internet of Things) Інтернет речей тощо); супутникові засоби

трансмісії даних (використовуються для передачі даних з віддалених або важкодоступних областей).

3. Централізована система збору та зберігання даних: центральний сервер або хмарне сховище (зберігає та обробляє отримані дані); бази даних (використовуються для зберігання структурованих даних, таких як координати, температура, вологість тощо).

4. Обробка та аналіз даних: системи обробки сигналів(використовуються для фільтрації та обробки сигналів від сенсорів та спостережень); аналітичні інструменти (включають в себе алгоритми відслідковування об’єктів, аналіз змін у середовищі, прогнозування поведінки тощо).

5. Відображення та повідомлення: Веб-інтерфейси та додатки (дозволяють користувачам відслідковувати дані в реальному часі, переглядати звіти та взаємодіяти з системою); системи візуалізації (відображають дані у вигляді карт, графіків та іншої графіки для легшого сприйняття); повідомлення та автоматичні сповіщення (системи можуть автоматично повідомляти користувачів про події або стан системи).

6. Керування та автоматизація: системи управління (дозволяють налаштовувати параметри системи та відстежувати її стан); автоматизовані системи реагування (використовують алгоритми для автоматичного реагування на певні події, такі як виявлення аварійних ситуацій або незвичайних змін у середовищі).

1.1.2 Сенсорні мережі як складова частина СММ

Основу сучасних СММ складають сенсорні мережі (або мережі сенсорів). Сенсорні мережі представляють собою розгалужену систему, яка включає в себе сенсори пов’язані бездротовими технологіями, розташовані у конкретній області або навколо об'єкта моніторингу. Ці сенсори здатні вимірювати різні фізичні параметри, такі як температура, вологість, тиск, звук, рух, світло, рівень газу, а також інші величини, що характеризують оточуюче середовище.

У роботі [7] розглянута розробка системи моніторингу на основі бездротових сенсорних мереж. Основна мета дослідження полягала у впровадженні екологічного моніторингу за допомогою безпровідних сенсорних мереж на пасовищах. Досліджено бездротові сенсорні мережі та їх роль у екологічному моніторингу, описано взаємодію бездротової мережі і протоколів передачі даних. Експериментально визначено максимальні відстані передачі параметрів (температури та вологості) з урахуванням перешкод. Розроблено алгоритми маршрутизації та вивчено передачу даних від датчиків до сервера. Розроблено наземну систему моніторингу на базі технології Zigbee, здатну приймати нові вузли. Досліджено застосування безпровідних мереж у бджільництві, вибрана Mesh-топологія та створені вузли з використанням датчиків Arduino та модулів ZigBee XBee від компанії Digi. У результаті дослідження виявлено нестабільність роботи вузлів XBee з Wi-Fi маршрутизаторами та в умовах перешкод.

У статті [1] розглядається актуальна проблема забезпечення ефективного та безпечного використання повітряного простору пілотованою та безпілотною авіацією в умовах обмеження застосування систем навігаційного забезпечення польотів та моніторингу об’єктів аеродромної інфраструктури. У статті обґрунтовано модель системи моніторингу та навігаційного забезпечення польотів на основі інтеграції сенсорної мережі та ГІС. Система функціонує за допомогою мережі розподілених у просторі сенсорів та бездротової технології далекого радіусу дії LoRa. Ефективність моделі перевірена шляхом математичного моделювання та експериментальних досліджень. Модель системи моніторингу району аеродрому універсальна для застосування в різних фізико-географічних умовах. Виправлення параметрів інтегрованої системи моніторингу в умовах радіоелектронного впливу здійснюється шляхом розширення атрибутів ГІС щодо масивів місцевості та перешкод, які виступають елементами бази даних аеронавігаційної інформації.

У роботі [102] розглядається метод синтезу сенсорних систем та безпілотних літальних апаратів для моніторингу атмосферних параметрів та

виявлення грозової активності. Метод базується на застосуванні прогностичних моделей, розрахунку індексу конвекції атмосфери та інтеграції сенсорних систем, безпілотних апаратів та бездротових технологій зв'язку. Результати експериментальних досліджень підтверджують ефективність методу. Отримані дані можуть бути використані для створення локальних та комплексних систем моніторингу грозової активності в повітряному просторі, забезпечуючи безпечність та ефективність авіаційних операцій.

З досліджень видно, що сенсорні мережі використовуються для збору даних у реальному часі з навколишнього середовища та аналізу цих даних для різних цілей. Сенсорні мережі можна використовувати для моніторингу місцевості в режимі реального часу та дистанційного моніторингу, що робить їх важливим інструментом вивчення та збереження навколишнього середовища в рамках сучасних наукових досліджень.

1.1.3 Історичний аналіз розвитку сенсорних мереж

Історія розвитку сенсорних мереж є еволюційною та динамічною, відзначається ключовими подіями та етапами [108]:

1. Початковий етап (1980-1990-ті роки): Перші дослідження в галузі сенсорних мереж почалися у 1980-1990-х роках. У цей період було здійснено базові кроки у розробці бездротових сенсорних систем та комунікаційних протоколів для обміну даними між сенсорами.

2. Етап стандартизації та розширення (2000-2010-ті роки): У 2000-2010-ті роки настала ера стандартизації протоколів та алгоритмів для сенсорних мереж. З'явилися різні стандарти, такі як Zigbee, Bluetooth та Wi-Fi, що дозволило розширити область застосування сенсорних мереж у різних галузях.

3. Розвиток енергоефективності та мініатюризації (2010-2023): Сучасні дослідження спрямовані на поліпшення енергоефективності сенсорних мереж та їх мініатюризацію. Розроблені нові техніки збереження енергії та створення мініатюрних сенсорів, які можна використовувати в різних умовах.

4. Розширені застосування (Сучасність): Сенсорні мережі зараз застосовуються у багатьох галузях, включаючи екологію, метеорологію,

медицину, агрокультуру, моніторинг будівель, управління міським середовищем та інші галузі.

Цей процес постійного розвитку дозволяє сенсорним мережам стати потужним інструментом для збору та аналізу даних у реальному часі в різних галузях, вирішуючи важливі завдання моніторингу та управління навколишнім середовищем.

З цього можна виділити наступну інформацію щодо історії розвитку сенсорних мереж:

1. Еволюція архітектури IoT: Історична еволюція архітектури IoT почалася з моменту появи перших реалізацій. Початкові архітектури були зосереджені на локальних сценаріях без глобального зв’язку через Інтернет. З часом архітектури стали більш глобальними та стандартизованими. Величезне поширення гаджетів та потреба підключення великої кількості сенсорів зробили «хмару» стандартом де-факто. Швидке поширення гаджетів, мобільних пристроїв та цифрових рішень для різних проблем призвело до хаосу в «Інтернеті Речей». Багато компаній пропонують вертикальні архітектури, які виключають можливість взаємозв’язку різних пристроїв. Це створює проблему «уособленності», де кожен виробник створює свої рішення, не думаючи про сумісність з іншими системами.

2. Розвиток технологій та сенсорів: Вдосконалення технологій дозволило створювати більш складні системи для оптимізації споживання енергоносіїв. Нові технологічні досягнення включають менші, міцніші та потужніші сенсори. Це дозволяє вбудовувати їх у різноманітні пристрої та використовувати в небезпечних для людей областях. Основна увага приділяється енергоефективності, оскільки більшість пристроїв «Інтернету Речей» використовують бездротовий зв’язок і повинні мати автономні джерела живлення.

3. Бездротові сенсорні мережі (WSN) [104]: WSN стали основою для систем «Інтернету Речей». Поява IoT підвищила користь WSN до нового рівня. WSN можуть бути однорідними або неоднорідними. Однорідні WSN

складаються з сенсорів з однаковими характеристиками, тоді як неоднорідні мережі включають сенсори з різними чутливими елементами та протоколами зв'язку.

Сенсорні мережі дозволяють збирати детальну інформацію про місцевість, відстежувати різноманітні параметри та реагувати на зміни в реальному часі. Однак, як показує практика, сенсорні мережі можуть бути неефективними у випадках, коли необхідно виявляти та ідентифікувати рухомі об'єкти, особливо якщо ці об'єкти намагаються уникнути ідентифікації.

1.1.4 Поняття та класифікація об'єктів моніторингу

Об'єкт моніторингу (ОМ) можна визначити як, конкретний об'єкт, який піддається спостереженню, вимірюванням або вивченню з метою отримання даних щодо його стану, параметрів чи змін. Об'єкти моніторингу можуть бути різноманітні та включати в себе різні області, такі як природні середовища, будівлі, інфраструктура, технічні системи, кліматичні явища, люди тощо.

Об'єкти моніторингу визначаються в контексті конкретної задачі моніторингу та можуть бути різних розмірів, стаціонарними або рухомими, природними або штучними.

Об'єкти моніторингу можуть бути визначені як статичні, такі як будівлі або природні резервуари, або динамічні, такі як рухомі транспортні засоби чи люди. Важливо визначити об'єкти моніторингу для точного збору даних та відстеження їхнього стану або змін у часі.

Класифікація за рухомістю є важливою з точки зору планування та реалізації систем моніторингу і дозволяє визначити оптимальні стратегії спостережень та вимірювань для кожного типу об'єктів моніторингу. Таким чином нерухомі об'єкти мають такі характеристики:

1. Фіксованість - нерухомі об'єкти залишаються на одному місці та не зміщуються.

2. Статичність - ці об'єкти можуть мати сталі характеристики у часі або змінюватися повільно.

Приклади нерухомих об’єктів: будівлі та інфраструктура (будівлі, мости, дамби), природні об’єкти (гори, річки, ліси), метеорологічні станції, сейсмічні датчики.

Рухомі об’єкти моніторингу, в свою чергу мають такі характеристики:

1. Рухливість - рухомі об’єкти можуть мати постійну рухливість (наприклад, транспортні засоби), або рухатися періодично (наприклад, люди, тварини) [9];

2. Простір руху - рухомі об’єкти можуть переміщатися в обмеженому просторі (наприклад, в місті) або на великі відстані (наприклад, повітряний транспорт, який подолає сотні чи тисячі кілометрів).

Прикладами можуть бути: транспортні засоби (автомобілі, автобуси, поїзди транспортний літак, корабель, дрон) [68], живі організми (люди, тварини, птахи) [59].

Рухомі об’єкти моніторингу можуть мати різні характеристики траєкторії руху, які класифікуються як хаотичні, встановлені та інтелектуальні траєкторії. Ця класифікація визначає специфічний спосіб, якими об’єкти переміщаються у просторі, що може мати значущий вплив на методи моніторингу та прогнозування їхньої поведінки.

1. Хаотична траєкторія руху - характеризується непередбачуваним та незвичайним змішаним рухом, який не слідує жодним законам або сталим схемам. Має такі характеристики: незвичайний, неординарний рух; непередбачувана поведінка; випадкові зміни напрямку та швидкості.

2. Встановлені траєкторії руху - передбачає сталий, неперервний та повторюваний рух по певній траєкторії. Має такі характеристики: сталість руху вздовж певної лінії або шляху; повторюваність руху з невеликими відхиленнями; регулярна схема руху.

3. Інтелектуальна траєкторія руху - характеризується здатністю об’єкта адаптувати свій рух залежно від зовнішніх умов та власних навичок або навчання. Має такі характеристики: здатність взяти до уваги навколишнє середовище та адаптувати рух до умов навколишнього світу; вміння навчатися

та змінювати стратегії руху на основі набутого досвіду; здатність до прийняття рішень та реакції на непередбачені ситуації. Прикладами можуть бути автономні роботи, які вміють уникати перешкод та адаптувати свій рух до змінних умов навколишнього середовища, а також тварини, які вивчають нові шляхи для досягнення мети та адаптують свій рух для отримання їжі чи уникнення небезпеки.

Ці характеристики траєкторій руху визначаються фізичними властивостями об’єктів та їхнім взаємодією з навколишнім середовищем. З цього випливає необхідність інтеграції роботизованих засобів у СММ. Роботизовані засоби, такі як дрони або автономні роботи, можуть значно підвищити ефективність системи, дозволяючи швидко реагувати на зміни та відстежувати рухомі об’єкти.

1.1.5 Роботизовані засоби у системах моніторингу місцевості

Роботизовані системи моніторингу можуть бути оснащені різними типами сенсорів, такими як камери, лідари, радари, GPS-приймачі тощо, і мати здатність працювати в автономному режимі або під керуванням оператора, що дозволяє їм збирати дані в реальному часі та в реальних умовах. Нижче наведені способи як роботизовані системи можуть бути використані в сфері моніторингу місцевості з використанням різних технічних аспектів:

1. Автономні роботи для точного збору даних:
   1) Автономні дрони: використання безпілотних літальних апаратів (дронів) для високоточного зйомки з повітря. Дрони можуть бути оснащені камерами, лідаром або іншими сенсорами для вимірювання рельєфу місцевості та виявлення об'єктів [101].
   2) Мобільні роботи на колесах чи гусеницях: роботи, які можуть рухатися по нерівному ґрунту, оснащені різними сенсорами, що забезпечують збір даних про поверхню та об'єкти навколо [69].
2. Системи візуального спостереження та обробки даних:

1) Роботи з камерами та лідарами: роботи з камерами та лідарами можуть виявляти та відстежувати об'єкти, створюючи точні 3D-моделі місцевості [37].
2) Аналіз зображень з використанням машинного навчання: використання алгоритмів машинного навчання для аналізу великих обсягів візуальних даних. Це може включати в себе розпізнавання об'єктів, класифікацію та відстеження руху [90].

3. Геодезичні та геоінформаційні аспекти:
   1) Інтеграція з геоінформаційними системами: роботизовані системи можуть бути інтегровані з геоінформаційними системами для збору та аналізу даних в реальному часі, створюючи динамічні карти місцевості [45].
   2) Точність та калібрування: правильна калібрування сенсорів та вимірювання точності важливі для забезпечення точності та надійності отриманих даних.
4. Інтеграція з іншими системами:
   1) Системи збору та збереження даних: роботизовані системи можуть бути інтегровані з системами збору та збереження даних для автоматичного архівування та аналізу отриманих даних.
   2) Керування та моніторинг у віддаленому режимі: можливість керування та моніторингу роботизованими системами в реальному часі з використанням віддалених систем дозволяє вдосконалити процес моніторингу та реагувати на зміни негайно [77].

Роботизовані системи у сфері моніторингу місцевості відіграють ключову роль, спрощуючи та автоматизуючи процес збору даних. Вони не лише полегшують рутинну роботу, але й забезпечують високу точність, ефективність та безпеку у процесі збору та обробки інформації.

Інтеграція роботизованих систем дозволяє здійснювати високоточний та систематичний моніторинг, що є ключовим для здобуття нових знань та виявлення закономірностей у вивченні навколишнього середовища. Цей підхід

сприяє науковим відкриттям, оптимізації технічних рішень та впровадженню нових технологій у практичні застосування.

1.2 Методики оцінки ефективності

Оцінка ефективності вимагає визначення того, наскільки добре система відповідає поставленим цілям, виконує свої функції та сприяє досягненню визначених результатів. Наразі існує кілька методик оцінки ефективності роботизованих систем моніторингу, які використовуються для визначення їхньої точності, надійності та продуктивності. Однак кожен метод має свої переваги та недоліки, які важливо враховувати при виборі підходящої методики для конкретної задачі моніторингу місцевості.

1.2.1 Методика порівняння з реальними даними

Методика порівняння з реальними даними (ground truth comparison) [53] використовується для оцінки ефективності роботизованих систем моніторингу шляхом порівняння їхніх вимірів або результатів з реальними даними, які вважаються точними та надійним. Ця методика базується на порівнянні вимірювань або даних, зібраних роботизованою системою, з відомими реальними значеннями.

Кроки методики порівняння з реальними даними:

1. Взаємодія з іншими системами: RL дозволяє PCMM ефективно взаємодіяти з іншими системами, адаптуючи свою поведінку на основі взаємодії та отриманих відгуків. Це може бути корисним у складних сценаріях, де декілька систем моніторингу працюють разом.
2. Збір даних роботизованою системою: роботизована система проводить моніторинг і збирає власні дані відносно того ж самого об’єкта або явища, що і реальні дані.
3. Порівняння та аналіз: отримані дані роботизованої системи порівнюються з реальними даними. Це може включати в себе порівняння вимірів, розташування об’єктів, руху чи інших параметрів.

4. Оцінка точності та надійності: на основі порівняння роботизовані дані оцінюються з точки зору їхньої точності, надійності та відповідності реальним даним.

Перевагою методики порівняння з реальними даними є її об’єктивність, оскільки використання реальних даних надає віддзеркалення об'єктивної картини в процесі оцінки ефективності роботизованої системи. Точність цього методу може забезпечити високу точність оцінки, особливо якщо реальні дані є відомими за своєю точністю.

Однак варто враховувати його недоліки, зокрема складність збору реальних даних, що може вимагати значних затрат за часом та ресурсами, особливо у випадку вивчення великих територій чи ускладнених умов. Крім того, обмежена вибірка даних може вплинути на загальну репрезентативність результатів, а також залежність від умов, оскільки отримані результати можуть бути відображенням лише умов, в яких зібрані реальні дані, і, отже, можуть бути не універсальними для різних умов застосування роботизованої системи.

Методика порівняння з реальними даними є однією з найбільш точних та об’єктивних методик оцінки ефективності роботизованих систем моніторингу. Однак вона вимагає уважного планування, збору та обробки реальних даних для забезпечення надійності та вірогідності результатів оцінки.

1.2.2 Методика порівняння з синтетичними даними

Методика порівняння з синтетичними даними (synthetic data comparison) [31] використовується для оцінки ефективності роботизованих систем моніторингу шляхом порівняння їхніх вимірювань або результатів із синтетичними даними, які генеруються математичними моделями чи комп’ютерними симуляціями. Ця методика використовує вигадані дані, які відтворюють характеристики реальних об’єктів чи явищ, і порівнює вимірювання роботизованої системи з цими синтетичними даними.

Кроки методики порівняння з синтетичними даними:

1. Генерація синтетичних даних: синтетичні дані генеруються за допомогою математичних моделей, комп’ютерних симуляцій або інших методів, що відтворюють реальні умови чи об’єкти моніторингу.

2. Збір даних роботизованою системою: роботизована система проводить моніторинг і збирає власні дані відносно того ж самого об’єкта чи явища.

3. Порівняння та аналіз: отримані дані роботизованої системи порівнюються зі згенерованими синтетичними даними. Це може включати в себе порівняння вимірювань, розташування об’єктів, руху чи інших параметрів.

4. Оцінка точності та надійності: на основі порівняння синтетичних та реальних даних оцінюється точність, надійність та відповідність роботизованих даних синтетичним даним.

Методика порівняння з синтетичними даними має свої переваги. По-перше, ця методика дозволяє працювати в контрольованих умовах, де синтетичні дані можна створити в умовах, які можна систематично контролювати. Це надає можливість тестувати роботизовану систему в різних сценаріях, відтворюючи різноманітні умови та випробування, що допомагає в оцінці її продуктивності. По-друге, в порівнянні зі збором реальних даних, цей метод може бути ефективнішим та менш витратним, сприяючи більш швидкому та зручному аналізу системи.

Незважаючи на ці переваги, існують і недоліки методики порівняння з синтетичними даними. По-перше, створення ідеальних синтетичних даних може бути важким завданням через можливі неточності у використовуваних моделях. Ці неточності можуть вплинути на достовірність результатів та відповідність синтетичних даних реальним умовам. По-друге, синтетичні дані можуть не відображати усі реальні умови моніторингу, що може обмежити їхню застосовність у практичних сценаріях вивчення та аналізу місцевості.

Методика порівняння з синтетичними даними є корисним інструментом для тестування роботизованих систем моніторингу в контрольованих умовах. Вона дозволяє вирішувати питання ефективності та надійності системи без

необхідності збору складних або дорогих реальних даних. Однак, щоб бути корисною, синтетична модель повинна відображати реальні умови та об’єкти як найточніше.

1.2.3 Методика внутрішньої оцінки

Методика внутрішньої оцінки (internal evaluation) [33] використовується для оцінки ефективності роботизованих систем моніторингу, враховуючи внутрішні параметри та характеристики самої системи, без порівняння її з реальними чи синтетичними даними. Ця методика полягає у внутрішньому аналізі даних, зібраних роботизованою системою, та оцінці їхньої якості та точності на основі внутрішніх метрик та показників ефективності.

Кроки методики внутрішньої оцінки:

1. Збір та запис даних: роботизована система збирає дані під час моніторингу, включаючи різноманітні параметри, такі як відстань, швидкість, кут нахилу, обсяг об’єктів тощо.

2. Аналіз та вимірювання внутрішніх метрик: внутрішні метрики оцінюють якість та точність зібраних даних. це може включати в себе аналіз точності вимірювань, стабільність системи під час руху, швидкість збору даних, час реакції системи тощо.

3. Внутрішні тести та сценарії: внутрішні тести можуть бути розроблені для оцінки реакції системи на різні сценарії чи умови моніторингу. Наприклад, якість вимірювань у різних погодних умовах або на різних типах поверхні.

4. Аналіз надійності та відомостей про відновлення: оцінка надійності системи та її здатності до відновлення після втрати зв’язку чи інших перешкод.

Методика внутрішньої оцінки має свої переваги, зокрема, в її автономності. Цей підхід дозволяє системі самостійно проводити оцінку, не залежачи від зовнішніх утручань чи додаткових даних, що робить його зручним та ефективним. Крім того, метод внутрішньої оцінки надає деталізовану інформацію про внутрішні параметри та характеристики системи, що може бути корисною для подальшого її вдосконалення та оптимізації.

Однак існують певні недоліки цієї методики. По-перше, результати оцінки можуть бути обмежені об’єктивністю, оскільки сама система оцінює свою ефективність, що може викликати внутрішні перекоси та спотворення в результатах. По-друге, цей метод чутливий до змін у внутрішніх параметрах системи та умов експлуатації, що може впливати на надійність та достовірність отриманих даних.

Методика внутрішньої оцінки є швидким і зручним способом для внутрішньої перевірки системи на предмет її ефективності та надійності. однак вона повинна супроводжуватися іншими методиками оцінки, такими як порівняння з реальними чи синтетичними даними, для забезпечення більш повного розуміння ефективності системи в реальних умовах.

Крім того, можна розглянути інші методики, які спеціально розроблені для оцінки ефективності систем моніторингу місцевості, такі як методика оцінки якості зображення (Image quality assessment, IQA) та методика оцінки точності геопросторових даних (Accuracy assessment of spatial data, AASD) [51, 64]. Ці методики дозволяють оцінити якість та точність зображень та геопросторових даних, що отримуються в результаті моніторингу місцевості.

Оцінка ефективності роботизованих СММ є складним завданням, що вимагає глибокого наукового підходу. Однією з основних проблем є моделювання динаміки ідентифікації рухомих об’єктів. Це ускладнює процес оцінки, адже потрібно враховувати численні фактори: швидкість руху об’єкта, його розміри, форму, колір та інші характеристики.

Для розв’язання цієї проблеми вчені та інженери звертаються до сучасних методів машинного навчання. Ці методи дозволяють створювати алгоритми, які можуть адаптуватися до змінюваних умов та вдосконалювати свою роботу на основі накопиченого досвіду.

Аналіз існуючої теорії оцінки ефективності дав змогу зробити висновок щодо ускладнення її використання для оцінки ефективності роботизованих систем моніторингу. Це пов’язано передусім із неможливістю урахування складної структури роботизованої системи та її динамічного характеру,

залежності ефективності елементів системи від їх просторово-часового розподілу та відповідного розподілу об’єктів моніторингу. Таким чином виникає необхідність розробити математичні моделі структури роботизованої системи моніторингу та її функціонування, що в свою чергу, дасть змогу забезпечити удосконалення існуючих методик оцінки ефективності систем моніторингу, на основі відповідних показників та критеріїв.

Reinforcement Learning або навчання з підкріпленням є одним з передових методів машинного навчання [54], який знаходить застосування у численних сферах, включаючи робототехніку [85] та системи моніторингу [28]. В контексті РСММ, RL може відігравати ключову роль у підвищенні їх ефективності.

1. Адаптивність до змінюваних умов**:** Однією з основних переваг RL є його здатність адаптуватися до нових умов без необхідності повторного навчання з нуля. Це особливо важливо для РСММ, які можуть працювати в різних умовах та стикаються з різноманітними викликами.

2. Оптимізація стратегій моніторингу**:** За допомогою RL, РСММ можуть вивчати та оптимізувати свої стратегії моніторингу, вибираючи найкращі траєкторії руху, точки спостереження та інші параметри для максимізації ефективності виявлення та ідентифікації об’єктів.

3. Взаємодія з іншими системами**:** RL дозволяє РСММ ефективно взаємодіяти з іншими системами, адаптуючи свою поведінку на основі взаємодії та отриманих відгуків. Це може бути корисним у складних сценаріях, де декілька систем моніторингу працюють разом [42].

4. Моделювання динаміки ідентифікації**:** Як було вказано раніше, моделювання динаміки ідентифікації рухомих об’єктів є викликом. RL може допомогти у цьому, навчаючись прогнозувати поведінку об’єктів та адаптуючи стратегії моніторингу відповідно до цього.

5. Неперервне навчання та вдосконалення**:** Оскільки умови моніторингу можуть змінюватися, RL дозволяє системам постійно вдосконалюватися, адаптуючись до нових викликів та збираючи новий досвід.

У підсумку, методи RL можуть значно підвищити ефективність РСММ, дозволяючи їм адаптуватися до змінюваних умов, оптимізувати свої стратегії та взаємодіяти з іншими системами. Вони представляють великий потенціал для подальших досліджень та розробок у цій сфері.

1.3. Постановка завдання та мети дослідження

Роботизовані системи моніторингу місцевості, які базуються на штучному інтелекті та машинному навчанні, здатні забезпечити високий рівень точності та автономності у зборі та обробці даних. Однак, розробка ефективних методик оцінки функціональності та надійності цих систем залишається важливою та актуальною проблемою.

Мета цього дослідження полягає в розробці та впровадженні методики оцінки ефективності роботизованої системи моніторингу місцевості, яка базується на принципах машинного навчання, з використанням моделювання за допомогою RL.

Для створення методики оцінки ефективності за допомогою моделювання ідентифікації рухомого об'єкту з використанням RL, необхідно вирішити такі наукові завдання:

1. Обґрунтувати показники та критерії оцінки ефективності РСММ.
2. Розробити модель середовища моніторингу з урахуванням роботизованих засобів та рухомих об'єктів.
3. Розробити модель роботизованого засобу (що включає розробку алгоритму формування множини можливих дій, які система може виконувати для ідентифікації об'єкта та модель вибору поведінки)
4. Розробити методику оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання а саме RL.

Таким чином щоб досягнути поставленої мети необхідно виконати такі кроки:

1. Розробка методики визначення станів та дій РСММ: Визначення параметрів, які входитимуть до стану системи для використання у RL, зокрема, координати об’єктів, їхню швидкість, орієнтацію тощо. Розробка набору можливих дій, які РСММ може виконувати для ідентифікації об’єктів, включаючи зміну траєкторії руху, збільшення/зменшення зближення тощо.

2. Розробка функції нагороди: Створення функції, яка визначатиме ефективність дій агента RL в конкретному стані. Ця функція повинна відображати цілі системи моніторингу місцевості.

3. Вибір алгоритму RL: Визначення найбільш підходящого алгоритму RL (наприклад, Q-learning, Deep Q Networks, Policy Gradient Methods тощо) для вирішення даної задачі.

4. Створення симуляційного середовища: Розробка або вибір існуючого симуляційного середовища, де можна тренувати агента RL перед його застосуванням у реальних умовах.

5. Тренування та валідація моделі: Проведення тренування агента в симуляційному середовищі, збирання даних про його ефективність та результатів. Валідація моделі на реальних або більш складних сценаріях для перевірки її здатності до загальної адаптації.

6. Оптимізація та налаштування: На основі отриманих результатів, внесення корективів у модель, алгоритм або параметри навчання для підвищення ефективності системи моніторингу.

Реалізація цих завдань дозволить розробити та оптимізувати ефективну методику оцінки роботизованих систем моніторингу місцевості на основі машинного навчання з використанням підходу Reinforcement Learning, що відкриє можливості для подальших наукових досліджень та практичних застосувань у сфері моніторингу та аналізу місцевості.

1.4 Висновки до розділу

Перший розділ дослідження присвячено аналізу практичних та науково-методологічних підходів до роботизованих систем моніторингу місцевості. У розділі вивчені та проаналізовані основні аспекти, пов’язані з системами моніторингу місцевості. Дослідження ретельно висвітлює структуру СММ та розглядає ключову їх складову - сенсорні мережі, які є важливим елементом у цьому контексті. За допомогою історичного аналізу розвитку сенсорних мереж відзначено їх важливість у розвитку і впровадженні систем моніторингу місцевості. Було встановлено, що сенсорні мережі, хоча і є ефективними у багатьох випадках, можуть мати обмеження у виявленні рухомих об’єктів. Роботизовані засоби, у поєднанні з методами машинного навчання, можуть стати відповіддю на ці виклики, проте потребують подальших наукових досліджень для оптимізації їх роботи.

У розділі представлена систематизація та класифікація СММ, а також введене поняття об’єкту моніторингу. Особлива увага приділяється рухомим об’єктам моніторингу, де детально розглядаються різні траєкторії їхнього руху, включаючи хаотичні, встановлені та інтелектуальні траєкторії. У розділі проведено аналіз методів ідентифікації рухомих об’єктів системою моніторингу місцевості.

У розділі висвітлено методики оцінки ефективності СММ, зокрема з фокусом на моніторинг рухомих об’єктів, і відзначено недоліки та обмеження існуючих методик. Це призводить до висунення вимог до моделювання взаємодії між роботизованими засобами та об’єктами моніторингу. Для забезпечення оптимальної взаємодії висвітлено можливість застосування методів машинного навчання, зокрема навчання з підкріпленням. Цей підхід не лише автоматизує процеси в системах моніторингу місцевості, але й підвищує їх ефективність та точність, роблячи їх більш гнучкими та адаптивними до різних умов та завдань.

РОЗДІЛ 2
АНАЛІЗ ТЕОРЕТИЧНИХ ПІДХОДІВ ДО ВИКОРИСТАННЯ МЕТОДІВ RL ДЛЯ ОЦІНКИ ЕФЕКТИВНОСТІ РСММ

2.1. Теоретичні основи оцінки ефективності РСММ з використанням методів машинного навчання

РСММ знаходять все більше застосування у різноманітних сферах [84], від екологічного нагляду [86] до безпеки та оборони [60]. Ефективність цих систем є критично важливою для гарантування точності, надійності та економічності виконуваних ними задач. У цьому контексті, методи машинного навчання відіграють ключову роль у підвищенні ефективності РСММ. Для оцінки ефективності РСММ важливо визначити відповідні метрики, що дозволяють кількісно оцінювати їх продуктивність та оптимізацію. Серед ключових метрик, для оцінки ефективності, можна використовувати:

1. Точність моніторингу - ця метрика вимірює, наскільки точно система виявляє та ідентифікує об'єкти або події в області моніторингу. Вона може бути виражена через відсоток правильно ідентифікованих об'єктів або подій до загальної кількості подій, які відбулися.

$$P = \frac{TP}{N_p} 100\%, \tag{2.1}$$

де P – точність; TP -кількість правильно ідентифікованих подій, а N_p – загальна кількість подій.

2. Ефективність використання ресурсів - ця метрика оцінює, наскільки ефективно РСММ використовує свої ресурси, такі як енергія, час або обладнання. Це може бути виражено як відношення досягнутих результатів до витрачених ресурсів.

$$E_{rs} = \frac{R_a}{C_{rs}}, \tag{2.2}$$

де E_{rs} – ефективність використання ресурсів; R_a – досягнуті результати; C_{rs} – витрачені ресурси

3. Час реакції: важливо, як швидко система реагує на виявлені зміни або події у місцевості. Ця метрика може вимірювати час між виявленням події та відповіддю на неї.

$$t_r = t_v - t_d , \tag{2.3}$$

де t_r – час реакції; t_v – час відповіді; t_d – час виявлення.

4. Обхват моніторингу: визначає, наскільки широкою або всебічною є зона моніторингу. Може бути виражено як відсоток від загальної площі, яку необхідно моніторити.

$$O = \frac{A_m}{A_t} 100\%, \tag{2.4}$$

де t_r – Обхват; t_v – Площа моніторингу; t_d – Загальна площа для моніторингу.

5. Адаптивність: відображає здатність системи адаптуватися до змін умов або вимог. Цю метрику можна визначити як кількість успішних адаптацій до нових сценаріїв у певний період.

$$A = \frac{S_a}{T}, \tag{2.5}$$

де A - адаптивність; S_a - кількість успішних адаптацій; T - період часу.

Ці метрики дозволяють глибше аналізувати та оцінювати ефективність РСММ, що є критично важливим для оптимізації їх роботи та впровадження ускладнень, які можуть виникати у реальних умовах.

Теоретичні підходи до оцінки ефективності РСММ, математично можна формалізувати включаючи такі аспекти:

1. Огляд існуючих теорій та методик:

Основні метрики оцінки можна формалізувати за допомогою визначення ключових метрик, таких як точність P, повнота R, і F-міра F, які використовуються для оцінки якості моніторингу.

$$F = 2\,\frac{P \times R}{P + R} \tag{2.6}$$

Моделі оцінки продуктивності використовують математичні моделі, такі як лінійні моделі [3] або більш складні статистичні моделі [19], для оцінки продуктивності РСММ.

2. Аналіз застосування та обмежень:

Можуть використовуватись моделі оптимізації ресурсів, що передбачає використання математичних моделей для оптимізації розподілу ресурсів, наприклад, за допомогою лінійного програмування. Мінімізувати $\sum_{i-1}^{n} c_i x_i$ за умов $\sum_{i-1}^{n} a_{ji} x_i \geq b_j\,, x_i \geq 0$

Також існують моделі поведінкової аналітики, де застосовуються стохастичні моделі або Марківські процеси для аналізу та прогнозування поведінки РСММ у різних умовах.

$$P(X_{n+1} = x | X_1 = x_1, \dots, X_n = x_n) = P(X_{n+1} = x | X_n = x_n) \tag{2.7}$$

Ці математичні формалізації дозволяють глибше зрозуміти існуючі теоретичні підходи та методології оцінки ефективності РСММ. Вони також допомагають виявити потенційні обмеження цих підходів і напрямки для подальших досліджень.

РСММ використовують різні технології, включаючи машинне навчання, для забезпечення ефективного та точного моніторингу оточуючого середовища.

Машинне навчання - це галузь штучного інтелекту, яка займається розробкою алгоритмів і моделей, які навчаються на основі даних і можуть виконувати завдання без явно заданої програми [98]. Замість того, щоб

програмувати комп'ютери напряму для виконання конкретних завдань, в машинному навчанні використовуються статистичні методи, які дають можливість системам самостійно навчатися і вдосконалюватися на основі досвіду.

Існує кілька основних категорій або типів машинного навчання [67], в залежності від того, як відбувається процес навчання та як взаємодіють дані і моделі. Основні типи машинного навчання включають в себе:

1. Навчання з вчителем (Supervised Learning) [40] - у цьому типі навчання модель навчається на основі пари “вхід-вихід”. Модель отримує набір даних, де для кожного вхідного значення є відоме відповідне вихідне значення. Мета полягає в тому, щоб навчити модель передбачати вихідні значення для нових, раніше невідомих вхідних даних.

2. Навчання без вчителя (Unsupervised Learning) [34] - у цьому типі навчання модель отримує тільки вхідні дані і намагається знайти внутрішні закономірності або структури у даних. Немає відомих вихідних значень, і модель самостійно намагається виявити корисні властивості або кластери в даних.

3. Навчання з частковим наглядом (Semi-supervised Learning) [22] - цей тип навчання використовує як дані з відомими вихідними значеннями, так і дані без відомих вихідних значень. Він використовує невеликий набір даних з відомими вихідними значеннями разом із великим набором даних без відомих вихідних значень для покращення точності моделі.

4. Навчання з підкріпленням (Reinforcement Learning (RL)) [95] - у цьому типі навчання агент взаємодіє з динамічною середовищем і вчиться вибирати оптимальні дії для максимізації нагороди чи мінімізації покарання. Агент отримує відгук від середовища на кожну дію, яку він виконує, і вчиться оптимальним стратегіям на основі цього відгуку.

5. Навчання з передачею знань (Transfer Learning) [75] - цей підтип включає в себе використання знань, набутих від розв’язання одного завдання, для поліпшення продуктивності в іншому, схожому завданні.

Кожен з цих типів машинного навчання має свої особливості і використовується в різних ситуаціях залежно від характеру даних і задачі, яку потрібно вирішити. Застосування машинного навчання дозволяє автоматизувати процес оцінки, роблячи його більш об'єктивним, швидшим та точнішим. Проте важливо мати на увазі, що правильний вибір моделей, навчання на відповідних та репрезентативних даних, а також ретельний аналіз результатів, є ключовими етапами для успішного впровадження машинного навчання у методики оцінки ефективності роботизованих систем моніторингу. Навчання з підкріпленням може бути потужним інструментом для розробки роботизованих систем моніторингу, оскільки воно дозволяє роботам навчатися і вдосконалювати свої навички в реальному часі, роблячи їх більш ефективними та адаптивними до змінних умов.

У випадку PCMM, RL може бути використано для автоматизації та оптимізації процесів моніторингу, дозволяючи системам адаптуватися до змінних умов місцевості та вимог завдань. Це може включати підвищення точності виявлення об'єктів, оптимізацію шляхів моніторингу, адаптацію до змін у навколишньому середовищі та ефективне реагування на несподівані події.

Оптимізація рішень у контексті навчання з підкріпленням може бути математично сформульована через концепцію цільової функції [47], яка визначає стратегію (політику) агента для максимізації сумарної очікуваної винагороди з плином часу.

Формально, це виглядає так:

1 Стратегія (політика) агента π : функція $\pi(a \mid s)$ визначає ймовірність вибору дії a в стані s.

2 Винагорода R : функція винагороди $R(s, a)$ визначає винагороду, яку агент отримує за виконання дії a в стані s.

3 Цільова функція (функція значення): функція значення $V^{\pi}(s)$ представляє сумарну очікувану винагороду, яку агент може отримати, діючи згідно з політикою π з початкового стану s.

Вона розраховується як:

$$V^{\pi}(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^{t} R(s_t, a_t) \mid s_0 = s, \pi], \quad (2.8)$$

де γ - коефіцієнт зниження (discount factor) у діапазоні [0, 1], що визначає важливість майбутніх винагород.

4. Оптимізація політики: Задача оптимізації полягає у виборі такої політики π^*, яка максимізує цільову функцію для всіх станів:

$$\pi^* = \arg \max_{\pi} V^{\pi}(s) \quad (2.9)$$

5 Функція дії-значення Q : Альтернативний підхід полягає у використанні функції дії-значення $Q^{\pi}(s, a)$, яка оцінює очікувану винагороду за вибір дії a в стані s та подальше дотримання політики π.

$$Q^{\pi}(s, a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^{t} R(s_t, a_t) \mid s_0 = s_{\downarrow} a_0 = a, \pi] \quad (2.10)$$

Оптимізація полягає у знаходженні політики π такої, що максимізує $Q^{\pi}(s, a)$ для всіх s та a [95].

У випадку РСММ, ці математичні принципи можуть бути застосовані для вибору оптимальних маршрутів моніторингу, забезпечуючи максимально ефективне використання ресурсів системи.

Інтеграція алгоритмів RL з роботизованими системами є складним завданням, яке вимагає систематичного підходу для забезпечення ефективної взаємодії та оптимальної функціональності. Розглянемо ключові аспекти цього процесу.

У першу чергу, основи інтеграції передбачають ретельне розуміння інтерфейсів, що дозволяє RL алгоритмам ефективно взаємодіяти з різними компонентами роботизованих систем, включаючи датчики та актуатори. Також

важливо забезпечити сумісність програмного забезпечення RL алгоритмів із існуючими системами керування роботами.

Другий аспект стосується взаємодії з датчиками. Використання датчиків для збору інформації про навколишнє середовище та обробка цих сигналів для подальшого використання в RL алгоритмах є важливим етапом.

Керування актуаторами включає розробку механізмів для перекладу висновків RL алгоритмів у конкретні команди для актуаторів. Важливо враховувати фізичні обмеження актуаторів, щоб уникнути можливих пошкоджень.

Зворотний зв'язок та адаптація, як четвертий етап, визначають використання інформації з датчиків для забезпечення зворотного зв'язку алгоритмам та можливість неперервного навчання для оптимізації алгоритмів на основі реальних даних.

Безпека та надійність, як п'ятий аспект, передбачають визначення меж безпеки, тестування міцності та впровадження заходів для автоматичного виявлення та запобігання потенційно небезпечним діям.

Застосування RL у РСММ відкриває можливості для створення більш інтелектуальних, систем, що самовдосконалюються, які можуть забезпечити вищий рівень автономії та ефективності. Це, в свою чергу, може значно знизити вартість та часові витрати на моніторинг, а також підвищити його якість і точність.

2.2. Методи навчання з підкріпленням RL та їх застосування для оцінки ефективності РСММ

РСММ знаходять все більше застосування у різноманітних сферах, від екологічного нагляду до безпеки та оборони. Ефективність цих систем є критично важливою для гарантування точності, надійності та економічності виконуваних ними задач. У цьому контексті, методи навчання з підкріпленням RL відіграють ключову роль у підвищенні ефективності РСММ.

Ідентифікація основних цілей та завдань для системи моніторингу, яка використовує методи RL, є невід'ємною частиною процесу її оптимізації. Зазначені цілі визначаються з урахуванням важливих аспектів, що включають в себе виявлення об'єктів, моніторинг змін та збір специфічної інформації.

Спрямованість на виявлення об'єктів передбачає можливість системи розпізнавати різні типи об'єктів, такі як транспортні засоби, люди чи зміни в природному середовищі. Це має важливе застосування в контексті безпеки, екологічного моніторингу та управління земельними ресурсами. Додатково, система може здійснювати збір специфічної інформації, такої як дані про погоду, якість повітря та інші екологічні показники.

Формулювання завдань для RL включає розробку різних сценаріїв, в межах яких система може тренуватися для ефективної реакції на реальні умови. Оптимізація поведінки системи визначається налаштуванням RL алгоритмів з метою максимізації ключових показників ефективності, таких як точність виявлення об'єктів та швидкість реагування.

Встановлення показників ефективності включає оцінку таких характеристик, як точність виявлення, швидкість реакції та адаптивність системи до нових умов чи непередбачуваних ситуацій. Це сприяє об'єктивній оцінці продуктивності системи моніторингу з використанням методів RL.

RL, є одним із підходів машинного навчання, де агент навчається приймати оптимальні рішення шляхом взаємодії з середовищем [55]. У випадку PCMM, RL може бути використано для автоматизації та оптимізації процесів моніторингу, дозволяючи системам адаптуватися до змінних умов місцевості та вимог завдань. Це може включати підвищення точності виявлення об'єктів, оптимізацію шляхів патрулювання, адаптацію до змін у навколишньому середовищі та ефективне реагування на несподівані події [24].

Застосування RL у PCMM відкриває можливості для створення більш інтелектуальних систем, які можуть забезпечити вищий рівень автономії та ефективності. Це, в свою чергу, може значно знизити вартість та часові витрати на моніторинг, а також підвищити його якість і точність. Однак, використання

RL також ставить перед дослідниками нові виклики, зокрема щодо стабільності, безпеки та етичних аспектів управління автономними системами.

Методи навчання з підкріпленням відіграють ключову роль у підвищенні ефективності РСММ з наступних причин:

1. Адаптація до змінних умов: РСММ часто оперують у динамічних та непередбачуваних середовищах. RL дозволяє системам адаптуватися до змінних умов, навчаючись з досвіду. Наприклад, система може навчитися виявляти нові види об'єктів або адаптуватися до змін у ландшафті [103].

2. Оптимізація рішень: RL орієнтований на визначення оптимальних стратегій дій для досягнення максимальної винагороди (або мінімізації втрат). Це означає, що РСММ можуть ефективніше використовувати свої ресурси, наприклад, вибираючи найкращі маршрути для моніторингу [46].

3. Самонавчання та автономність: за допомогою RL, РСММ можуть самостійно вчитися без необхідності безперервного людського втручання. Це забезпечує високий рівень автономності та можливість безперервної оптимізації процесів моніторингу [28].

4. Ефективне реагування на непередбачувані події: у місцевостях, де можуть виникати непередбачувані події (наприклад, природні катастрофи або безпекові інциденти), RL дозволяє РСММ швидко адаптуватися та змінювати тактику моніторингу залежно від ситуації [80].

5. Підвищення точності та надійності: RL може допомогти у вдосконаленні алгоритмів виявлення та класифікації об'єктів, підвищуючи точність та надійність системи. Це особливо важливо для завдань, які вимагають високої точності, як-от екологічний моніторинг або безпека [32].

6. Зниження Витрат: Завдяки підвищеній ефективності та автономності, РСММ, які використовують RL, можуть знизити загальні витрати на моніторинг та обслуговування, оскільки потребують менше людського втручання та ресурсів.

У сукупності, RL надає PCMM можливість стати більш інтелектуальними, адаптивними та ефективними, що є вирішальним для виконання широкого спектру завдань моніторингу у сучасному світі [21].

Оптимізація рішень у контексті навчання з підкріпленням може бути математично сформульована через концепцію цільової функції, яка визначає стратегію (політику) агента для максимізації сумарної очікуваної винагороди з плином часу.

2.2.1 Введення в Навчання з Підкріпленням

RL є одним з ключових напрямків машинного навчання, де агенти навчаються приймати оптимальні рішення, коли взаємодіють з середовищем.

Історія навчання з підкріпленням включає в себе довгий еволюційний шлях, який об’єднав ідеї теорії ігор, оптимального контролю, статистичного навчання та комп’ютерного моделювання. Початкові кроки в цьому напрямі були пов’язані з розвитком теорії ігор та оптимального контролю.

1. Теорія ігор та оптимальний контроль: Вже в середині 20-го століття роботи в галузі теорії ігор та оптимального контролю визначили теоретичні засади прийняття рішень в умовах невизначеності. Розвиток ідей, таких як стратегії гравців і оптимальний контроль, зробив важливий внесок у теорію управління та визначення оптимальних стратегій в різних сценаріях взаємодії [5].

2. Статистичне навчання: З виникненням комп’ютерів та збільшення обчислювальних можливостей виникла можливість застосування статистичних методів до задач вибіркового навчання. Підходи, засновані на ідеях регресії та класифікації, почали використовуватися для вирішення проблем, де агент має навчитися вибирати дії на основі спостережень [17].

3. Комп’ютерне моделювання: З розвитком комп’ютерної техніки з’явилася можливість створювати більш складні та реалістичні моделі середовищ, з якими взаємодіє агент. Це дозволило проводити експерименти та виконувати великі обчислювальні завдання для вдосконалення стратегій агента в реальному часі [2].

4. З’явлення RL та алгоритмів Q-навчання: У 80-х роках минулого століття концепція RL отримала визнання завдяки роботам вчених, таких як Крістофер Воткінс та Джеррі Тесауро. Вони розробили алгоритми Q-навчання, що дозволили агентам вивчати оптимальні стратегії в умовах невизначеності [105].

5. Поглиблення нейронних мереж: У останні десятиліття із зростанням потужності обчислювальних ресурсів та популярністю глибокого навчання важливу роль в RL відіграли глибокі нейронні мережі. Вони дозволили агентам ефективно вивчати складні залежності великої кількості даних [4].

Навчання з підкріпленням - це процес, в якому агент вчиться приймати рішення, виконуючи дії у середовищі, щоб максимізувати деяку нагороду. Математично, цей процес може бути представлений як пошук оптимальної політики π у визначеному просторі станів S та дій A, яка максимізує кумулятивну винагороду R [95].

Система RL складається з таких основних елементів:

1. Агент: Визначається як сутність, що взаємодіє з середовищем, приймаючи рішення на основі спостережень та винагород.

2. Середовище (Environment): Комплекс факторів та обставин, в якому агент виконує дії.

3. Стан (State): Опис поточної ситуації в середовищі, що впливає на рішення агента.

4. Дія (Action): Конкретне втручання агента у середовище, що може змінити стан.

5. Політика або стратегія (Policy): Стратегія або правило, за яким агент обирає дії в залежності від стану.

6. Винагорода (Reward): Значення, яке агент отримує за виконання певних дій. Мета агента - максимізація кумулятивної винагороди.

Система RL включає в себе кілька ключових концепцій, які допомагають в розумінні та вдосконаленні процесу прийняття рішень агентом в невизначеному середовищі:

1. Модель (Model): Модель може бути використана для моделювання взаємодії агента із середовищем. Вона може передбачати, які стани та винагороди будуть в майбутньому, що допомагає агентові приймати кращі рішення.

2. Траєкторія (Trajectory): Це послідовність станів, дій і винагород, які агент спостерігає протягом певного періоду часу. Агент вивчає на основі траєкторій, які допомагають йому краще адаптуватися до середовища.

3. Функція цінності (Value Function): Ця функція оцінює очікувану кумулятивну винагороду в конкретному стані або при конкретній дії. Вона допомагає агентові оцінити, наскільки добре він виконується у певному стані.

4. Функція передачі (Transition Function): Ця функція визначає ймовірність переходу в новий стан після виконання певної дії в певному стані. Вона є ключовою для моделювання динаміки середовища.

5. Дисконтований фактор (Discount Factor): Цей фактор визначає, наскільки агент враховує майбутні винагороди. Дисконтований фактор може бути використаний для надання переваги більш раннім винагородам порівняно зі збільшенням винагород у майбутньому.

6. Дослідження та експлуатація (Exploration and Exploitation): Агент повинен збалансувати свої дії між вивчанням нового (дослідженням) і використанням вже вивченого (експлуатація), щоб максимізувати загальну винагороду.

7. Функція стратегії (Policy Function): Ця функція визначає, як агент обирає дії в залежності від стану. Вона може бути визначена як детермінована стратегія або ймовірнісна стратегія.

Ці концепції разом формують основні складові системи навчання з підкріпленням і допомагають агентові досягати своєї мети - максимізації кумулятивної винагороди в невизначеному середовищі.

Процес прийняття рішень Маркова (Markov Decision Processes, MDPs) є математичною моделлю для формалізації задач RL. Вони надають теоретичний фреймворк для розуміння та вирішення проблем навчання з підсиленням [81].

MDP - це математична модель, яка використовується для моделювання послідовності рішень у ситуації, де результати попередніх рішень впливають на майбутні рішення та їх наслідки.

MDP складається з чотирьох компонентів:

1. Множина станів (States): це множина всіх можливих станів, в яких може перебувати система.

2. Множина можливих дій (Actions): це множина всіх можливих дій, які можна здійснювати у кожному стані.

3. Функція переходу (Transition function): це функція, яка визначає ймовірність переходу системи з одного стану в інший стан при здійсненні певної дії.

4. Функція винагороди (Reward function): це функція, яка визначає винагороду за кожну дію, що здійснюється в певному стані.

Ці компоненти використовуються для моделювання послідовності рішень. У кожному кроці рішення приймається на основі поточного стану системи та можливих дій. Результатом кожної дії є новий стан системи та відповідна винагорода.

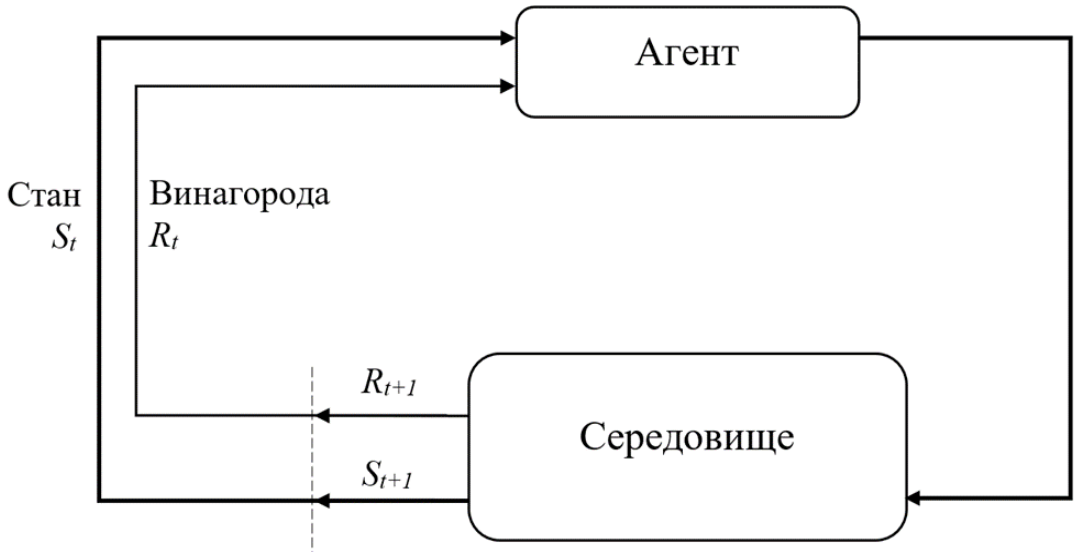

Рис.2.1 Взаємодія агента і середовища в процесі MDP

Цей процес повторюється доти, доки не досягнуто певної кінцевої точки або не було досягнуто максимального числа кроків. Метою процесу є

максимізація загальної винагороди за весь час виконання послідовності рішень [95].

2.2.2 Основні Алгоритми і Моделі Навчання з Підкріпленням

У сфері навчання з підкріпленням існує різноманітність алгоритмів, які можна розділити на категорії в залежності від підходів до вивчення середовища. Дві основні класифікації включають модельні та безмодельні підходи [95].

Модельні алгоритми полягають у створенні експліцитної моделі середовища, яка дозволяє агентові передбачати наслідки різних дій. Це призначено для оптимізації вибору дій за меншу кількість ітерацій чи спроб.

У свою чергу, безмодельні алгоритми не конструюють чіткої моделі середовища, а замість цього вони навчаються безпосередньо з досвіду, набутого в ході взаємодії з навколишнім середовищем. Такий підхід спирається на накопичений досвід агента для прийняття рішень та вирішення завдань без явної моделі, що описує середовище.

У категорії опису основних алгоритмів для задач навчання з підкріпленням можна виділити декілька ключових підходів.

Першим є Q-learning [62], що є безмодельним алгоритмом і використовує таблицю вартостей дій (Q-table) для визначення оптимальної дії в кожному конкретному стані. Основна мета цього алгоритму полягає в максимізації очікуваної винагороди. Q-learning використовує функцію цінності дії, відому як Q-функція. Q-функція оновлюється за формулою:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right), \tag{2.11}$$

де:

$Q(s, a)$ - цінність виконання дії a у стані s.

α - швидкість навчання.

r_{t+1} - винагорода, отримана після переходу у стан s_{t+1}.

γ - коефіцієнт дисконтування.

$\max_a Q(s_{t+1}, a)$ - максимальна оцінена цінність серед усіх можливих дій у наступному стані s_{t+1}.

Другий алгоритм, Deep Q-Network (DQN) [44], представляє собою розширення Q-learning за допомогою використання глибоких нейронних мереж. DQN ефективно вирішує проблеми, пов'язані з обробкою великих просторів станів і дій, що непрактично обробляти традиційним Q-learning. DQN розширює Q-learning, використовуючи глибокі нейронні мережі для апроксимації Q функції. Основне рівняння залишається тим же, але $Q(s, a)$ оцінюється за допомогою нейронної мережі.

Третій підхід - Policy Gradients - відмінний від Q-learning, оскільки цей алгоритм прямо оптимізує стратегію дій, намагаючись визначити найкращі дії для кожного стану. Це надає більшу гнучкість вибору дій у порівнянні з іншими методами. Policy Gradients оптимізує політику напряму через градієнтний підхід [65]. Політика $\pi(a \mid s)$ визначає ймовірність вибору дії a у стані s. Функція цінності політики оцінюється як:

$$J(\theta) = \mathbb{E}_{\pi_\theta}[R_t], \tag{2.12}$$

де R_t - загальна винагорода, а θ - параметри політики. Оновлення параметрів політики виконується за допомогою градієнту:

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \tag{2.13}$$

Нарешті, існують методи Actor-Critic (Актор-Критик) [43], які комбінують ідеї Policy Gradients та Q-learning. Ці методи використовують дві нейронні мережі: одна (actor) для визначення дії, а інша (critic) для оцінки цієї дії. Ця комбінація дозволяє збалансувати дослідження та експлуатацію в процесі вибору оптимальних дій в невизначеному середовищі. Методи Actor-Critic використовують дві сутності: "actor" (який вибирає дії) та "critic" (який оцінює дії). Actor оновлює свою політику за допомогою градієнтів, подібно до Policy

Gradients, а Critic оцінює цінність станів або дій (наприклад, за допомогою Q-функції).Формально, оновлення actor виглядає так:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t \mid s_t) Q^\pi(s_t, a_t), \tag{2.14}$$

де $\pi_\theta(a_t \mid s_t)$ - це політика actor, а $Q^\pi(s_t, a_t)$ - цінність, оцінена critic.

Кожен з алгоритмів демонструє свої унікальні переваги та обмеження, які визначають області їхнього практичного використання. Q-learning, який є відносно простим та ефективним для проблем із невеликим простором станів, обмежений у вирішенні складних середовищ. Другий алгоритм, Deep Q-Network (DQN), вирішує проблему великих просторів станів та дій, але вимагає значних обчислювальних ресурсів для своєї ефективної реалізації. Policy Gradients, які забезпечують більшу гнучкість у виборі дій, можуть краще справлятися з проблемами, де винагороди розподілені нерівномірно, але можуть стикатися з високою дисперсією оцінок. Нарешті, Actor-Critic Methods націлені на збалансування переваг Policy Gradients та Q-learning, але їх складно налаштовувати, і вони можуть бути нестабільними під час навчання. Отже, вибір конкретного алгоритму для вирішення задачі навчання з підсиленням вимагає уважного врахування його характеристик та властивостей в контексті конкретного завдання.

2.2.3 Методи Оптимізації в Навчанні з Підкріпленням

У стратегіях RL, агент взаємодіє з оточуючим середовищем, вчиться вибирати дії для максимізації очікуваних винагород. Проте, виникає проблема балансу між двома ключовими аспектами: експлуатацією і дослідженням [72].

Експлуатація - це стратегія вибору дій на основі вже наявних знань для максимізації негайних винагород. Агент використовує найбільш оптимальні дії, які він вже вивчив, щоб отримати максимальну винагороду в поточному стані середовища.

Дослідження - це процес вибору нових дій для отримання нових знань. Агент ризикує вибором менш відомих дій для вивчення їх ефективності та

отримання більш глибокого розуміння середовища. Це особливо важливо на початкових етапах навчання, коли агент ще не має повного розуміння середовища.

Проблема виникає у виборі оптимального балансу між цими двома стратегіями. Занадто сильне вкладення в експлуатацію може призвести до того, що агент “утримується” від дослідження нових можливостей, що може веде до стагнації та пропуску оптимальних стратегій. Навпаки, велике дослідження може призвести до того, що агент буде тратити час на вивчення неефективних дій, втрачаючи можливості отримання винагороди.

Різні методи RL, такі як epsilon-жадібність [39], Upper Confidence Bound (UCB) [66] або Thompson Sampling [99], спрямовані на розв’язання цієї проблеми. Застосування параметрів, таких як epsilon, дозволяє контролювати рівень дослідження у стратегії агента.

Забезпечення адекватного балансу між експлуатацією та дослідженням є ключовим для ефективного RL. Сучасні дослідження активно вивчають нові методи і стратегії для досягнення оптимального балансу в різних умовах та задачах. Розуміння цього аспекту допомагає поліпшити швидкість навчання та збільшити стабільність стратегій RL.

Для оптимізації RL можуть використовуватись різні засоби та методи, серед них градієнтний спуск, та вже згадане раніше Q-навчання, та методи політики.

Градієнтний спуск є основним методом оптимізації, який широко використовується в навчанні з підкріпленням. У RL, цей метод використовується для навчання параметрів стратегії агента. Сутність полягає в тому, щоб коригувати параметри стратегії так, щоб максимізувати очікувані винагороди. У контексті RL існують різні модифікації градієнтного спуску, призначені для ефективного використання в навчанні агентів. До них входять:

1. Policy Gradients: Використовує градієнт політики для покращення стратегії агента. Спрямований на безперервні та дискретні простори дій.

2. Actor-Critic методи: Поєднує градієнт політики (актор) з оціночною функцією (критик), що дозволяє оцінювати винагороду та покращувати стратегію.

Q-навчання – це метод, який вивчає функцію цінності дій в кожному стані. Використовується для оцінки та вдосконалення стратегії агента. Основна ідея полягає в навчанні Q-функції, яка визначає, наскільки вигідно вибрати конкретну дію у конкретному стані.

Методи політики включають в себе прямий і обернений підходи. Прямий підхід напряму оптимізує параметри політики, вибираючи дії, які максимізують очікувані винагороди. Обернений підхід оцінює градієнт функції корисності та використовує його для корекції політики. В реальних задачах часто використовують комбінації різних методів оптимізації. Наприклад, комбінування Q-навчання з методами політики може призвести до більш стійкої та швидкої стратегії агента.

Ці методи оптимізації застосовуються в різноманітних областях, таких як робототехніка, гральні ігри та фінанси. Аналіз та порівняння ефективності цих методів у різних контекстах дозволяє визначити їхню придатність для конкретних завдань.

Гіперпараметри [100] грають критичну роль у навчанні з підкріпленням, оскільки вони визначають параметри моделі та алгоритму навчання, які не можуть бути оптимізовані в процесі самого навчання. Це включає параметри нейромереж, крок навчання, розмір пам’яті та інші.

Різновиди гіперпараметрів:

1. Гіперпараметри нейромережі: Включають в себе кількість шарів, кількість нейронів у кожному шарі, функції активації та інші параметри, які визначають архітектуру моделі.

2. Гіперпараметри алгоритму навчання: Такі як крок навчання, кількість епох, параметри методів оптимізації та тому подібні.

3. Гіперпараметри агента в RL: Включають параметри стратегії, темп дослідження та експлуатації, функції винагороди, тощо.

Вплив гіперпараметрів на процес навчання можна визначити таким чином:

1. Швидкість навчання (learning rate): визначає, наскільки швидко агент/модель адаптується до нової інформації. Занадто великий крок може призвести до розбіжності, а занадто малий – до повільного навчання [49].

2. Архітектура мережі: оптимальний розмір та структура нейромережі можуть значно впливати на ефективність навчання. Велика модель може призвести до перенавчання, а занадто мала – до недонавчання.

3. Темп дослідження та експлуатації в RL: гіперпараметри, які контролюють баланс між експлуатацією та дослідженням, можуть суттєво впливати на траєкторію навчання.

4. Гіперпараметри оптимізації: параметри методів оптимізації, такі як момент, параметри оптимізації Adam, можуть визначити швидкість та стабільність навчання.

Гіперпараметри можна оптимізувати за допомогою таких методів:

1. Grid Search та Random Search: Методи, які систематично або випадковим чином перебирають гіперпараметри для знаходження найкращих комбінацій [38].

2. Баєсівська оптимізація: Використовує ймовірнісний підхід для пошуку гіперпараметрів, що може бути ефективним для гарячого пошуку [93].

3. Автоматична настройка гіперпараметрів: Використання алгоритмів, таких як Hyperband [20] або Bayesian Optimization [23], для автоматизованого підбору гіперпараметрів.

Правильний вибір гіперпараметрів може значно вплинути на швидкість, стабільність та якість навчання в системах підкріпленого навчання. Дослідження в галузі оптимізації гіперпараметрів є важливим напрямком, оскільки воно сприяє поліпшенню ефективності та розумінню процесів навчання з підкріпленням.

RL успішно застосовується на практиці. Практичні Приклади Успішного Застосування Reinforcement Learning включають:

1. Автономні транспортні системи: в автономних автомобілях, літаках та різноманітних транспортних системах RL використовується для навчання агентів приймати рішення в реальному часі. Компанія Waymo, колишня дочірня компанія Google, використовує RL для оптимізації стратегій водіння своїх автономних автомобілів. Агенти навчаються приймати рішення на основі вхідних даних, таких як відео потік, радар та лідар [76].

2. Ігрові стратегії: у великих комп'ютерних іграх, RL використовується для створення інтелектуальних агентів, які можуть адаптуватися до складних ігрових ситуацій. DeepMind використав RL для створення AlphaGo, який здолав світового чемпіона у грі Го. Агент вивчив оптимальні стратегії та реагував на ходи противника [107].

3. Робототехніка: в робототехніці RL використовується для навчання роботів приймати рішення та навчання новим завданням. OpenAI використовував RL для навчання робота рухатися та виконувати завдання, користуючись сенсорами та при взаємодії з навколишнім середовищем [36].

4. Фінансовий сектор: у фінансовому секторі RL використовується для розробки оптимальних стратегій в управлінні портфелем та торгівлі. Hedge-фонди використовують RL для оптимізації стратегій торгівлі та прийняття рішень у реальному часі на фондових ринках [56].

5. Управління виробництвом та логістикою: виробничі компанії використовують RL для оптимізації логістичних та виробничих процесів. RL може бути використано для оптимізації розміщення товарів на складі, визначення оптимальних маршрутів доставки та управління виробничими лініями [82].

6. Системи управління енергозабезпеченням: в системах енергозабезпечення RL використовується для оптимізації виробництва та розподілу електроенергії. В умовах відновлювальних джерел енергії RL може допомогти прогнозувати та оптимізувати використання сонячних батарей та вітряних генераторів [92].

7. Системи управління телекомунікаціями: в телекомунікаційних мережах RL використовується для управління потоками даних та оптимізації роботи мережі. RL може вирішувати завдання ресурсного розподілу, адаптуючи передачу даних в мережі в залежності від попиту та стану мережі [48].

Успішні застосування RL в різних областях свідчать про потужний потенціал цієї технології. Від автономних транспортних систем до фінансового сектору, RL допомагає створювати інтелектуальні та адаптивні системи, що призводить до покращення ефективності та розвитку нових технологій.

2.3. Моделювання середовища та рухомих об'єктів

Моделювання середовища для тренування та оцінки алгоритмів RL у контексті роботизованої системи моніторингу місцевості є критично важливим етапом. Цей детальний підхід включає ряд ключових аспектів.

По-перше, створення реалістичної моделі місцевості вимагає відтворення різноманітних ландшафтів, таких як ліси, пустелі, міські території і гірські регіони. Це сприяє адаптації системи до різних сценаріїв моніторингу. Також важливо інтегрувати елементи інфраструктури, такі як дороги, будівлі, мости тощо, що можуть впливати на функціональність системи.

Другий аспект полягає в моделюванні погодних умов. Це включає створення різноманітних кліматичних сценаріїв, таких як дощ, сніг, туман, сильний вітер тощо, що можуть вплинути на збір даних. Важливо також забезпечити можливість системи швидко адаптуватися до змін у погоді для забезпечення точності моніторингу.

Третій аспект передбачає інтеграцію зовнішніх факторів, таких як динамічні зміни в середовищі, що включають рухомі об'єкти (тварини, люди, транспортні засоби) та екологічні зміни, такі як рослинність і водойми.

Нарешті, для тестування та оптимізації системи визначаються різноманітні сценарії тестування з метою оцінки адаптивності, точності та надійності.

Оптимізація алгоритмів RL включає налаштування та тонке налаштування для забезпечення оптимальної продуктивності в різних умовах оточення.

Мета моделювання полягає в створенні віртуального представлення середовища, що підлягає моніторингу, для дослідження та тестування роботизованої системи. Основні аспекти визначення мети:

1. Дослідження ефективності моніторингу: моделювання дозволяє вивчити, наскільки добре роботизована система впорається із завданням моніторингу місцевості. Мета полягає в аналізі реакції системи на різноманітні умови та визначенні оптимальних стратегій моніторингу.

2. Вдосконалення алгоритмів Computer Vision: інтеграція системи розпізнавання об’єктів та зображень дозволяє вдосконалити алгоритми взаємодії роботизованої системи із навколишнім середовищем. Мета - забезпечити високу точність розпізнавання та забезпечити якісний збір інформації.

3. Моделювання реальних умов: мета моделювання полягає в тому, щоб наблизити віртуальне середовище до реальних умов місцевості. Це дозволяє здійснювати тестування та вдосконалення системи в контрольованому середовищі, що максимально відповідає реальності.

Для створення моделі роботизованої системи моніторингу, перш за все необхідно визначити такі аспекти:

1. Опис завдань системи: чітке визначення завдань, які передбачає виконувати роботизована система. Завдання можуть включати виявлення об’єктів, аналіз зображень, слідкування за рухом тощо.

2. Технічні характеристики робота: огляд основних технічних параметрів робота, таких як швидкість, точність рухів, час автономної роботи, апаратне забезпечення, сенсори, інші важливі характеристики, які необхідні для наближення моделювання до реальних умов.

3. Огляд використовуваних технологій: визначення ключових технологій, таких як Computer Vision, які використовуються для досягнення цілей моніторингу.

4. Контекст застосування: Зазначення того, де система моніторингу може бути застосована, наприклад, у сфері безпеки, транспорту, агропромислової та інших областях.

5. Очікувані результати: Визначення очікуваних від системи результатів моніторингу та їх значення для користувача або системи управління.

У сучасних умовах моніторингу місцевості існує ряд проблем, які обмежують ефективність та точність роботизованих засобів. Існуючі методи, такі як роботи та дрони, мають свої обмеження у точності та швидкості аналізу зображень місцевості. Традиційні підходи, що базуються на евристики та простих правилах, часто виявляються неефективними у складних умовах.

Важливими є отримання інформації про місцевість в реальному часі, особливо в сферах рятувальних операцій, безпеки та військових заходів. Традиційні методи нерідко недостатньо швидкі або не точні, що ускладнює ефективність вирішення конкретних завдань.

Аналіз зображень місцевості стає складним завданням через різноманітність її характеристик, таких як теренові висоти, водойми, ліси тощо. Це ставить завдання перед роботизованими засобами взаємодіяти ефективно з різноманітними елементами місцевості.

Для покращення системи моніторингу необхідно поліпшити точність розпізнавання об’єктів та характеристик місцевості. Роботизовані засоби повинні бути робастними до змін у місцевості та умовах експлуатації.

Важливим аспектом є розробка ефективного механізму передачі та обробки зображень місцевості з боку системи CV до роботизованого засобу. Забезпечення взаємодії між системами моніторингу та роботами є ключовим для прийняття інформованих рішень.

Нарешті, важливим є розгляд питань ефективного використання обчислювальних ресурсів та оптимізації апаратного забезпечення. Оптимізація витрат та ресурсів є важливим аспектом для забезпечення прийнятних витрат при використанні роботизованих засобів для моніторингу місцевості.

Для визначення параметрів роботизованих засобів (РЗ) відповідно до їх місцезнаходження в середовищі, можна скористатися методами CV [13] та загортковими нейронними мережами.

Система визначення параметрів РЗ відповідно до його місцезнаходження в середовищі, використовуючи методи CV та CNN, може бути реалізована через такий процес:

1. Збір зображень середовища -РЗ обладнаний камерами або іншими сенсорами для отримання зображень з навколишнього середовища.

2. Попередня обробка зображень - вхідні зображення можуть бути оброблені для покращення якості, фільтрації шуму, адаптації до освітлення тощо.

3. Визначення параметрів РЗ за допомогою CV. Використання методів CV для визначення позиції, орієнтації та інших параметрів РЗ на зображеннях. Виділення ключових ознак об'єкта та оточуючого середовища.

4.Використання CNN для аналізу зображень. Застосування згорткових нейронних мереж для визначення взаємодії між РЗ та середовищем. Навчання мережі на прикладах впливу різних частин середовища на рух РЗ.

5. Аналіз впливу середовища на РЗ. Визначення того, як кожна ділянка середовища впливає на РЗ (наприклад, обмеження або покращення видимості, зміна швидкості, маневреності тощо). Застосування ваг для важливості різних областей зображення.

6. Адаптація параметрів РЗ. Зміна параметрів РЗ в реальному часі відповідно до виявлених впливів. Можлива корекція траєкторії, швидкості, режимів роботи сенсорів тощо.

7. Спостереження за середовищем та динамічним об'єктом. Використання визначених параметрів для спостереження та взаємодії з динамічними об'єктами та оточуючим середовищем. Аналіз зображень для виявлення змін в середовищі та адаптація РЗ до цих змін.

Ця система може забезпечити РЗ ефективне сприйняття та реагування на зміни в навколишньому середовищі, використовуючи передові технології CV та CNN для обробки зображень та прийняття рішень в реальному часі.

2.4 Оцінка ефективності за допомогою RL

Навчання з підкріпленням — це підхід до машинного навчання, що полягає у навчанні агенту приймати рішення щодо послідовності виконаних дій, з урахуванням взаємодії з середовищем для досягнення максимальної винагороди. Перспективним напрямком RL є глибоке навчання з підкріпленням (Deep RL), що передбачає інтеграцію глибоких нейронних мереж для покращення здатності RL-моделей розпізнавати складні шаблони та адаптуватися до складних середовищ.

У контексті роботизованих систем моніторингу, методи RL можуть бути використані для:

1. Автономної навігації, а саме навчання робота самостійно переміщуватися в навколишньому середовищі.
2. Протидії з ОМ, що має на меті ідентифікацію та відстеження цільових об'єктів, які ухиляються.
3. Адаптації до динамічних умов, змін у середовищі та поведінки об'єктів.

Оцінка ефективності за допомогою RL в сфері моніторингу місцевості за участю роботизованих засобів може бути важливою для покращення продуктивності та точності систем [15]. Використання RL у таких системах дозволяє агентам (роботам) вчитися і вдосконалювати свої стратегії взаємодії з навколишнім середовищем на основі отриманих нагород.

Основні етапи впровадження RL в систему моніторингу та переслідування можуть виглядати наступним чином:

1. Визначення цілей системи моніторингу та переслідування.
2. Визначення області місцевості, яку необхідно вивчати.

3. Формулювання метрик ефективності для оцінки успішності системи.

4. Вибір та інтеграція різних типів сенсорів (камери, лідари, радари, тощо) для отримання інформації про навколишнє середовище.

5. Створення віртуального або фізичного середовища для тренування агента.

6. Моделювання різних сценаріїв та умов, які можуть виникнути в реальному середовищі.

7. Вибір моделі RL (наприклад, Q-learning, Deep Q Network, або actor-critic).

8. Налаштування параметрів моделі та обрання функції нагороди.

9. Проведення тренувань агента в середовищі.

10. Оптимізація стратегій агента на основі отриманої нагороди.

11. Тестування агента на нових сценаріях та в реальних умовах.

12. Оцінка ефективності системи на основі визначених метрик.

13. Якщо агент проявив ефективність у віртуальному середовищі, інтеграція його в реальні системи моніторингу.

14. Забезпечення можливості постійного навчання агента в реальному часі для адаптації до змін в середовищі.

Використання RL у таких системах може поліпшити адаптацію до змін в середовищі, забезпечити автономію в деяких сценаріях та зробити систему більш ефективною та надійною. Однак, важливо враховувати етичні та безпекові аспекти при впровадженні таких систем в реальне середовище.

2.5 Висновки до розділу

У даному розділі було проведено глибокий аналіз теоретичних підходів до використання методів RL для оцінки ефективності PCMM. Розглядаються теоретичні основи оцінки ефективності PCMM з використанням методів машинного навчання, зокрема RL.

Перш за все, розділ висвітлив методи навчання з підкріпленням RL та їх застосування в контексті PCMM. Виявлено, що ці методи можуть вигідно використовуватися для покращення ефективності систем моніторингу, дозволяючи агенту взаємодіяти з навколишнім середовищем та вдосконалювати свої стратегії на основі отриманих винагород.

Далі увага зосереджена на моделюванні середовища та рухомих об'єктів, де використовується поєднання комп'ютерного зору та агента, що навчається за допомогою RL. Це дозволяє відтворити умови моніторингу та навчання агента, що робить цей підхід дуже перспективним для застосування в реальних умовах.

Особлива увага була приділена самому процесу оцінки ефективності за допомогою RL. Виокремлені ключові аспекти та критерії, які дозволяють об'єктивно оцінювати роботу PCMM. Застосування RL в даному контексті не лише дозволяє оптимізувати стратегії моніторингу, але і забезпечує адаптацію до змінних умов, що робить систему більш гнучкою та придатною до застосування в різних областях.

Узагальнюючи, використання методів RL для оцінки ефективності PCMM є перспективним напрямком досліджень. Робота в цьому розділі спрямована на розкриття потенціалу RL у вдосконаленні систем моніторингу та відкриває шлях до подальших практичних застосувань цих методів у сучасних технологічних рішеннях.

РОЗДІЛ 3
РОЗРОБКА МЕТОДИКИ ОЦІНКИ ЕФЕКТИВНОСТІ РОБОТИЗОВАНОЇ СИСТЕМИ МОНІТОРИНГУ МІСЦЕВОСТІ

3.1. Обґрунтування показників та критеріїв оцінки ефективності РСММ

Оцінка ефективності РСММ є складним процесом, який вимагає розгляду багатьох факторів. Два ключових аспекти, що мають вирішальне значення для оцінки ефективності, - це ймовірність ідентифікації об'єктів (чи може система точно ідентифікувати об'єкти у своїй моніторинговій зоні) та час ідентифікації (швидкість, з якою система здатна виконати цю ідентифікацію). Ці показники відіграють ключову роль у визначенні загальної продуктивності системи, оскільки вони безпосередньо впливають на її здатність виконувати основні функції.

З огляду на це, розробка критерію ефективності, який максимізує мінімальне значення цих двох показників, є логічним підходом. Такий критерій забезпечує, що система не лише швидка або точна, але й забезпечує оптимальний баланс між цими двома ключовими характеристиками. Таким чином, можна гарантувати, що система буде ефективною у різноманітних умовах та для різних задач.

Для оцінки ефективності РСММ, які використовують методи RL для ідентифікації ОМ, які ухиляються від РЗ, можна використовувати такі показники ефективності:

1. Точність ідентифікації: здатність РСММ точно виявляти та ідентифікувати ОМ, які намагаються уникнути виявлення. Це включає визначення положення, швидкості та інших характеристик ОМ.

2. Час виявлення: час, який потрібно системі для виявлення ОМ після його появи у моніторинговій зоні.

3. Адаптивність до тактик ухилення: здатність РСММ адаптуватися до різних стратегій ухилення, які використовують ОМ, і ефективно відстежувати їх, навіть при зміні тактики ухилення.

4. Стратегічна гнучкість: здатність РСММ змінювати власні стратегії пошуку та ідентифікації відповідно до поведінки ОМ.

5. Енергоефективність: оцінка споживання енергії РСММ у процесі довготривалого моніторингу та відстеження ОМ, особливо в умовах постійної адаптації.

6. Стійкість до помилок: здатність системи уникати помилкових спрацьовувань та неправильних ідентифікацій при відстеженні ОМ, що ухиляються.

7. Вартість впровадження та експлуатації: економічна ефективність системи в умовах складних завдань моніторингу, включаючи витрати на розробку, встановлення, та обслуговування.

8. Діапазон дій: здатність РСММ ефективно відстежувати ОМ на великих територіях або в складних умовах моніторингу.

9. Інтеграція з іншими системами: ефективність взаємодії РСММ з іншими системами моніторингу та аналітичними інструментами.

10. Загальна надійність системи: здатність системи безперервно працювати та забезпечувати точні результати протягом тривалого часу, навіть у випадку складних завдань ідентифікації.

РСММ — це авангардні технологічні рішення, які інтегрують робототехніку, сенсорні мережі та штучний інтелект для ефективного моніторингу та аналізу місцевості. Основна мета РСММ полягає у зборі даних, ідентифікації об'єктів, виявленні аномалій та наданні оперативної інформації для прийняття рішень. Ці системи знаходять застосування у великому діапазоні сфер — від військових операцій та оборони до екологічного моніторингу та міського планування.

Ефективність РСММ загалом може бути визначена як їх здатність виконувати визначені завдання з максимальною точністю, надійністю та

ефективністю в заданих умовах. Ефективність оцінюється з урахуванням таких факторів, як швидкість реакції, точність ідентифікації, адаптивність до змінних умов, енергоефективність та вартість впровадження та експлуатації.

Основні функції РСММ, які безпосередньо впливають на їх ефективність, включають:

1. Ідентифікація об'єктів: здатність точно виявляти та ідентифікувати різні об'єкти в моніторинговій зоні, особливо ті, що намагаються уникнути виявлення.

2. Швидкість реакції: час, необхідний системі для реагування на зміни у місцевості або появи нових об'єктів.

3. Адаптивність: гнучкість системи щодо зміни тактик моніторингу в залежності від умов та поведінки об'єктів.

4. Надійність: здатність системи стабільно функціонувати та надавати точні дані протягом тривалого часу.

У контексті оцінки ефективності РСММ, *ймовірність ідентифікації* та *час ідентифікації* є особливо важливими. Ймовірність ідентифікації відображає здатність РСММ ефективно виявляти та ідентифікувати об'єкти, в тому числі ті, що намагаються уникнути виявлення. Це важливо для забезпечення точності та надійності системи. З іншого боку, час ідентифікації показує, наскільки швидко система може реагувати на зміни в місцевості, що є критичним для оперативного реагування на динамічні ситуації.

Таким чином, ймовірність і час ідентифікації не лише відображають основні функціональні можливості РСММ, а й є важливими показниками їх загальної ефективності.

Ймовірність ідентифікації та час ідентифікації є ключовими показниками для оцінки ефективності роботизованих систем моніторингу місцевості, зокрема в контексті ідентифікації рухомих об'єктів, які намагаються уникнути виявлення.

Ймовірність ідентифікації визначає спроможність РЗ виявляти та точно ідентифікувати ОМ, який намагається уникнути виявлення. Цей показник належить аналізу сенсорних та аналітичних можливостей системи, виражаючи її

здатність працювати в різноманітних умовах та адаптуватися до змін у поведінці об'єктів.

Висока ймовірність ідентифікації свідчить про надійність системи виявлення та ідентифікації об'єктів, незалежно від їх спроб уникнути виявлення. Це особливо важливо в контексті сценаріїв, де точність ідентифікації виявляється критичною, наприклад, у військових, безпекових чи екологічних дослідженнях.

Час ідентифікації відображає швидкість, з якою РЗ може досягти відстані впевненої ідентифікації об'єкта мети. Цей показник оцінює ефективність руху та навігаційних можливостей РЗ, а також його здатність швидко реагувати на зміни в поведінці ОМ.

Швидка ідентифікація є критичною в умовах, де час реакції може впливати на успіх місії або безпеку. Наприклад, у ситуаціях швидкого реагування або в умовах, де ОМ може швидко змінювати своє розташування або поведінку, здатність системи швидко ідентифікувати об'єкт є життєво важливою.

Таким чином, ймовірність і час ідентифікації є фундаментальними для оцінки ефективності РСММ, оскільки вони безпосередньо впливають на здатність системи виконувати свої ключові функції в різних сценаріях моніторингу та ідентифікації.

Ймовірність ідентифікації в контексті роботизованих систем моніторингу місцевості може бути математично виражена як функція, що залежить від різних параметрів та факторів, які впливають на здатність системи ідентифікувати об'єкт. Однак, конкретна формула може суттєво відрізнятися в залежності від специфіки системи, типів сенсорів, алгоритмів обробки даних та умов експлуатації.

Ймовірність ідентифікації $P_{\text{ідент}}$можна описати таким чином:

$$P_{\text{ідент}} = \frac{N_{\text{успіш}}}{N_{\text{загалом}}}, \tag{3.1}$$

де:

- $N_{\text{успіш}}$ - кількість успішних ідентифікацій об'єкта;
- $N_{\text{загалом}}$ - загальна кількість спроб ідентифікації.

Ця формула визначає ймовірність ідентифікації як відношення кількості успішних ідентифікацій до загальної кількості спроб. Вона є досить узагальненою і може бути доповнена або модифікована для врахування додаткових параметрів, таких як якість сенсорних даних, умови місцевості, видимість об'єкта, рівень шуму в даних, стратегії ухилення об'єкта та інші фактори, які можуть впливати на успішність ідентифікації.

Також можливо використовувати більш складні статистичні або машинно-навчальні моделі для оцінки ймовірності ідентифікації, які можуть включати багатовимірний аналіз з використанням історичних даних, моделі логістичної регресії або навіть глибоке навчання для оцінки ймовірності ідентифікації в складних умовах.

Математичне формулювання часу ідентифікації в контексті протидії між РЗ та ОМ, який намагається уникнути ідентифікації, може бути дещо складнішим, оскільки воно має враховувати динаміку руху обох сторін. Один із можливих підходів до формулювання цього часу — використання диференціальних рівнянь або систем рівнянь, що описують рух РЗ та ОМ.

Наприклад, можна визначити час ідентифікації $T_{\text{ідент}}$ як час, необхідний РЗ для наближення до ОМ на відстань, достатню для впевненої ідентифікації. Це можна виразити через наступне рівняння:

$$T_{\text{ідент}} = \int_{t_0}^{t_f} 1\,, dt \tag{3.2}$$

де t_0 — час початку спроби ідентифікації, а t_f — час, коли РЗ наближається до ОМ на відстань достатню для ідентифікації. Це інтегральне вираження фактично вимірює загальний час, протягом якого РЗ намагається ідентифікувати ОМ.

Для більш детального моделювання можна використовувати систему диференціальних рівнянь, яка враховує різні параметри, такі як швидкість та напрямок руху РЗ і ОМ, зміни в тактиці ухилення ОМ, а також можливі перешкоди в місцевості. Однак, таке моделювання вимагає глибокого аналізу конкретних умов місцевості та характеристик РЗ і ОМ.

Важливо відзначити, що час ідентифікації в реальних умовах може сильно відрізнятися в залежності від багатьох факторів, тому часто для його оцінки використовуються комп'ютерні симуляції та експериментальні дані.

Математичне формулювання критерію ефективності, який полягає у максимізації мінімального значення функції вказаних показників для РСММ, може бути представлене таким чином:

Нехай $P_{\text{ідент}}$ - це ймовірність ідентифікації, а $T_{\text{ідент}}$ - час ідентифікації. Критерій ефективності K може бути виражений як максимізація мінімального значення цих двох показників:

$$K = \max\left(\min\left(P_{\text{ідент}}, T_{\text{ідент}}\right)\right) \tag{3.3}$$

Однак, оскільки ймовірність ідентифікації та час ідентифікації вимірюються в різних одиницях та мають різну природу (ймовірність ідентифікації є безрозмірною величиною, в той час як час ідентифікації вимірюється в часових одиницях), може виникнути потреба у їх нормалізації або перетворенні до спільної шкали.

Наприклад, можна ввести вагові коефіцієнти для кожного показника, або перевести їх у відсоткове співвідношення до деякого заданого оптимального значення. Це може бути представлено так:

$$K = \max\left(\min\left(\frac{P_{\text{ідент}}}{P_{\text{опт}}}, \frac{T_{\text{опт}}}{T_{\text{ідент}}}\right)\right), \tag{3.4}$$

де $P_{\text{опт}}$ – оптимальне значення ймовірності ідентифікації; $T_{\text{опт}}$ – оптимальне значення часу ідентифікації.

Цей підхід дозволяє більш точно оцінити ефективність системи, забезпечуючи порівнянність та збалансованість показників.

Для розрахунку оптимального часу ідентифікації $T_{\text{опт}}$ для РСММ, необхідно визначити критерії, на основі яких це значення буде встановлено. Оптимальний час ідентифікації може залежати від цілого ряду факторів, включаючи:

1. *Тип місцевості та сценарій використання,* в залежності від місцевості (відкрита місцевість, міське середовище, лісиста територія тощо) та сценарію використання (військові операції, рятувальні місії, екологічний моніторинг тощо), оптимальний час ідентифікації може значно відрізнятися.

2. *Характеристики об'єктів моніторингу* такі як: розміри, швидкість, кольорова гамма та інші фізичні характеристики об'єктів, що також впливають на час, необхідний для їх ідентифікації.

3. *Можливості РСММ*, а саме технічні характеристики роботизованих засобів, включаючи швидкість руху, дальність та точність сенсорів, обчислювальні можливості, також впливають на цей показник.

Припустимо, що ми визначили оптимальний час ідентифікації на основі вищезгаданих факторів, а також на основі історичних даних та експертних оцінок. Цей час може бути встановлений, наприклад, як середнє значення з історичних даних або як певний поріг, що відповідає бажаним параметрам ефективності системи.

Формула для $T_{\text{опт}}$ може бути такою:

$$T_{\text{опт}} = \frac{1}{n}\sum_{i=1}^{n} T_{\text{істор}i} \tag{3.5}$$

або

$$T_{\text{опт}} = P, \tag{3.6}$$

де Tопт – Т оптимальне; $T_{\text{істор}_i}$ – значення і-го історичного часу; n – кількість історичних значень; P – поріг ефективності.

Важливо, що цей показник є динамічним і може змінюватися з часом з огляду на технологічний прогрес, зміни умов місцевості, вдосконалення в методиках моніторингу та інші фактори.

Якщо оптимальний час ідентифікації $T_{\text{опт}}$ для РСММ визначається через моделювання зустрічного руху ОМ та РЗ на конкретній місцевості, то це вимагає більш складного підходу, який враховує динаміку руху та протидію між ОМ та РЗ.

У цьому випадку, $T_{\text{опт}}$ може бути визначено як час, необхідний РЗ для наближення до ОМ на відстань, достатню для впевненої ідентифікації, за умови, що обидва об'єкти рухаються назустріч один одному на заданій місцевості. Формулювання може бути таким:

1. *Моделювання руху.* Спочатку моделюємо рух РЗ та ОМ з використанням відповідних фізичних та кінематичних рівнянь, враховуючи швидкість, напрямок руху, місцевість та інші релевантні фактори.

2. *Визначення часу зустрічі.* Визначаємо час, за який РЗ та ОМ зустрічаються, тобто час, необхідний РЗ для наближення до ОМ на відстань, достатню для ідентифікації. Це може бути виражено через розв'язок відповідної системи диференціальних рівнянь або через чисельне моделювання.

3. *Формулювання* $T_{\text{опт}}$**.** Визначаємо $T_{\text{опт}}$ як середнє або медіанне значення часу зустрічі, отриманого з численних симуляцій з різними початковими умовами, швидкостями, напрямками руху тощо.

$$T_{опт} = \frac{1}{n}\sum_{i=1}^{n} T_{зустрічі_i}, \tag{3.7}$$

де $T_{зустрічі}$ - масив часових значень, отриманих з моделювань; n- кількість зустрічей.

Цей підхід дозволяє отримати реалістичні оцінки $T_{опт}$, враховуючи динаміку реальних ситуацій, і дає змогу адаптувати систему РСММ до конкретних умов та викликів місцевості.

Використання критерію ефективності, що передбачає максимізацію мінімального значення функції для систем моніторингу та маневрового управління (РСММ), обґрунтовується з кількох аспектів.

В першу чергу, такий критерій дозволяє забезпечити всебічну оцінку ефективності РСММ. Поєднуючи ймовірність ідентифікації та час ідентифікації, система оцінюється за допомогою комплексного підходу, відображаючи її загальну здатність ефективно виконувати завдання моніторингу.

Такий критерій сприяє балансуванню ключових показників ефективності. Максимізація мінімального значення допомагає уникнути ситуації, коли один показник ефективності високий, а інший — низький. Це спонукає до розробки збалансованих систем, де обидва показники важливі й враховуються одночасно.

Важливим аспектом є врахування найслабшої ланки. Концентрація на максимізації мінімального значення спрямована на поліпшення тих аспектів системи, які є найслабшими, забезпечуючи тим самим загальну надійність та стабільність РСММ.

Такий підхід також сприяє адаптивності та гнучкості систем. Шляхом максимізації мінімального значення, система стимулює розробку РСММ, здатних адаптуватися до різноманітних умов місцевості та стратегій ухилення об'єктів моніторингу, що є ключовим для ефективності у динамічних та непередбачуваних умовах.

Вищезазначений підхід також сприяє підвищенню загальної продуктивності РСММ. Максимізація мінімального значення показників дозволяє оптимізувати систему для досягнення високої продуктивності в різних аспектах діяльності, підвищуючи ймовірність успішного виконання завдань, навіть у складних умовах.

Зосередження на мінімальному показнику також сприяє зниженню ризику помилок. Цей підхід допомагає системі зменшити ризик помилок чи недоліків,

що можуть виникати внаслідок недостатньої уваги до будь-якого з ключових аспектів функціонування.

Таким чином, максимізація мінімального значення функції вказаних показників є ефективним критерієм для оцінки РСММ, оскільки вона забезпечує збалансований та всебічний підхід до оцінки їх ефективності, стимулюючи поліпшення у всіх ключових областях їх функціонування.

3.2. Розробка моделі середовища моніторингу

Створення математичної моделі середовища [12] для використання в області навчання з підкріпленням вимагає кількох кроків:

1. Визначення цілей та задач

Визначення цілей: Що саме повинні робити роботизовані засоби? Наприклад, знаходити та слідкувати за рухомими об’єктами. Задачі: Які задачі повинні виконувати роботи? Наприклад, навігація до об’єкта, уникнення перешкод, ідентифікація об’єктів.

2. Моделювання середовища

Просторове моделювання: Створення карти місцевості, яка може включати перешкоди, зони видимості, різні типи поверхонь тощо. Динамічні елементи: Визначення параметрів рухомих об’єктів, таких як швидкість, напрямок, поведінка. Сенсорні моделі: Моделювання датчиків роботів, які використовуються для виявлення об’єктів та навігації (наприклад, лідари, камери, ультразвукові датчики).

3. Визначення станів, дій та нагород

Стани: Визначення можливих станів середовища, які можуть спостерігати роботи. Дії: Визначення набору дій, які можуть виконувати роботи (наприклад, рух вперед, поворот, захоплення об’єкта). Нагороди та покарання: Створення системи нагород, яка мотивує роботів виконувати бажані дії, та покарань за помилки або неефективні дії.

Нижче наводиться вербальне визначення цілей та задач системи.

Цілі роботизованих засобів:

РЗ мають за ціль виявлення та слідкування за об'єктами моніторингу. Основна мета полягає в тому, щоб мінімізувати час та відстань, необхідні для досягнення дистанції, достатньої для ідентифікації РО. Це включає здатність РЗ рухатися по місцевості, що впливає на швидкість та маневреність, та використовувати свої сенсори для точного визначення місцезнаходження ОМ.

Задачі РЗ:

1. Навігація до ОМ, враховуючи місцевість та перешкоди.
2. Уникнення перешкод та інших РЗ під час руху.
3. Ідентифікація ОМ, коли РЗ наближаються на достатню відстань.

Цілі ОМ: ОМ намагаються уникнути зближення з РЗ, що може включати зміну швидкості, напрямку руху та використання місцевості для приховування свого місцезнаходження.

З точки зору математичної формалізації, введемо такі позначення та функції:

Позначення: $x_t^{(r)}$ - положення РЗ у момент часу t, $x_t^{(o)}$ - положення ОМ у момент часу t, $v_t^{(r)}$ - швидкість РЗ у момент часу t, $v_t^{(o)}$ - швидкість ОМ у момент часу t, $a_t^{(r)}$ - дія, вибрана РЗ у момент часу t, $d_{iд}$ - відстань достатня для ідентифікації ОМ РЗ, $S(x)$ - функція стану місцевості, що впливає на параметри руху.

Цільова функція для РЗ: Мінімізувати T, де T - час, необхідний для досягнення умови $\left\|x_t^{(r)} - x_t^{(o)}\right\| \leq d_{iд.}$.

Динаміка РЗ:

$$\begin{aligned} x_{t+1}^{(r)} &= x_t^{(r)} + v_t^{(r)}\Delta t + \tfrac{1}{2}S\left(x_t^{(r)}\right)a_t^{(r)}\Delta t^2 \\ v_{t+1}^{(r)} &= v_t^{(r)} + S\left(x_t^{(r)}\right)a_t^{(r)}\Delta t \end{aligned} \tag{3.8}$$

Динаміка РО:

$$x_{t+1}^{(o)} = x_t^{(o)} + v_t^{(o)}\Delta t + \text{стратегія втечі} \tag{3.9}$$

$$V_{t+1}^{(o)} = \text{функція від стратегії втечі} \quad (3.10)$$

Функція втрат для РЗ:

$$J\left(a_t^{(r)}\right) = \sum_{t=0}^{T} \gamma^t C\left(x_t^{(r)}, x_t^{(o)}, a_t^{(r)}\right), \quad (3.11)$$

де C – вартість дії, що включає відстань до ОМ та витрати на маневрування, а γ – фактор дисконтування.

Проблема оптимізації для РЗ:

$$\min_{a_t^{(r)}} J\left(a_t^{(r)}\right) \quad (3.11)$$

З обмеженнями на динаміку РЗ та умовами безпеки.

Запропонована формалізація включає моделювання руху РЗ та ОМ, враховуючи вплив місцевості на параметри руху та ідентифікації. Вона також враховує стратегічну протидію між РЗ та ОМ, де ОМ намагаються уникнути зближення, а РЗ прагнуть мінімізувати відстань до ОМ для їх ідентифікації.

Надалі наводяться результати детального моделювання середовища

Просторове моделювання: Просторове моделювання включає створення віртуальної карти місцевості, яка відображає реальні умови якомога точніше. Це може бути зроблено за допомогою наступних кроків:

1. *Збір даних:* Використання існуючих картографічних сервісів (наприклад, Google Maps, OpenStreetMap) для отримання базової карти місцевості.

2. *Доповнення даними:* Інтеграція даних з нерухомих сенсорів, таких як камери та датчики руху, для додавання інформації про перешкоди, зони видимості та інші важливі особливості.

3. *Анотація карти:* Ручне або автоматизоване додавання метаданих до карти, таких як типи поверхонь, висота об’єктів, та інші фізичні характеристики.

Динамічні елементи: Динамічні елементи середовища включають рухомі об'єкти, які можуть бути модельовані за допомогою наступних параметрів:

1. *Швидкість v:* Скалярна або векторна величина, що визначає, наскільки швидко об'єкт переміщується в просторі.

2. *Напрямок θ:* Вектор, що показує напрямок руху об'єкта.

3. *Поведінка:* Набір правил або алгоритмів, які визначають, як об'єкт реагує на зміни у середовищі.

Відповідно до наведених кроків проведемо математичну формалізацію середовища протидії роботизованих засобів та об'єктів моніторингу.

1. Карта місцевості M:

M може бути представлена як матриця або граф, де кожен елемент або вузол відповідає певній області в просторі. Кожен елемент містить інформацію про тип поверхні S, наявність перешкод O, та інші характеристики.

2. Вплив місцевості на параметри руху:

Коефіцієнт швидкості $\mathrm{k_v(S)}$, який змінює швидкість об'єкта в залежності від типу поверхні. Коефіцієнт маневреності $\mathrm{k_m(S)}$, який впливає на здатність об'єкта змінювати напрямок на різних типах поверхонь.

3. Динаміка об'єктів моніторингу:

Положення об'єкта x_t в часі t може бути визначено як $x_{t+1} = x_t + k_v(S) \cdot v_t \cdot \Delta t$, де Δt – крок часу.

Напрямок руху об'єкта може бути оновлено з урахуванням коефіцієнта маневреності та вхідних дій $\theta_{t+1} = \theta_t + k_m(S) \cdot \Delta\theta$, де $\Delta\theta$ – зміна напрямку.

Ця формалізація дозволяє створити математичну модель середовища, яка може бути використана для симуляції та аналізу поведінки роботизованих засобів та об'єктів моніторингу у різних умовах.

Поняття станів середовища:

Нехай S_t позначає стан середовища в час t, який включає в себе:

Положення та швидкість усіх роботизованих засобів R_t та об'єктів моніторингу O_t.

Стан місцевості та інших елементів середовища M_t. Тоді $S_t = \{R_t, O_t, M_t\}$.

Рівняння переходу:

Рівняння переходу описує, як стан середовища S_t змінюється з часом до S_{t+1}, враховуючи дії роботизованих засобів та поведінку об'єктів моніторингу, а також стан місцевості:

Для роботизованих засобів: положення: $x_{t+1}^{(r)} = x_t^{(r)} + k_v(M_t) \cdot v_t^{(r)} \cdot \Delta t$, швидкість: $v_{t+1}^{(r)} = v_t^{(r)} + a_t^{(r)} \cdot \Delta t$, де $a_t^{(r)}$ – прискорення, яке може залежати від керування та властивостей місцевості M_t.

Для об'єктів моніторингу: положення: $x_{t+1}^{(o)} = x_t^{(o)} + k_v(M_t) \cdot v_t^{(o)} \cdot \Delta t$, швидкість: $v_{t+1}^{(o)} = v_t^{(o)} + a_t^{(o)} \cdot \Delta t$, де $a_t^{(o)}$ – прискорення, яке може бути результатом автономної поведінки або протидії з роботизованими засобами.

Для стану місцевості та інших елементів середовища M :

Стан місцевості M_t включає інформацію про перешкоди, типи поверхонь, зони видимості тощо. Зміни в стані місцевості можуть відбуватися через зовнішні впливи або дії роботизованих засобів: $M_{t+1} = f(M_t, R_t, O_t)$, де f – функція, що описує динаміку місцевості

Коефіцієнти $k_v(M_t)$ та множини дій $a_t^{(r)}, a_t^{(o)}$ можуть бути функціями стану місцевості M_t, які визначають, як середовище впливає на рухомість та поведінку об'єктів. Наприклад, коефіцієнт швидкості може зменшуватися в зонах з високою щільністю перешкод або на складних поверхнях.

Ця модель дозволяє симулювати динамічну протидію між роботизованими засобами, рухомими об'єктами та змінним середовищем, що є ключовим для розробки алгоритму.

Для створення математичної моделі, яка використовується в навчанні з підкріпленням, необхідно визначити стани, дії та нагороди, які будуть використовуватися агентами (в даному випадку роботами) для навчання.

Стани середовища S – це всі можливі спостереження, які робот може зробити. Вони можуть включати:

Положення робота p_r та об'єктів моніторингу p_o.

Швидкість робота v_r та об'єктів моніторингу v_o.

Стан перешкод M та інших елементів середовища.

Інформація про останню відому позицію об’єктів моніторингу.

Дії – це набір можливих кроків, які робот може виконати, наприклад рух вперед, поворот, захоплення об’єкта.

Функція нагороди визначається на основі зменшення дистанції між РЗ та ОМ порівняно з “ідеальною” дистанцією, яка була б, якби обидва об’єкти рухалися назустріч один одному з максимальною швидкістю без перешкод.

$$\begin{aligned} R_t^{(r)} &= k(d_{t-1}^* - d_t) \\ R_t^{(o)} &= -R_t^{(r)} \end{aligned} \quad , \qquad (3.12)$$

де d_t - фактична дистанція між РЗ та РО, d_t^* - "ідеальна" дистанція, а k - позитивний коефіцієнт масштабування.

Моделювання завершується, коли:

1. РЗ досягає РО$(d_t \leq \epsilon)$.
2. РО виходить за межі визначеного середовища ($p_o \notin$ Environment Bounds).
3. РЗ потрапляє в область, де він не може функціонувати $(M(p_r) =$ "untraversab"

Таким чином, наведена модель дозволяє створити чітку систему нагород, яка мотивує РЗ ефективно переслідувати РО, а ОМ - уникати переслідування, та визначає чіткі умови для завершення симуляції.

Тепер з використанням CNN визначимо коефіцієнти k впливу середовища на параметри РЗ та ОМ. Створимо графічне відображення середовища. Розширимо точку знаходження РЗ(ОМ) до певної ділянки. Створимо набір даних типових ділянок середовища з заданими значеннями параметрів. Виконаємо доповнення набору даних шляхом змішування у випадковій пропорції типових ділянок з відповідною апроксимацією значень параметрів. Далі створюємо CNN для визначення значень параметрів в будь-якій ділянці середовища. Надати

детальний опис, алгоритм та математичну формалізацію методики визначення впливу середовища на параметри РЗ(ОМ).

Для визначення коефіцієнтів k впливу середовища на параметри РЗ та об'єктів моніторингу за допомогою згорткових нейронних мереж (CNN), можна використати наступний підхід:

1. *Створення графічного відображення середовища:* Спершу потрібно створити графічне відображення середовища, яке може бути представлене у вигляді карт з різними текстурами, що відображають типи поверхонь (трава, пісок, вода тощо).

2. *Розширення точки знаходження РЗ(ОМ) до певної ділянки:* Кожен РЗ або ОМ може впливати на ділянку середовища, яка його оточує. Це може бути представлено як радіус дії або як зона впливу навколо кожного об'єкта.

3. *Створення набору даних типових ділянок середовища:* Необхідно зібрати набір даних, який містить зображення типових ділянок середовища з відомими значеннями параметрів, таких як шорсткість, кут нахилу, вологість тощо.

4. *Доповнення набору даних:* Для збільшення різноманітності набору даних можна використати техніки аугментації даних, такі як змішування зображень ділянок у випадковій пропорції та апроксимація значень параметрів для цих нових змішаних ділянок.

5. *Створення CNN для визначення параметрів ділянки середовища:* CNN може бути навчена на основі зібраного набору даних для визначення значень параметрів будь-якої ділянки середовища на основі її зображення.

На основі даного підходу, розроблено алгоритм та математичну формалізацію методики:

1 Попередня обробка даних:

Збір даних: $D = \{(x_i, y_i)\}_{i=1}^{N}$, де x_i - зображення ділянки, а y_i - вектор параметрів ділянки.

Аугментація даних: створення нових зображень x_{new} шляхом комбінування зображень x_i та апроксимація відповідних y_{new} .

2 Архітектура CNN:

Визначення архітектури CNN: $f_\theta(x)$, де θ - параметри мережі, які підлягають навчанню.

Втрати: $L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \text{loss}\,(f_\theta(x_i), y_i)$, де loss - функція втрат, наприклад, середньоквадратична помилка.

3 Навчання мережі:

Використання алгоритму зворотного поширення помилки для оптимізації θ з метою мінімізації $L(\theta)$.

4 Використання CNN для визначення k :

Для нового зображення x_{test} CNN визначає параметри ділянки $\hat{y}_{\text{test}} = f_\theta(x_{\text{test}})$.

Коефіцієнти k визначаються на основі $\hat{y}_{\text{test}}$, наприклад, через лінійну регресію або іншу модель залежності.

Цей підхід дозволяє автоматизувати процес визначення впливу середовища на параметри РЗ та ОМ, забезпечуючи адаптацію до різноманітних умов середовища.

Загальна методика моделювання середовища для задачі пошуку та переслідування РЗ об’єктів моніторингу в контексті навчання з підкріпленням може бути сформульована наступним чином:

1. Визначення цілей та задач: Встановлення конкретних цілей для РЗ та ОМ. Формулювання задач, які повинні виконувати РЗ для досягнення цілей.

2. Моделювання середовища: Створення графічного відображення середовища, включаючи перешкоди, зони видимості, різні типи поверхонь. Визначення динамічних елементів, таких як параметри руху ОМ.

3. Визначення станів, дій та нагород: Опис можливих станів середовища S , які включають положення та швидкість РЗ та ОМ, а також стан перешкод M. Визначення набору дій A, які можуть виконувати РЗ. Розробка системи нагород, яка мотивує РЗ виконувати бажані дії та покарань за помилки.

4. Формулювання функції переходу станів: Визначення рівнянь переходу, які описують, як середовище переходить з одного стану в інший відповідно до дій РЗ.

5. Визначення функції нагороди: Формулювання функції нагороди, яка базується на зменшенні дистанції між РЗ та ОМ та інших критеріях.

6. Критерії завершення моделювання: Встановлення умов, за яких моделювання завершується, таких як досягнення ОМ, вихід ОМ за межі середовища, або неможливість функціонування РЗ.

7. *Використання CNN для визначення впливу середовища:* Збір та аугментація даних про середовище. Навчання CNN на основі зібраних даних для визначення коефіцієнтів k, які впливають на параметри РЗ та ОМ. Використання CNN для аналізу нових ділянок середовища та прогнозування впливу на РЗ та ОМ.

8. Інтеграція моделі в RL: Впровадження моделі середовища в алгоритми RL. Використання отриманих даних для оптимізації стратегій поведінки РЗ та ОМ.

Ця методика дозволяє створити детальну та адаптивну модель середовища, яка може використовуватися для тренування РЗ в різноманітних умовах та підвищення ефективності їхніх дій у реальному світі.

3.3. Розробка моделі роботизованого засобу та моделі переміщення об’єкта пошуку

Для побудови моделі РЗ необхідно визначити такі компоненти [63]:

Стан (S): Представляє поточне відображення середовища РЗ. Стан може бути представлений як вектор $s \in S$, де S - простір усіх можливих станів.

Дія (A): Конкретні дії, які РЗ може виконати. Дія подається як $a \in A(s)$, де $A(s)$ - множина можливих дій у стані s.

Політика (π): Стратегія або правило, за яким РЗ вибирає дії, базуючись на поточному стані. Політика $\pi(a \mid s)$ визначає ймовірність вибору дії a у стані s.

Функція винагороди (R): Визначає винагороду або покарання за виконання певної дії в певному стані.

$R(s, a)$ визначає винагороду за виконання дії a у стані s

Функція цінності (V): Оцінює очікувану суму майбутніх винагород, починаючи з певного стану s, слідуючи політиці π. $V^{\pi}(s)$ визначає цінність стану s при політиці π. Зазвичай формується на основі рівняння Беллмана:

$$V^{\pi}(s) = \sum_{a \in A(s)} \pi(a \mid s) \sum_{s' \in S} P(s' \mid s, a)[R(s, a) + \gamma V^{\pi}(s')], \quad (3.13)$$

де γ - коефіцієнт дисконтування, що визначає важливість майбутніх винагород.

Функція переходу станів (P): Визначає ймовірність переходу з одного стану в інший після виконання певної дії. $P(s' \mid s, a)$ визначає ймовірність переходу в стан s' з стану s після виконання дії a. Задача навчання з підкріпленням полягає у навчанні агенту (у нашому випадку, роботизованого засобу) вибору оптимальних дій в середовищі для максимізації загальної винагороди протягом часу. Агент взаємодіє з середовищем, яке реагує на його дії та змінює свій стан, надаючи агенту винагороду або покарання.

Ціль агента – знайти оптимальну політику π^*, яка максимізує очікувану суму винагород:

$$\pi^* = \arg \max_{\pi} \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R_t \mid \pi], \quad (3.14)$$

де R_t - винагорода в час t, а γ - коефіцієнт дисконтування (зазвичай між 0 та 1), який визначає важливість майбутніх винагород.

Для оцінки та оптимізації дій агента в різних станах використовується Q-функція $Q^{\pi}(s, a)$, яка визначає очікувану суму винагород за вибір дії a у стані s та слідування політиці π після цього:

$$Q^{\pi}(s,a) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^{k} R_{t+k+1} \mid s_t = s, a_t = a, \pi] \quad (3.15)$$

Рівняння Беллмана для Q-функції матиме такий вигляд:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) \max_{a'} Q^{\pi}(s',a') \,, \quad (3.16)$$

де $P(s' \mid s,a)$ - ймовірність переходу в стан s' після виконання дії a у стані s.

Наведені відповідно до [63] математичні залежності, складають основу моделі, що дозволяє РЗ адаптуватися до змін у середовищі та поведінці об'єктів, оптимізуючи свої дії для досягнення поставлених цілей. Надалі надаються результати досліджень щодо визначення складових залежностей (3.13-3.16).

Для опису та математичного формулювання середовища, в якому діють РЗ, використовується інтегрований підхід, заснований на початкових даних карти місцевості, отриманих з відкритих джерел, та на даних, які доповнюються в процесі діяльності РЗ.

Початкова конфігурація середовища створюється на основі карти місцевості M, за рахунок використання відкритих геоданих для створення базової карти місцевості.

Карта визначається як матриця $M = [m_{ij}]$, де кожен елемент m_{ij} репрезентує ділянку місцевості з певними атрибутами або параметрами P (наприклад, тип поверхні, перешкоди). Параметри місцевості подаються у формі $P = \{p_1, p_2, \dots, p_n\}$, де кожен $p_i \epsilon$ характеристикою місцевості.

В процесі функціонування роботизованої системи моніторингу буде здійснюватися доповнення середовища на основі даних, отриманих під час функціонування РЗ – D_{RZ}, шляхом внесення відповідних змін до M та P: $A(M, D_{RZ}) \rightarrow M'$, де M' – оновлена карта місцевості.

Таким чином, в процесі навчання та адаптації РЗ використовується динамічне середовище, що неперервно оновлюється, для моделювання реальних умов та поліпшення стратегій протидії. Цей підхід дозволяє створити адаптивне та гнучке середовище, яке відображає реальні умови місцевості та забезпечує ефективне навчання та функціонування РЗ. Важливим елементом є здатність середовища динамічно адаптуватися на основі зібраних РЗ даних, що дозволяє забезпечити актуальність та релевантність моделі середовища [14].

Для визначення параметрів РЗ відповідно до їх місцезнаходження в середовищі, використовувались методи CV та CNN:

Використовуються дані, отримані в ході експлуатації РЗ, які можуть включати зображення, сенсорні дані та іншу інформацію про середовище. Для кожного типу ділянки (наприклад, міська місцевість, лісові масиви, водойми) заздалегідь визначаються типові параметри РЗ, такі як швидкість, енергоспоживання, ефективність сенсорів. Припустимо, що кожній ділянці d відповідає набір параметрів $P_d = \{p_1, p_2, \dots, p_n\}$, де p_i відображає конкретний параметр РЗ.

Тоді за допомогою CNN можливо провести навчання нейронної мережі розпізнаванню різних типів ділянок місцевості та визначати на цій основі відповідні параметри РЗ за рахунок згортки параметрів отриманого зображення до типової ділянки місцевості $f_{CNN}(\mathrm{D_{RZ}}) \rightarrow d$, де d – тип ділянки середовища. В подальшому, на основі визначеного типу ділянки d, обчислюються відповідні параметри P_d. Даний підхід дозволяє здійснювати динамічне оновлення параметрів РЗ в залежності від змін середовища та адаптувати поведінку РЗ в залежності від оновлених параметрів, що забезпечує оптимальне використання ресурсів та ефективне виконання завдань у різних умовах середовища.

Цей підхід дозволяє РЗ адаптуватися до різних умов середовища, використовуючи передові технології комп'ютерного зору та машинного навчання для аналізу та відповідного налаштування їх параметрів. Використання CNN для аналізу зображень дозволяє точно визначати тип місцевості та адаптувати поведінку РЗ згідно з вимогами конкретної ділянки.

Під час моделювання розглядаються основні типи дій РЗ, що включають переміщення в середовищі, поворот, зупинку та зміну швидкості руху. За потреби множину дій може бути розширено. Припустимо, що стан РЗ у середовищі описується як $S = (x, y, \theta)$, де x, y - координати РЗ, а θ – орієнтація. Тоді дії РЗ можна формалізувати наступним чином:

Переміщення:

$$x' = x + d \cdot \cos(\theta)$$
$$y' = y + d \cdot \sin(\theta)$$
$$S' = (x', y', \theta), \tag{3.17}$$

де d - відстань переміщення.

Поворот:

$$\theta' = \theta + \phi$$
$$S' = (x, y, \theta'), \tag{3.18}$$

де ϕ - кут повороту.

Зупинка: $S' = S$ (без зміни стану). Зміна параметра швидкості РЗ, не впливає безпосередньо на S, але змінює динаміку переміщення.

Кожна дія A викликає зміну стану РЗ у середовищі відповідно до функції середовища $F_{\text{env}}(S, A) \to S'$.

Вибір дії A залежить від поточного стану S, цілей РЗ та умов середовища.

Цей підхід дозволяє формалізувати та регулювати поведінку РЗ у середовищі, забезпечуючи ефективне планування траєкторій та адаптивність до змін умов. Використання математичних моделей для представлення дій дозволяє точно моделювати поведінку РЗ і робить можливим використання складних алгоритмів навчання та оптимізації.

Політика агента є набором правил або стратегій, які визначають дії РЗ в залежності від стану середовища та динаміки ОМ. Ця політика має на меті оптимізувати певні цілі, наприклад, ефективність моніторингу або мінімізацію часу для досягнення завдань. Політика π може бути визначена як функція, яка відображає стан середовища S у дії A, які має виконати РЗ $\pi: S \to A$, де S – це

вектор стану, що включає інформацію про розташування та орієнтацію РЗ, а також про стан та динаміку ОМ в даній місцевості.

Інтелектуальні динамічні ОМ постійно змінюють стан середовища, що вимагає від РЗ адаптації своєї стратегії, а саме:

реакція на зміни: політика РЗ повинна включати механізми для адаптації до змін в поведінці ОМ;

прогнозування дій ОМ: Використання методів прогнозування потенційних змін у поведінці ОМ.

Використання навчання з підкріпленням на основі політики агента дозволяє РЗ ефективно адаптуватися до динамічних змін у поведінці інтелектуальних ОМ, забезпечуючи гнучкість та високу реактивність системи моніторингу.

Функція винагороди в контексті навчання з підкріпленням для РЗ враховує час та ймовірність наближення до ОМ та запобігання виходу ОМ за межі зони моніторингу, може бути описана та формалізована наступним чином:

Функція винагороди $R(s, a)$ визначає винагороду або штраф, яку отримує РЗ за виконання дії a в стані s. Ця функція оцінює ефективність дій РЗ у контексті двох основних цілей:

Наближення до ОМ: РЗ отримує позитивну винагороду за успішне наближення до ОМ, що збільшується з підвищенням ймовірності виявлення ОМ.

Запобігання виходу ОМ з зони моніторингу: РЗ отримує штраф, якщо ОМ виходить за межі зони моніторингу.

Припустимо, що:

P_{approach} – ймовірність успішного наближення РЗ до ОМ.

T_{approach} – час, необхідний РЗ для наближення до ОМ.

P_{exit} – ймовірність виходу ОМ за межі зони моніторингу.

Тоді функція винагороди може бути визначена як:

$$R(s, a) = w_1 \cdot f\left(P_{\text{approach}}, T_{\text{approach}}\right) - w_2 \cdot g(P_{\text{exit}}), \tag{3.19}$$

де: $f\left(P_{\text{approach}}, T_{\text{approach}}\right)$ – функція, що оцінює винагороду за наближення до ОМ, може включати час та ймовірність наближення.

$g(P_{\text{exit}})$ – функція, що оцінює штраф за вихід ОМ за межі моніторингу.

w_1, w_2 – вагові коефіцієнти, що визначають важливість кожної цілі.

Для практичного моделювання поведінки РЗ доцільно використовувати TF-Agents (агенти TensorFlow) в рамках фреймворку TensorFlow. Процес моделювання включає встановлення необхідних бібліотек TensorFlow та TF-Agents. Після цього створюється відповідне середовище, що відображає умови, в яких буде функціонувати РЗ, включаючи типи дій і стан РЗ. Дані, отримані за допомогою комп'ютерного зору, та інтеграція моделей згорткових нейронних мереж допомагає у визначенні стану РЗ на основі зображень.

Наступним кроком є налаштування агента TF-Agents, що найкраще відповідає задачам та умовам середовища, у якому оперує РЗ. Агент навчається на основі взаємодій з середовищем, збираючи досвід через серію дій та відповідних винагород.

Процес навчання агента включає збір даних про протидію, оптимізацію політики на основі цих даних та регулярне оновлення політики агента для покращення його ефективності. Після навчання важливо оцінити ефективність агента, запустивши його в середовищі та проаналізувавши отримані результати. Це дозволяє визначити, наскільки добре агент адаптується до змін умов та виконує поставлені задачі.

Для послідовного тренування ОМ та РЗ можна використати підхід навчання з підкріпленням, де обидва типи агентів – ОМ та РЗ – тренуються за допомогою алгоритмів, які використовують протилежні функції винагород наступним чином:

Ініціалізація ОМ: ОМ спочатку тренуються без протидії з боку РЗ. З наступним завданням — навчитися уникати ідентифікації, при цьому функція винагороди має стимулювати уникати РЗ та досягнути межі зони моніторингу

Тренування ОМ: ОМ тренуються за допомогою обраного алгоритму навчання з підкріпленням, намагаючись максимізувати свою винагороду.

Після досягнення певного рівня вмінь ОМ, розпочинається тренування РЗ, що проводиться подібно до тренування ОМ із протилежною функцією винагороди.

Процес тренування включає в себе декілька ітерацій:

Оцінка та адаптація: після кожного циклу тренування проводиться оцінка ефективності та адаптація стратегій як РЗ, так і ОМ.

Моніторинг параметрів: за результатами ітерації визначаються ймовірність та час ідентифікації ОМ.

Критерії завершення: тренування продовжується до тих пір, поки параметри не стабілізуються.

Цей підхід дозволяє створити балансовану взаємодію між ОМ та РЗ, де обидва типи агентів оптимізують свої стратегії у відповідь на дії один одного. Використання протилежних функцій винагород дозволяє симулювати реалістичну взаємодію, де ОМ намагаються уникати, а РЗ здійснити ідентифікацію.

Таким чином, в ході досліджень було сформовано модель роботизованого засобу моніторингу інтелектуальних динамічних об'єктів. Методика фокусується на протидії між РЗ та ОМ у динамічному середовищі. Суть моделі полягає у використанні методів навчання з підкріпленням, де РЗ та ОМ оптимізують свої стратегії відповідно до функцій винагороди, що враховують ймовірність ідентифікації та час. Модель призначена для створення ефективного механізму моделювання моніторингу та протидії між РЗ та ОМ, з можливістю адаптації до змінних умов середовища. Алгоритм реалізації моделі являє собою ітеративний процес, що включає налаштування середовища, тренування агентів, оцінку їхньої взаємодії та адаптацію стратегій.

Розроблена модель є корисною для розробки систем моніторингу, де важлива адаптивність до змін умов та ефективне взаємодіюче навчання РЗ та ОМ.

Для оцінки працездатності розробленої моделі РЗ проведено експеримент, що полягав у моделюванні протидії між РЗ та ОМ, де параметри визначалися

випадковим чином. Використовуючи TensorFlow у якості основного інструменту для обчислень та аналізу даних, експеримент проводився протягом 20 ітерацій. Кожна ітерація включає оцінку взаємодії між РЗ та ОМ, аналізуючи їх поведінку, стратегії руху, та реакції на зміну умов. Цей дозволило сформувати та дослідити ефективність різних стратегій протидії в динамічному середовищі. Результати проведених експериментальних досліджень наведено на рисунку 3.1.

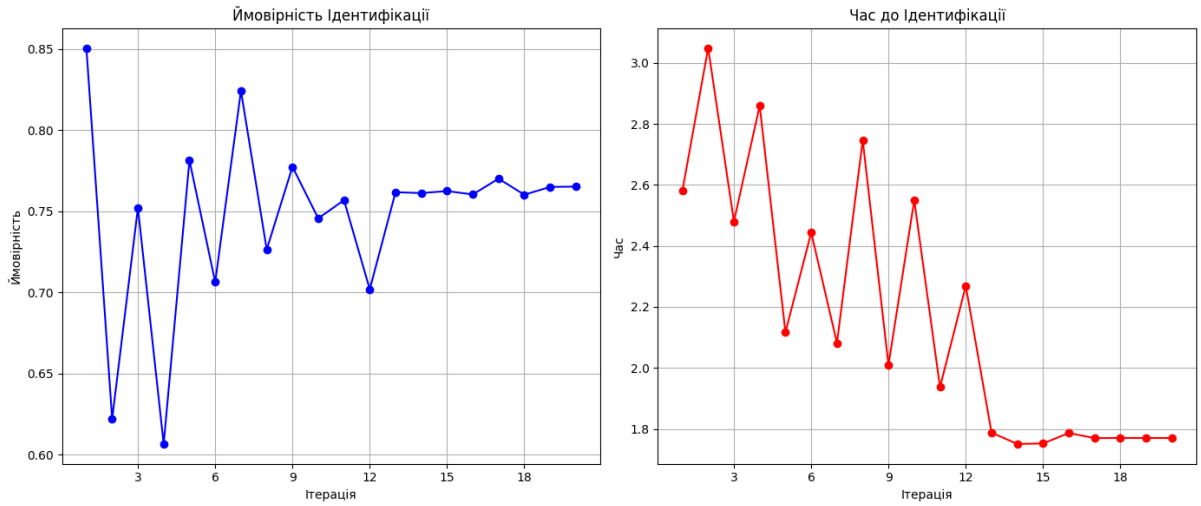

Рис. 3.1 Результати експериментальних досліджень

Результати експериментів дозволяють дійти таких висновків: модель демонструє високу точність у відстеженні ОМ, особливо в складних умовах; РЗ швидко адаптуються до змін у поведінці ОМ та умовах середовища; модель показує хорошу стійкість до помилок та непередбачуваних ситуацій.

Ці результати свідчать про високу ефективність та адаптивність розробленої моделі, що робить її перспективною для застосування в реальних умовах.

3.4 Розробка методики оцінки ефективності

Методику оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання, а саме навчання з підкріпленням

можна описати через алгоритм на рисунку 3.2. Основні етапи алгоритму включають:

1. *Побудова взаємодії:* Розробка середовища, де роботизовані засоби взаємодіють з об'єктами моніторингу, які уникають цих засобів.

2. *Визначення характеристик:* Оцінка характеристик ОМ та РЗ відносно властивостей місцевості.

3. *Початкова конфігурація:* Початок із одного РЗ.

4. *Визначення політики ОМ:* Розробка стратегії для об'єктів моніторингу.

5. *Цикл навчання:* Послідовне навчання РЗ та ОМ із врахуванням їх взаємодії. РЗ повинен знаходити та ідентифікувати ОМ, а ОМ повинен уникати ідентифікації.

6. *Оцінка параметрів:* Визначення часу ідентифікації (t_i) та ймовірності знаходження (P_i).

7. *Аналіз ітерацій*: Оцінка різниці параметрів між ітераціями.

8. *Умови завершення:* Якщо зміна параметрів стабілізується, але ймовірність ідентифікації є вищою за 80% і час знаходження менший за максимальний час роботи РЗ, алгоритм завершується і визначаються показники ефективності. Якщо умови не виконані, процес продовжується з додаванням ще одного РЗ.

Цей підхід є комплексним і включає ітеративний процес, що дозволяє систематично оцінювати та вдосконалювати взаємодію між РЗ і ОМ. Ключовим елементом є адаптація та оптимізація стратегій поведінки об'єктів моніторингу та роботизованих засобів з урахуванням змінних умов місцевості.

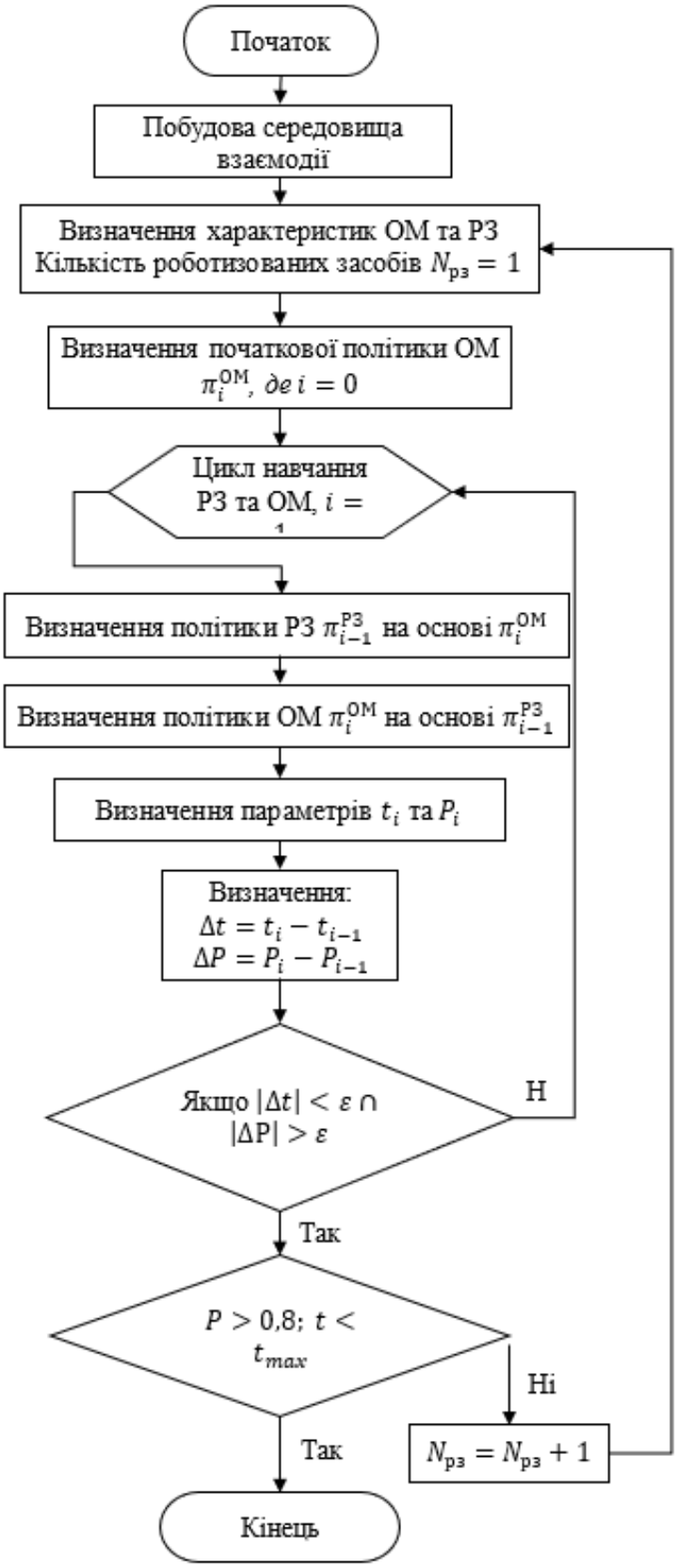

Рис. 3.2 Алгоритм методики оцінки ефективності роботизованої системи моніторингу місцевості

Першим кроком алгоритму є побудова середовища взаємодії між роботизованими засобами та об'єктами моніторингу, що уникають цих засобів, що є ключовим компонентом методики. Розробка середовища взаємодії може використовувати як віртуальне, так і фізичне середовище. Віртуальне середовище використовується для моделювання взаємодії. Це може бути програмне забезпечення для симуляції, яке дозволяє тестувати різні сценарії та умови без реальних ризиків чи витрат. Фізичне середовище передбачає застосування реальних роботів та об'єктів у контрольованому оточенні. Це забезпечує реалістичні дані, але вимагає більше ресурсів та обережності. В роботі передбачено використання віртуального середовища з метою застосування його до різноманітних РЗ. Це обумовлено можливістю моделювання різноманітних типів місцевості.

У контексті моделювання взаємодії враховуються РЗ, які можуть включати дрони, наземні роботи та інші пристрої. Ці РЗ оснащені сенсорами для виявлення та слідкування. Паралельно з РЗ розглядаються ОМ, які можуть представляти собою інші роботизовані пристрої або спеціально розроблені об'єкти. Ці об'єкти імітують поведінку цільових об'єктів моніторингу.

У рамках стратегій уникнення розроблені алгоритми для ОМ, які дозволяють їм виявляти присутність РЗ та активно уникати їх. Ці алгоритми можуть передбачати зміну маршруту чи швидкості ОМ з метою ефективного уникнення потенційних конфліктів.

Реалізація інтелектуальних алгоритмів включає в себе навчання з підкріпленням та адаптацію до умов місцевості. Для РЗ навчання з підкріпленням спрямоване на здатність виявляти та ідентифікувати ОМ, коли вони уникають ідентифікації. Для ОМ навчання орієнтоване на ефективне уникнення ідентифікації, адаптуючись до тактик роботизованих засобів. РЗ та ОМ повинні адаптуватися до різноманіття місцевості, включаючи перешкоди, різні типи ландшафтів тощо. Також необхідно враховувати зміни у погоді, освітленні та інші фактори, що можуть впливати на взаємодію. Ці аспекти взаємодіють, створюючи інтелектуальні алгоритми, які дозволяють

роботизованим засобам та об’єктам моніторингу адаптуватися до різноманітних умов та оптимально взаємодіяти між собою.

Моніторинг і оцінка включають кілька етапів, починаючи зі збору даних. На цьому етапі фіксуються різноманітні аспекти взаємодії, такі як час виявлення, маршрути уникнення та ефективність сенсорів. Подальшим етапом є аналіз ефективності, де проводиться оцінка того, наскільки добре РЗ виявляють ОМ, а також наскільки ефективно об'єкти моніторингу уникають виявлення. Цей комплексний підхід дозволяє отримати повніше розуміння взаємодії та ефективності системи. Цей процес не тільки дозволяє оцінити ефективність роботизованої системи моніторингу, але й сприяє вдосконаленню алгоритмів штучного інтелекту та автономних систем.

Використання технологій CV та CNN в роботі для моделювання значно розширює потенціал середовища взаємодії. Застосування технологій CV для обробки зображень та карт місцевості включає кілька етапів:

1. Обробка зображень та карт, що включає збір даних, а саме використання зображень та карт місцевості з різних джерел, таких як супутникові знімки, аерофотозйомка тощо, і попередню обробку- покращення якості зображень, корекція спотворень, нормалізація освітлення та інші техніки для підготовки даних до аналізу.

2. Виявлення особливостей місцевості, що складається з класифікації місцевості - визначення типів ландшафту, таких як ліси, водойми, відкриті простори, урбанізовані зони тощо, і виявлення об’єктів -розпізнавання штучних та природних об’єктів, які можуть впливати на місії моніторингу.

Застосування CNN в роботі включає кілька ключових етапів для аналізу зображень та визначення характеристик РЗ:

1. Аналіз Зображень. Тренування CNN включає навчання мереж на великій кількості анотованих зображень для виявлення та класифікації особливостей місцевості. Після тренування CNN застосовується для екстракції ознак із зображень, необхідних для визначення умов місцевості.

2. Визначення Характеристик РЗ. Адаптація до умов місцевості включає аналіз зображень, що дозволяє РЗ адаптуватися до змінних умов, оптимізуючи маршрути та стратегії пошуку. За допомогою генералізації областей місцевості встановлюються зони з певними характеристиками, які впливають на поведінку РЗ. Цей підхід дозволяє ефективно використовувати згорткові нейронні мережі для оптимізації взаємодії роботизованих засобів з навколишнім середовищем.

Інтеграція у Середовище Взаємодії включає кілька ключових етапів, які сприяють вдосконаленню взаємодії між РЗ та ОМ:

1. Створення віртуальних моделей РЗ та ОМ, які взаємодіють в умовах, симульованих на основі аналізу зображень. Таким чином, можна відтворити різноманітні сценарії взаємодії для ефективного тестування та аналізу.

2. Експериментування з різними стратегіями взаємодії в умовах, симульованих згідно з отриманими даними. Це дозволяє визначити оптимальні підходи до взаємодії між РЗ та ОМ.

3. Оптимізація Поведінки Навчання з Підкріпленням. Використання набору даних з CV та CNN для покращення алгоритмів навчання з підкріпленням, адаптуючи РЗ та ОМ до реальних умов. Цей підхід сприяє оптимізації поведінки системи в реальному середовищі з урахуванням отриманих даних та забезпеченням адаптації до різних умов взаємодії.

Застосування цих технологій не лише підвищує точність і адаптивність роботизованих систем моніторингу, але й відкриває нові можливості для розширення їхнього функціоналу в різних умовах та сценаріях.

Визначення характеристик ОМ та РЗ відносно властивостей місцевості включає математичну формалізацію, що дозволяє оцінити, як властивості місцевості впливають на поведінку та ефективність цих об'єктів. Детально цей процес можна описати так.

1. Визначення властивостей місцевості. Місцевість класифікується за типом T.

$$T = t_1, t_2, \dots, t_n, \tag{3.20}$$

де t_i представляє різні типи місцевості (наприклад, ліс, міська зона, водойма). Також необхідно взяти до уваги параметри місцевості P.

$$P = p_1, p_2, \dots, p_m, \tag{3.21}$$

де кожен p_i відображає специфічний параметр місцевості, як-от перешкоди, щільність об'єктів, видимість.

2. Визначення характеристики ОМ та РЗ. Характеристики ОМ включають:

Спостережуваність S_{OM}, а саме ймовірність виявлення ОМ РЗ.

$$S_{OM} = f(T, P, C_{OM}), \tag{3.22}$$

де C_{OM} – характеристики ОМ.

Маневреність M_{OM} - здатність ОМ уникати виявлення.

$$M_{OM} = g(T, P, C_{OM}) \tag{3.23}$$

Характеристики РЗ включають:

Ефективність пошуку $E_{\text{РЗ}}$ - здатність РЗ ефективно знаходити ОМ.

$$E_{\text{РЗ}} = h(T, P, C_{\text{РЗ}}), \tag{3.24}$$

де $C_{\text{РЗ}}$ – характеристики РЗ.

Мобільність $M_{\text{РЗ}}$ - здатність РЗ пересуватися та адаптуватися до умов місцевості.

$$M_{\text{РЗ}} = i(T, P, C_{\text{РЗ}}) \tag{3.25}$$

3. Математичне моделювання протидії має на меті створити функцію протидії:

$$I(T, P, C_{\text{ОМ}}, C_{\text{РЗ}}) \tag{3.26}$$

Це функція, що оцінює ефективність протидії між РЗ та ОМ. Враховує всі вищезгадані параметри та характеристики. Окрім того для протидії необхідна оптимізацію цільової функції, а саме максимізація або мінімізація цільової функції, наприклад, максимізація $E_{\text{РЗ}}$ та мінімізація $S_{\text{ОМ}}$.

4. Адаптація та оптимізація проводиться за допомогою алгоритмів навчання з підкріпленням. Налаштування параметрів $C_{\text{ОМ}}$ та $C_{\text{РЗ}}$ відбувається за допомогою використання алгоритмів навчання з підкріпленням для оптимізації поведінки ОМ та РЗ відносно динамічних умов місцевості.

Цей підхід дозволяє не тільки кількісно оцінити протидію між РЗ та ОМ, але й ефективно адаптувати їхні стратегії до конкретних умов місцевості, підвищуючи загальну ефективність системи моніторингу.

Початкова конфігурація алгоритму, передбачає початок з одного РЗ, що є важливим етапом для встановлення базових параметрів системи та оцінки її початкової ефективності.

Перший етап включає в себе вибір типу РЗ відповідно до конкретних цілей моніторингу, що може охоплювати такі варіанти, як дрони, наземні роботи, чи підводні апарати.

Оцінка технічних параметрів РЗ становить важливий аспект цього процесу. Серед основних характеристик враховуються швидкість, маневреність, дальність дії, вантажопідйомність та типи використовуваних сенсорів. Крім того, визначається необхідність різноманітних сенсорів, таких як камери, лідари, інфрачервоні датчики, для ефективної роботи РЗ.

Другий аспект полягає в інтеграції обраного РЗ з відповідною сенсорною апаратурою. Це включає в себе забезпечення сумісності РЗ із системами управління та обробки даних, щоб забезпечити ефективне використання отриманих інформаційних потоків. Такий підхід сприяє оптимізації функціональності та забезпеченню ефективного використання обраного роботизованого засобу в рамках визначених завдань моніторингу.

Другий етап - налаштування початкових параметрів включає в себе кілька ключових аспектів для ефективного впровадження обраного РЗ. Перше завдання полягає в конфігурації системи управління РЗ, де встановлюються первинні параметри для автономного та дистанційного управління. Це включає в себе розробку налаштувань режимів роботи, що враховують специфіку завдань.

Додатково, проводиться програмування базових алгоритмів навігації та маневрування, адаптованих до конкретного типу місцевості, що дозволяє РЗ ефективно взаємодіяти з оточуючим середовищем. Цей етап важливий для забезпечення точності та надійності руху РЗ під час виконання завдань.

Щодо ініціалізації задач, визначаються первинні цілі моніторингу, такі як збір даних, розпізнавання об'єктів та відстеження руху. Також розробляються початкові сценарії для тестувань, що відображають потенційні умови використання РЗ. Ці сценарії допомагають оцінити реакцію системи на різноманітні ситуації та забезпечують базову платформу для подальшого удосконалення алгоритмів та функціональності РЗ.

У контексті тестування та валідації РЗ, розглядаються два основні аспекти: перевірка функціональності та оцінка початкової ефективності.

Першим етапом є проведення функціональних тестів для оцінки робочих характеристик РЗ. Це включає тести маневреності, визначення дальності дії та точності сенсорів. Після виконання тестів проводиться аналіз зібраних даних з метою оцінки якості та точності інформації, отриманої в ході тестових місій.

Другий аспект включає оцінку початкової ефективності РЗ. Це визначення ключових метрик ефективності, таких як час виявлення об'єктів, точність ідентифікації та витривалість обладнання. Отримані результати служать

основою для корекції та оптимізації конфігурації РЗ. Внесення виправлень базується на аналізі результатів тестів та отриманому зворотному зв'язку, що сприяє подальшій покращенню функціональності та ефективності системи.

Третій етап включає тестування та валідацію РЗ, де розглядаються два основні аспекти: перевірка функціональності та оцінка початкової ефективності.

Першим етапом є проведення функціональних тестів для оцінки робочих характеристик РЗ. Це включає тести маневреності, визначення дальності дії та точності сенсорів. Після виконання тестів проводиться аналіз зібраних даних з метою оцінки якості та точності інформації, отриманої в ході тестових місій.

Другий аспект включає оцінку початкової ефективності РЗ. Це визначення ключових метрик ефективності, таких як час виявлення об'єктів, точність ідентифікації та витривалість обладнання. Отримані результати служать основою для корекції та оптимізації конфігурації РЗ. Внесення виправлень базується на аналізі результатів тестів та отриманому зворотному зв'язку, що сприяє подальшій покращенню функціональності та ефективності системи.

Ця початкова конфігурація із одного РЗ дозволяє встановити базові вимірювання та оцінити загальну готовність системи до більш складних задач та сценаріїв взаємодії, що є важливим для подальшої оптимізації та масштабування системи.

Визначення політики для ОМ є наступним кроком алгоритму, що включає розробку стратегій, які ОМ використовують для уникнення виявлення роботизованими засобами (РЗ). Наведемо детальний опис:

1. Модель поведінки ОМ складається зі стану ОМ та множини можливих дій. Стан ОМ $\boldsymbol{s}$: кожен стан ОМ s визначається набором параметрів, таких як розташування, швидкість, напрямок руху.

Параметри стану P_s:

$$P_s = \{x, y, v, \theta, \dots\}, \tag{3.27}$$

де x, y – координати, v – швидкість, θ – напрямок.

Можливі Дії визначаються набором

$$A = \{a_1, a_2, \dots, a_n\} \tag{3.28}$$

Ці дії ОМ може виконувати для зміни свого стану.

2. Функція переходу визначає динаміку переходу.

$$T(s, a, s') \tag{3.29}$$

Функція переходу показує ймовірність переходу з стану s в стан s' після виконання дії a.

3. Функція винагороди оцінює вигоду або втрату за виконання дії a в стані s.

$$R(s, a) \tag{3.30}$$

Наприклад, висока винагорода за уникнення виявлення.

4. Оптимізація політики. Політика $\pi(\boldsymbol{s})$ – це функція, що визначає оптимальну дію a в кожному стані s.Мета оптимізації політики максимізація сумарної винагороди: $\max \sum R\big(s, \pi(s)\big)$.

5. Алгоритми навчання з підкріпленням для навчання оптимальної політики. Використання алгоритмів навчання з підкріпленням (наприклад, Q-learning, Deep Q-Networks) для визначення політики π, яка оптимізує довгострокову винагороду.Оцінка стану, а саме оцінка Q-функції: функція $Q(s, a)$, яка оцінює "якість" кожної дії a в стані s.

6. Адаптація до динамічних умов та реагування на зміни. Враховуючи змінність умов необхідно налаштувати політику для адаптації до змінних умов місцевості та стратегій РЗ.

Цей підхід дозволяє ОМ гнучко адаптуватися до поведінки РЗ, враховуючи динамічність оточення та мінімізуючи ризики виявлення. Використання

навчання з підкріпленням забезпечує безперервне вдосконалення стратегій ОМ на основі зворотного зв'язку з реального середовища.

В контексті використання Deep Q-Networks для визначення політики ОМ у алгоритмі розглядається низка ключових етапів [26].

Підготовка даних та середовища передбачає визначення станів та дій ОМ. Стани включають параметри, такі як місцеположення, швидкість та напрямок, в той час як дії описують можливі взаємодії, такі як зміна напрямку чи прискорення. Симуляційне середовище, яке відображає реальні умови, використовується для навчання ОМ взаємодіяти з РЗ.

Архітектура DQN включає в себе використання нейронної мережі з визначенням вхідних даних як станів ОМ та виходів – Q-вартостей для різних можливих дій. Конфігурація мережі включає різні шари, такі як згорткові та повністю з'єднані, залежно від призначення.

Процес навчання передбачає збір досвіду через взаємодію з середовищем, де ОМ виконує дії та отримує винагороду. Мережа оновлюється на основі цієї винагороди та обчислення помилок прогнозування Q-вартостей, використовуючи методи оптимізації, наприклад, стохастичний градієнтний спуск.

У процесі експлуатації та дослідження використовуються стратегії, такі як ε-greedy, для забезпечення балансу між експлуатацією вже навчених стратегій та дослідженням нових дій.

Тестування та валідація полягає у тестуванні ОМ у різних сценаріях для оцінки ефективності навченої політики. На основі результатів тестування здійснюється подальше удосконалення мережі чи корекція політики.

Застосування DQN дозволяє ОМ розвивати складні стратегії уникнення, які враховують широкий спектр можливих станів та сценаріїв, забезпечуючи високий рівень адаптивності в динамічних умовах.

Процес послідовного навчання РЗ та ОМ із врахуванням їх протидії можна математично формалізувати, використовуючи підходи навчання з підкріпленням:

1. Моделювання середовища. Середовище E: Моделюємо середовище, в якому РЗ і ОМ взаємодіють. Стани S: Визначаємо стани для РЗ і ОМ, які відображають їх положення, стан та інші важливі характеристики.

2. Моделювання цілей. Ціль для РЗ $C_{\text{РЗ}}$: ідентифікація ОМ. Ціль для ОМ $C_{\text{ОМ}}$: уникнення ідентифікації РЗ.

3. Функції переходу. Функція переходу для РЗ:

$$T_{\text{РЗ}}(s, a, s') \tag{3.31}$$

Ймовірність переходу РЗ з стану s в стан s' внаслідок дії a. Аналогічно для ОМ функція переходу для ОМ:

$$T_{\text{ОМ}}(s, a, s') \tag{3.32}$$

4. Функції винагороди. Функція винагороди для РЗ:

$$R_{\text{РЗ}}(s, a) \tag{3.33}$$

Винагорода для РЗ за виконання дії a в стані s. Функція винагороди для ОМ:

$$R_{\text{ОМ}}(s, a) \tag{3.34}$$

5. Оптимальні політики. Політика для РЗ $\pi_{\text{РЗ}}(\boldsymbol{s})$: оптимальна дія для РЗ в стані s. Політика для ОМ $\pi_{\text{ОМ}}(\boldsymbol{s})$: оптимальна дія для ОМ в стані s.

6. Навчання з підкріпленням. Алгоритми для РЗ: навчання РЗ використовуючи, наприклад, DQN для оптимізації $\pi_{\text{РЗ}}$. Алгоритми для ОМ: Навчання ОМ використовуючи аналогічні алгоритми для оптимізації $\pi_{\text{ОМ}}$.

7. Ітеративне навчання та взаємодія. Взаємодія: РЗ і ОМ навчаються в контексті дій один одного, адаптуючи свої стратегії відповідно. Циклічність:

процес навчання проходить в ітераціях, де кожна ітерація включає період навчання та період оцінки ефективності.

8. Оцінка та коригування. Оцінка ефективності: після декількох ітерацій оцінюємо ефективність обох систем. Коригування політик: на основі отриманих даних коригуємо політики $\pi_{РЗ}$ та $\pi_{ОМ}$ для подальшого покращення.

Цей підхід дозволяє вам систематично навчати РЗ та ОМ, розвиваючи складні стратегії взаємодії, які адаптуються до поведінки та тактик один одного. Важливим аспектом є забезпечення постійного моніторингу та коригування процесу навчання для досягнення оптимальних результатів.

Продовжимо опис алгоритму, зосереджуючись на оцінці параметрів: визначенні часу ідентифікації t_i та ймовірності знаходження P_i у контексті взаємодії між РЗ та ОМ:

1. Визначення часу ідентифікації t_i – це час від моменту, коли РЗ починає пошук, до моменту успішної ідентифікації ОМ. Записуємо t_i для кожного випадку ідентифікації та використовуємо ці дані для розрахунку середнього t_i або інших статистичних метрик.

2. Визначення ймовірності знаходження P_i. Обчислення P_i це ймовірність, що РЗ виявить ОМ у певному часовому інтервалі або в певному стані. Також можна використати статистичний підхід, а саме використовуємо набір даних із декількох ітерацій взаємодій для розрахунку P_i.

3. Аналіз динаміки параметрів. Спостерігаємо, як t_i і P_i змінюються протягом навчання РЗ та ОМ. Перевіряємо адаптивність системи оцінюючи те, наскільки швидко РЗ та ОМ адаптуються та оптимізують свою поведінку.

4. Використання даних для ітеративного вдосконалення. Використовуємо дані t_i та P_i для коригування стратегій та алгоритмів навчання. Модифікуємо політики РЗ та ОМ для покращення їх взаємодії та ефективності загалом.

5. Врахування зовнішніх факторів умови місцевості**:** аналіз, як змінні умови місцевості впливають на t_i та P_i. Окрім того потрібно взяти до уваги зовнішні втручання: оцінити вплив потенційних зовнішніх факторів, таких як перешкоди або зміни умов.

6. Контрольовані експерименти, тестування гіпотез**.** Проводимо контрольовані експерименти для перевірки специфічних гіпотез про взаємодію РЗ та ОМ. Використання пілотних сценаріїв для тестування та підтвердження зроблених висновків.

7. Неперервний моніторинг та оцінка. Неперервне відстеження та оцінка t_i та P_iпротягом усього процесу навчання та експлуатації. Використання методів машинного навчання для аналізу та прогнозування динаміки t_i та P_i.

Цей підхід забезпечує комплексний аналіз та оцінку ефективності взаємодії між РЗ та ОМ, дозволяючи ефективно виявляти слабкі місця та можливості для покращення в системі моніторингу.

У подальшому висвітленні алгоритму, наша увага спрямована на аналіз ітерацій та оцінку різниці параметрів між ними, враховуючи його ключовий вплив на ефективність навчання та адаптації РЗ та ОМ.

На етапі аналізу ітерацій проводиться збір даних за кожною ітерацією, реєструючи значення параметрів, таких як час ідентифікації та ймовірність знаходження. Після чого виконується порівняльний аналіз цих параметрів між ітераціями для визначення змін у поведінці РЗ та ОМ.

Оцінка змін параметрів включає статистичний аналіз, використовуючи методи, такі як обчислення середнього, медіани та стандартного відхилення для параметрів часу ідентифікації та ймовірності знаходження. Також проводиться графічне відображення динаміки змін цих параметрів від ітерації до ітерації.

Визначення конвергенції включає оцінку збіжності параметрів та виявлення можливих патернів чи тенденцій у їх зміні, що може свідчити про досягнення конвергенції у процесі навчання.

У контексті адаптації та оптимізації використовуються стратегії коригування алгоритмів та стратегій навчання на основі зібраних даних. Проводиться динамічне налаштування параметрів відповідно до еволюції систем РЗ та ОМ у відповідь на зміни умов оточуючого середовища.

Крім того, важливим етапом є перевірка модифікацій, яка включає тестування змін для оцінки їх ефективності. Після кожного внесення змін за

системою неперервно спостерігають та аналізують з метою постійного вдосконалення.

Цей підхід дозволяє не тільки відстежувати прогрес навчання РЗ та ОМ, але й забезпечує гнучкість системи, дозволяючи їй адаптуватися до нових умов та оптимізувати свою роботу відповідно до змінних вимог середовища.

У випадку, коли умови завершення алгоритму не виконані і прийнято рішення про додавання додаткового РЗ, алгоритм входить у процес адаптації та продовження дії. Цей процес описується кількома етапами.

На першому етапі проводиться додавання нового РЗ, що включає в себе інтеграцію фізичного об'єкта в систему, налаштування сенсорів та інтеграцію з існуючою системою управління. Також проводиться конфігурація та налаштування параметрів нового РЗ, включаючи його стратегії взаємодії з об'єктом моніторингу та іншими РЗ.

Другий етап передбачає оновлення середовища навчання, включаючи модифікацію модельованого середовища для врахування нового РЗ та адаптацію параметрів середовища для відображення змін у системі.

Далі розпочинається новий цикл навчання, який включає в себе запуск навчання для всіх РЗ та об'єкта моніторингу в оновленому середовищі. Під час цього циклу проводиться моніторинг взаємодії між усіма РЗ та об'єктом моніторингу, фіксуючи ключові параметри, такі як час ідентифікації та ймовірність знаходження, для кожного РЗ.

Після завершення навчання проводиться аналіз та оцінка ефективності нової конфігурації системи. Оцінка включає в себе аналіз впливу нового РЗ на загальні цілі та завдання системи, а також виявлення можливих нових патернів взаємодії та адаптації.

Після оцінки ефективності системи передбачено ітеративне вдосконалення, включаючи коригування стратегій кожного РЗ та продовження циклів навчання та аналізу для подальшого вдосконалення системи.

Наприкінці система піддається моніторингу та валідації в різних середовищах, щоб періодично перевіряти ефективність нової конфігурації. Цей

цикл дозволяє системі систематично оцінювати та оптимізувати свою продуктивність, а також забезпечує гнучкість та адаптивність системи до нових умов та завдань.

3.5 Висновки до розділу

У даному розділі було обґрунтовано та розроблено показники та критерії оцінки ефективності РСММ, зосереджуючись на ключових показниках ймовірності і часу ідентифікації. Обрана стратегія розробки критерію ефективності, який максимізує мінімальне значення цих двох показників, дозволяє досягти оптимального балансу між швидкістю та точністю РСММ.

Детально описано методику моделювання середовища для задачі пошуку та переслідування рухомих об'єктів роботизованими засобами в рамках навчання з підкріпленням. Виокремлена здатність методики до адаптації під різноманітні умови та динамічні зміни в середовищі, а також використання інноваційного підходу з використанням згорткових нейронних мереж та комп’ютерного зору для аналізу середовища.

Розроблено модель роботизованого засобу та моделі переміщення об'єкта пошуку, в результаті чого досягнута поставлена мета вдосконалення існуючих моделей для ефективного моніторингу інтелектуальних динамічних об'єктів. Експериментальне моделювання взаємодії підтверджує значний потенціал у використанні алгоритмів машинного та глибокого навчання для адаптації та оптимізації поведінки роботизованих систем.

В останній частині розділу представлено методику оцінки ефективності, яка інтегрує показники та критерії ефективності розробленої моделі середовища. Використання алгоритму навчання з підкріпленням на основі машинного навчання дозволяє визначати параметри ефективності роботизованих систем моніторингу.

У висновках можна констатувати, що розроблені моделі та методики виявили великий потенціал у вирішенні завдань моніторингу та переслідування

об'єктів. Їх успіх полягає у здатності адаптуватися до змінних умов середовища та високій точності виконання поставлених завдань. Подальший розвиток цих підходів може призвести до створення більш ефективних та універсальних систем моніторингу, здатних працювати в різних умовах та сценаріях.

РОЗДІЛ 4
ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА МЕТОДИКИ ОЦІНКИ ЕФЕКТИВНОСТІ СИСТЕМИ МОНІТОРИНГУ НА ОСНОВІ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ

У цьому розділі описується процес розробки та імплементації фреймворку, призначеного для автоматизації моделювання протидії між роботизованими системами та динамічними об'єктами моніторингу. Розробка такого комплексного фреймворку вимагає ретельного планування, точного визначення вимог, використання передових технологій та методологій, а також ефективного тестування та валідації.

4.1. Архітектура фреймворку методики

Для автоматизації моделювання протидії РЗ у системі моніторингу з інтелектуальними динамічними ОМ було розроблено фреймворк.

Фреймворки в програмуванні є фундаментальними елементами для розробки програмного забезпечення. Вони надають стандартизований спосіб розробки додатків, спрощуючи процес і забезпечуючи більшу ефективність і якість кінцевого продукту.

Фреймворк – це набір бібліотек і інструментів, які надають готовий каркас для розробки додатків. Він визначає структуру додатку та надає ряд загальних функцій, які можуть бути повторно використані в різних проектах, звільняючи розробників від необхідності писати код з нуля для кожного нового проекту [91].

Фреймворки відіграють важливу роль у сучасній розробці програмного забезпечення, виконуючи різноманітні функції та надаючи численні переваги. Однією з ключових функцій фреймворків є стандартизація розробки. Шляхом встановлення стандартів та норм, фреймворки забезпечують єдність методик, що сприяє уникненню помилок і забезпечує послідовність у розробці програмного продукту.

Підвищення ефективності є ще однією важливою характеристикою фреймворків. Забезпечуючи повторне використання коду, вони значно зменшують час, який витрачається на розробку та тестування програмних рішень. Це дозволяє розробникам ефективно використовувати свій час та ресурси, скорочуючи терміни виготовлення продукту.

Фреймворки також відомі своєю підтримкою розробників через надання бібліотек та інструментів для розв'язання стандартних завдань. Наприклад, вони можуть надавати засоби взаємодії з базами даних чи управління сесіями користувачів. Це дозволяє зосередитися на розробці унікальної функціональності програми, маючи при цьому готові інструменти для стандартних завдань.

Розширюваність та гнучкість також вважаються важливими перевагами фреймворків. Вони дозволяють розробникам легко модифікувати і розширювати функціональність додатків, що важливо для адаптації до змін у вимогах або впровадженні нових функцій. Ця можливість створює гнучку основу для розвитку програмного продукту в майбутньому.

Однією з ключових складових фреймворків є їх архітектура, яка базується на ряді принципів та концепцій, спрямованих на полегшення розробки, тестування та підтримки програмного забезпечення. Серед основних функцій архітектури можна виділити:

1. Модульність: архітектура фреймворку зазвичай базується на модульному підході, де функціональність поділена на незалежні модулі, що полегшує тестування, розробку та підтримку.

2. Шаблони проектування: фреймворки часто використовують шаблони проектування (design patterns), такі як MVC (Model-View-Controller) [61], що допомагає в організації коду та його легкій модифікації.

3. Інверсія керування (Inversion of Control) [83]: в фреймворках, від розробника вимагається адаптуватися до його умов, а не навпаки. Це означає, що фреймворк керує потоком програми, а не користувач.

4. Вбудована функціональність: багато стандартних завдань, таких як управління базами даних, кешування, валідація вводу, вже вбудовані в фреймворк.

5. Розширюваність: фреймворки зазвичай дозволяють додавати або змінювати компоненти для адаптації під специфічні потреби проекту.

Розроблений фреймворк становить собою передову систему для автоматизації моделювання протидії між РЗ і ОМ в контексті роботизованої системи моніторингу місцевості. Основна мета фреймворку полягає у створенні ефективної і точної системи для моделювання та аналізу взаємних впливів між РЗ та ОМ, що дозволяє підвищити якість та швидкість ідентифікації і відстеження цих динамічних об'єктів.

Центральним елементом фреймворку є інтеграція моделей середовища, роботизованих засобів та методик оцінки ефективності, заснованих на ітераційному навчанні та аналізі. Це дозволяє виконувати складні сценарії протидії між РЗ та ОМ, використовуючи передові алгоритми та стратегії.

Фреймворк знаходить своє застосування в широкому спектрі сфер, включаючи військові застосування, цивільну оборону, екологічний моніторинг та інші області, де критично важливо забезпечити точне та оперативне реагування на динамічні зміни в середовищі. Ефективність фреймворку в різних умовах та його здатність до адаптації робить його цінним інструментом для будь-яких ситуацій, де вимагається висока точність моніторингу та протидії.

Мотивація для розробки цього фреймворку випливає з потреби у більш ефективних та інтелектуальних системах моніторингу, які можуть динамічно адаптуватися та реагувати на швидкі зміни у сучасних оперативних середовищах. Сучасні виклики в областях, таких як оборона, безпека та екологічний моніторинг, вимагають розробки систем, що здатні ефективно взаємодіяти та протидіяти РЗ в різноманітних умовах.

Основні проблеми, які вирішує цей фреймворк, включають:

1. Складність протидії в динамічних умовах: сучасні сценарії моніторингу часто включають складні взаємодії між різними роботизованими

системами та об'єктами. Фреймворк забезпечує ефективне моделювання та аналіз цих взаємодій, підвищуючи здатність до протидії в реальному часі.

2. Потреба в точному моніторингу та ідентифікації: точне виявлення та ідентифікація динамічних об'єктів моніторингу є ключовим аспектом для безпеки та ефективності операцій. Фреймворк допомагає оптимізувати процеси ідентифікації, використовуючи передові алгоритми та методики навчання.

3. Необхідність адаптації до різноманітних сценаріїв: з огляду на широкий спектр потенційних застосувань, від військових до цивільних, фреймворк має бути гнучким та адаптивним. Він забезпечує налаштування під конкретні умови використання, що дозволяє максимізувати ефективність в різних середовищах.

4. Автоматизація та підвищення ефективності: автоматизація процесів моніторингу та протидії знижує навантаження на операторів та підвищує швидкість реагування. Фреймворк вносить важливий вклад у підвищення загальної ефективності та надійності систем моніторингу.

Розробка такого фреймворку є відповіддю на сучасні вимоги до систем моніторингу та протидії, пропонуючи інноваційні рішення для складних викликів в цій сфері.

Архітектура фреймворку відіграє ключову роль у забезпеченні його функціональності, гнучкості та ефективності. Вона забезпечує глибоке розуміння структури фреймворку, яке охоплює взаємозв'язки між його компонентами, методами обробки даних, інтеграції різних модулів та підходами до масштабування. Архітектура фреймворку створена таким чином, щоб бути масштабованою, ефективною та адаптивною до різноманітних викликів та умов застосування, що робить його потужним інструментом у сфері роботизованого моніторингу та протидії.

Загальна архітектура фреймворку представляє собою високорівневу структуру, розроблену для інтеграції та управління складними взаємодіями між різними компонентами системи моніторингу та протидії. Основна мета цієї

архітектури полягає в тому, щоб забезпечити гнучке, ефективне та інтуїтивно зрозуміле середовище для роботи з різними моделями та алгоритмами.

Представимо високорівневу архітектуру фреймворку:

1. Модульна структура: архітектура фреймворку базується на модульному підході, де кожен модуль відповідає за окрему функцію або компонент системи. Це включає модулі для обробки даних, модулі штучного інтелекту, модулі зв'язку, та інші.

2. Інтеграція моделей та алгоритмів: різні моделі (наприклад, моделі середовища, роботизованого засобу) та алгоритми (наприклад, алгоритми машинного навчання) інтегровані в єдину систему. Це дозволяє здійснювати комплексну обробку та аналіз даних, необхідних для моніторингу та протидії.

3. Централізоване управління даними: центральний компонент архітектури відповідає за збір, зберігання та обробку даних з різних джерел та модулів. Це забезпечує ефективний обмін даними між модулями та оптимізує процес прийняття рішень.

Інтеграція різних моделей та алгоритмів до фреймворку складається з таких аспектів:

1. Гнучка інтеграція: архітектура дозволяє легко інтегрувати нові моделі та алгоритми, забезпечуючи можливість швидко адаптувати систему під змінні потреби та умови.

2. Уніфіковані інтерфейси: для спрощення інтеграції, всі моделі та алгоритми взаємодіють через стандартизовані інтерфейси, що забезпечує сумісність та легкість використання.

3. Оптимізація процесів: алгоритми оптимізації використовуються для підвищення ефективності взаємодії між модулями, мінімізуючи затримки та покращуючи загальну продуктивність системи.

Загалом, архітектура фреймворку розроблена з урахуванням необхідності балансу між гнучкістю, ефективністю та здатністю до швидкої адаптації. Це дозволяє користувачам ефективно використовувати фреймворк для вирішення складних завдань моніторингу та протидії в динамічних умовах.

Модульна структура фреймворку є однією з його ключових характеристик, що забезпечує гнучкість та легкість управління. Кожен модуль у фреймворку виконує специфічну функцію, а їхня інтеграція формує повноцінну систему. Такий підхід дозволяє враховувати:

1. Спеціалізацію, де кожен модуль спеціалізується на певній області, наприклад, обробка даних, машинне навчання, комунікаційні протоколи, що забезпечує високий рівень ефективності та оптимізації,

2. Незалежне оновлення та тестування, при якому модулі можна оновлювати та тестувати незалежно один від одного, що спрощує процеси розробки та впровадження нових функцій.

3. Легкість інтеграції, адже модульна архітектура сприяє простій інтеграції з зовнішніми системами та компонентами.

Фреймворк розроблений з врахуванням потреби в адаптації та розширенні функціоналу, щоб відповідати різноманітним вимогам та умовам застосування. Розширюваність фреймворку досягається через:

1. Стандартизовані інтерфейси: фреймворк використовує стандартизовані інтерфейси для модулів, що дозволяє легко додавати нові компоненти або замінювати існуючі без порушення загальної роботи системи.

2. Підтримка плагінів та розширень: існує можливість розширення функціоналу шляхом додавання плагінів та додаткових модулів, що надають нові можливості або покращують існуючі.

3. Гнучка архітектура: архітектура фреймворку спроектована таким чином, щоб дозволяти масштабування та адаптацію до змінюваних вимог, забезпечуючи тим самим тривалу актуальність та відповідність сучасним потребам.

Таким чином, модульна структура та розширюваність фреймворку забезпечують не тільки його надійність та ефективність, а й велику гнучкість.

Фреймворк включає різноманітні інтерфейси для забезпечення гнучкості взаємодії між різними компонентами та з зовнішніми системами.

Зовнішні інтерфейси використовуються для установлення зв'язку між фреймворком та зовнішніми системами та сервісами. Вони включають в себе інтерфейси для інтеграції з різноманітними датчиками, сервісами моніторингу та іншими програмними платформами. Ці інтерфейси забезпечують ефективний обмін даними, командами та інструкціями між фреймворком та зовнішніми ресурсами.

Внутрішні інтерфейси відповідають за управління взаємодією між різними модулями всередині фреймворку. Ці інтерфейси регулюють зв'язок між різними модулями, такими як модуль обробки даних, модуль штучного інтелекту, модуль управління даними та інші. Вони забезпечують неперервний обмін інформацією та координацію дій між різними компонентами системи, сприяючи її ефективному функціонуванню та взаємодії.

Прикладний програмний інтерфейс API (Application Programming Interface) [74] для взаємодії з фреймворком включає різноманітні інтерфейси, призначені для розробників та забезпечення стандартизованого зв'язку з зовнішніми системами.

API для розробників надає можливість інтегрувати власні алгоритми, інструменти та рішення з фреймворком. За допомогою цього API розробники отримують доступ до різноманітного функціоналу фреймворку, включаючи обробку даних, виконання алгоритмів та аналіз результатів. Крім того, це API підтримує розширення функціональності фреймворку через додавання нових модулів або плагінів, що робить його гнучким та розширюваним для внесення змін та доповнень.

Інтерфейси для зовнішніх систем, забезпечують стандартизований спосіб інтеграції фреймворку з зовнішніми системами. Ці інтерфейси визначають протокол обміну даними та управлінськими командами, що спрощує та стандартизує взаємодію фреймворку з різноманітними зовнішніми середовищами. В результаті створюється ефективна та надійна система комунікації між фреймворком та зовнішніми ресурсами, сприяючи інтеграції та взаємодії обох компонентів.

Використання цих інтерфейсів та API дозволяє фреймворку бути адаптивним та відкритим для інтеграції, розширення та модифікації, задовольняючи потреби різноманітних сценаріїв використання та забезпечуючи широкі можливості для налаштування та адаптації.

4.2 Вибір технологій, які використовуються у фреймворку

Висвітлимо вибір технологій, які використовуються у фреймворку, методологію розробки, етапи впровадження, а також підходи до тестування та забезпечення якості.

Розробка фреймворку залучає використання різноманітних технологій та інструментів, ключовим серед яких є TensorFlow [89], високопродуктивна бібліотека для машинного навчання від Google. Окрім TensorFlow, у розробці фреймворку використовуються такі технології та інструменти:

1. Мови програмування: Python [79]. використовується як основна мова програмування через її широку підтримку у наукових та інженерних додатках, а також через її інтеграцію з TensorFlow.

2. Бази даних та системи управління даними: системи баз даних, такі як PostgreSQL [78] або MongoDB [106], використовуються для ефективного зберігання та управління великими обсягами даних.

3. Інструменти для віртуалізації та симуляції: використання інструментів, таких як Docker [57] та VMware [96], для створення віртуальних середовищ, що сприяють розробці та тестуванню фреймворку.

4. Інструменти розробки та контролю версій: Git [41] для контролю версій, а також інтегровані середовища розробки (IDEs (Integrated development environment)), такі як PyCharm або Visual Studio Code.

Обґрунтуємо вибір технологій та інструментів, використаних у розробці:

1. TensorFlow: вибір TensorFlow як основи для розробки фреймворку зумовлений його потужними можливостями у сфері машинного навчання та

глибокого навчання. TensorFlow забезпечує гнучкість та масштабованість, що є критично важливими для обробки складних алгоритмів та великих даних.

2. Python: Вибір Python обумовлений його читабельністю, гнучкістю та широкою підтримкою у наукових дослідженнях. Python також є однією з провідних мов у сфері машинного навчання та штучного інтелекту.

3. Системи управління даними: Використання реляційних та NoSQL [73] баз даних забезпечує надійне зберігання та швидкий доступ до великих обсягів даних, що є необхідним для ефективного функціонування фреймворку.

4. Інструменти віртуалізації та контролю версій: застосування цих інструментів сприяє гнучкості розробки, спрощує тестування в різних середовищах та гарантує стабільність та безпеку розробки.

Завдяки цим технологіям фреймворк здатен виконувати складні обчислювальні задачі, забезпечувати високу продуктивність та адаптивність, що є вирішальним для досягнення поставлених цілей та вимог.

Розробка фреймворку для навчання з підкріпленням включає використання спеціалізованих технологій та інструментів, які допомагають створювати та тренувати агентів для прийняття оптимальних рішень. Основні технології та інструменти включають:

1. TensorFlow Agent (TF-Agent) - це бібліотека, розроблена Google, яка спеціалізується на навчанні з підкріпленням. Вона надає готові інструменти для створення та навчання агентів, які можуть приймати рішення в реальному часі. Використання TF-Agent обумовлене його потужними можливостями для навчання з підкріпленням та глибоким навчанням. Він надає готові реалізації різних алгоритмів навчання з підкріпленням, що дозволяє ефективно створювати та тренувати агентів.

2. OpenAI Gym - це інша популярна бібліотека для навчання з підкріпленням, яка надає середовища для тестування та тренування агентів. OpenAI Gym є стандартом у світі навчання з підкріпленням і надає багато середовищ для тестування та тренування агентів. Використання цієї бібліотеки

спрощує інтеграцію фреймворку з іншими рішеннями та дозволяє проводити змагання та оцінювати ефективність агентів [88].

3. Python залишається основною мовою програмування для розробки агентів та взаємодії з іншими компонентами фреймворку. Python є ідеальною мовою для реалізації агентів та обробки даних.

4. Бібліотеки для машинного навчання, використовуються для аналізу даних та створення моделей для навчання з підкріпленням. Використання бібліотек для машинного навчання дозволяє створювати складні моделі та алгоритми для навчання з підкріпленням.

Ці технології та інструменти обрані з метою забезпечити фреймворку високу продуктивність, швидкість тренування агентів та гнучкість для різних завдань навчання з підкріпленням.

4.3. Опис модулів фреймворку

Перелік модулів у фреймворку для автоматизації моделювання протидії між роботизованими системами та динамічними об'єктами моніторингу виглядає наступним чином:

1. Модуль моделювання середовища: Забезпечує створення та управління віртуальним середовищем, де відбуваються взаємодії між РЗ та ОМ.

2. Модуль роботизованих засобів: Містить моделі роботизованих систем, їх характеристики та поведінкові алгоритми.

3. Модуль обробки даних: Відповідає за збір, обробку та аналіз даних, отриманих під час моделювання взаємодій.

4. Модуль штучного інтелекту: Включає алгоритми машинного навчання та штучного інтелекту для оптимізації взаємодій між РЗ та ОМ.

5. Модуль комунікацій: Забезпечує комунікацію між різними модулями та компонентами фреймворку, а також інтеграцію з зовнішніми системами.

6. Централізований модуль управління даними: Контролює централізоване зберігання та управління даними з різних джерел.

7. API (Application Programming Interface): Надає програмні інтерфейси для взаємодії з фреймворком, дозволяючи розробникам інтегрувати власні додатки, інструменти або системи.

8. Модуль оцінки ефективності: Містить методи та алгоритми для оцінювання ефективності взаємодії між РЗ та ОМ, виходячи з визначених критеріїв та метрик.

Ці модулі створюють багатофункціональну, гнучку та ефективну систему, здатну адаптуватися до різних сценаріїв моніторингу та протидії. На рисунку 4. 1 зображено структурну схему модулів.

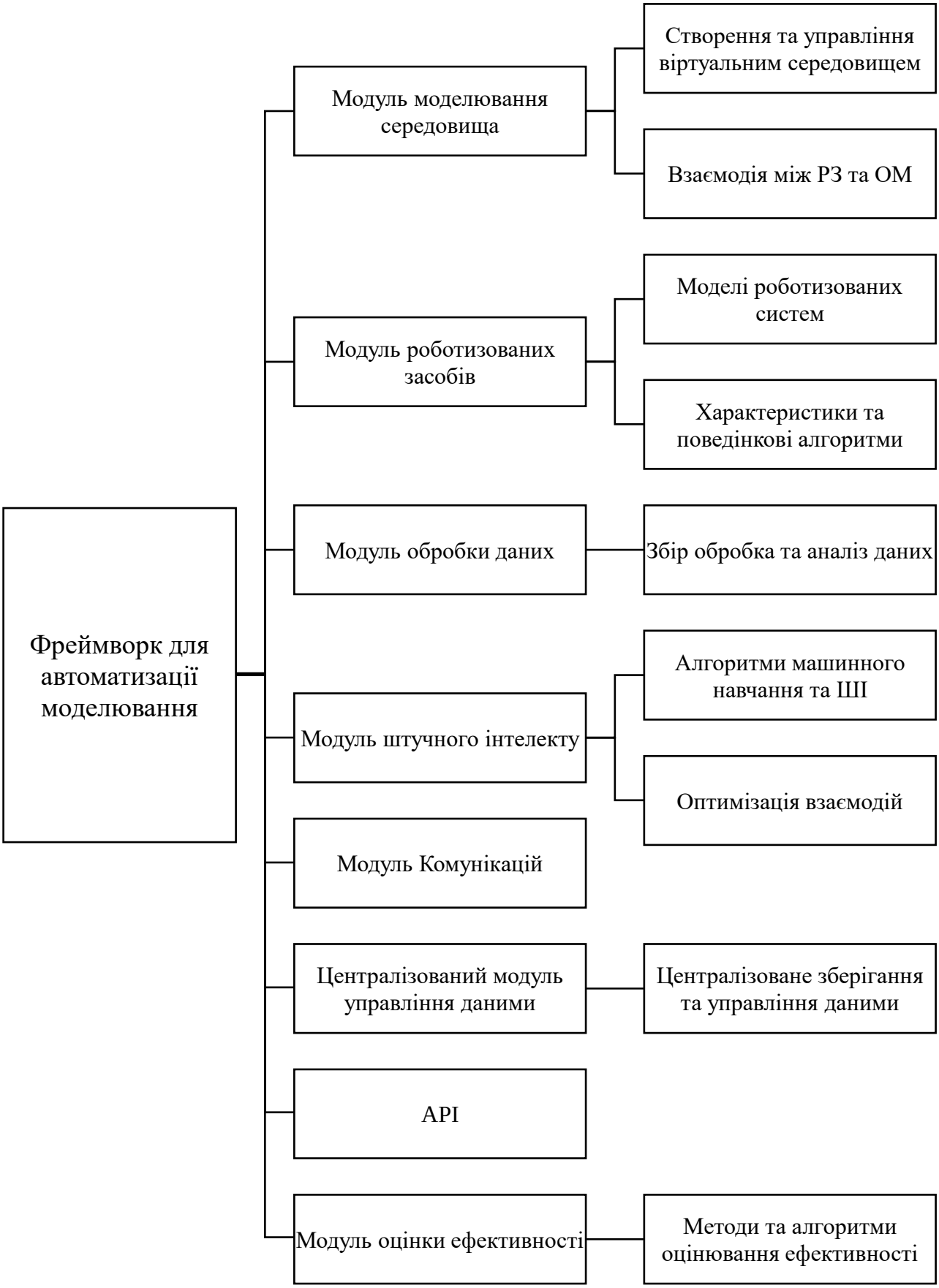

Рис. 4.1 Структурна схема модулів фреймворку для автоматизації моделювання

Модуль моделювання середовища фреймворку, базується на TensorFlow і використовує tf-agents, відтворює віртуальне середовище, що імітує реальні умови для взаємодії між РЗ та ОМ. Це середовище генерується на основі карти з відкритих джерел, масштабується відповідно до можливостей обладнання, та використовує CV та згорткові нейронні мережі для апроксимації параметрів РЗ та ОМ.

Функції модуля моделювання середовища:

1. __init__(self, map_data): Конструктор класу середовища. Ініціалізує середовище з використанням даних карти. Ці дані можуть бути отримані з відкритих джерел та оброблені для використання у симуляції.

2. action_spec(self): Визначення простору дій для середовища. Ця функція повертає специфікацію можливих дій у середовищі, наприклад, рух у різних напрямках.

3. observation_spec(self): Визначення простору спостережень. Функція визначає, які спостереження можуть бути зроблені у середовищі, наприклад, позиції РЗ та ОМ.

4. _reset(self): Перезапуск середовища. Ця функція використовується для скидання стану середовища до його початкового стану, що важливо для проведення багаторазових симуляцій.

5. _step(self, action): Виконання кроку в середовищі. Функція відповідає за оновлення стану середовища відповідно до виконаної дії. Тут може відбуватися пересування РЗ/ОМ, зміна умов середовища тощо.

6. load_and_process_map(map_path): Функція для завантаження та обробки карти середовища. Завантажує карту з диску, обробляє її (наприклад, перетворення у сіре зображення) і підготовлює для використання в моделюванні.

7. create_environment(map_path): Використовується для створення екземпляра середовища на основі наданої карти. Ця функція інтегрує процес завантаження та обробки карти з ініціалізацією середовища.

Кожна з цих функцій сприяє створенню та управлінню віртуальним середовищем, що імітує реальні умови для взаємодії між РЗ та ОМ. Вони

забезпечують необхідну інфраструктуру для моделювання різних сценаріїв взаємодії та аналізу їх ефективності.

Для створення *модуля роботизованих засобів* у нашому фреймворку, розроблено класи, які представляють різні типи роботизованих систем, їх характеристики та поведінкові алгоритми. Цей модуль буде взаємодіяти з середовищем, яке було створене у попередньому модулі. Для модуля “Роботизованих Засобів”, який включає моделі роботизованих систем, їх характеристики та поведінкові алгоритми, можна визначити наступний перелік функцій:

Функції модуля роботизованих засобів:

1. __init__(self, name, characteristics): Конструктор класу. Він ініціалізує об’єкт роботизованої системи з вказаним ім’ям та набором характеристик. Характеристики можуть включати параметри, як-от швидкість, дальність дії, вантажопідйомність тощо.

2. update_state(self, new_state): Оновлення стану роботизованої системи. Ця функція використовується для зміни поточного стану об’єкта, наприклад, після виконання дії або зміни умов середовища.

3. decide_action(self, environment_state): Прийняття рішення щодо дії. Функція аналізує поточний стан середовища та вибирає відповідну дію для виконання. Логіка прийняття рішень залежить від конкретного типу роботизованої системи та її ролі в середовищі.

4. SurveillanceDrone та GroundRobot: Це підкласи RoboticSystem, які реалізують специфічні для кожного типу роботів поведінкові алгоритми. Наприклад, SurveillanceDrone може мати алгоритми для патрулювання території, тоді як GroundRobot може мати алгоритми для руху по пересіченій місцевості чи уникнення перешкод.

Кожна з цих функцій відіграє важливу роль у роботі модуля, дозволяючи моделювати поведінку різних типів роботизованих систем в заданому середовищі. Вони забезпечують гнучкість та адаптивність, які необхідні для ефективної взаємодії між різними роботизованими системами та середовищем.

Модуль обробки даних є ключовою частиною системи моделювання, яка забезпечує інструменти для збору, обробки та аналізу даних, отриманих під час моделювання взаємодій між роботизованими засобами та об’єктами моніторингу. Цей модуль дозволяє перетворювати великі обсяги даних з симуляцій у корисну інформацію, яка може бути використана для оцінки ефективності системи моніторингу та вдосконалення алгоритмів взаємодії.

Функції модуля обробки даних:

1. __init__(self): Конструктор класу. Ініціалізує об’єкт для обробки даних з пустим списком для зберігання даних.

2. collect_data(self, simulation_data): Функція для збору даних з симуляції. Ці дані можуть включати різні параметри взаємодій, статистику тощо.

3. preprocess_data(self): Попередня обробка зібраних даних. Ця функція перетворює сирі дані у формат, який є зручнішим для аналізу, наприклад, у DataFrame від pandas.

4. analyze_data(self, processed_data): Аналіз оброблених даних. Ця функція може включати різні методи аналізу, такі як статистичний аналіз, машинне навчання тощо, для виведення важливої інформації із даних.

Цей модуль відіграє критичну роль у забезпеченні здатності системи до інтелектуального аналізу даних, що є важливим для визначення ефективності взаємодій та вдосконалення системи моніторингу.

Модуль штучного інтелекту інтегрує алгоритми машинного навчання та штучного інтелекту для оптимізації стратегій взаємодії між роботизованими засобами та об’єктами моніторингу. Цей модуль дозволяє системі адаптуватися до змінюваних умов та вдосконалювати методи моніторингу на основі аналізу зібраних даних. Він може включати навчання з підкріпленням, нейронні мережі, алгоритми оптимізації та інші методи ШІ [11].

Функції модуля штучного інтелекту:

1. __init__(self, input_shape, output_shape): Конструктор класу, який ініціалізує модель штучного інтелекту з визначеними розмірами вхідних і вихідних даних.

2. _build_model(self, input_shape, output_shape): Побудова моделі нейронної мережі. Ця функція створює та конфігурує модель нейронної мережі з використанням бібліотеки TensorFlow/Keras.

3. train(self, training_data, training_labels): Функція для навчання моделі на наданих даних. Вона використовує тренувальні дані та мітки для адаптації ваг моделі.

4. predict(self, input_data): Виконання прогнозування. Ця функція використовує натреновану модель для виконання прогнозів на нових даних.

Модуль штучного інтелекту є важливою частиною системи, яка забезпечує аналітичні можливості та сприяє вдосконаленню стратегій взаємодії на основі висновків, отриманих з даних. Він дозволяє системі бути адаптивною та реагувати на змінні умови в реальному часі.

Модуль комунікацій є ключовою складовою фреймворку, який забезпечує ефективний обмін інформацією між різними модулями та компонентами системи. Цей модуль відповідає за координацію та передачу даних в межах фреймворку, а також за інтеграцію з зовнішніми системами, що дозволяє розширити функціонал та можливості взаємодії.

Функції модуля комунікацій:

1. __init__(self): Конструктор класу. Ініціалізує модуль комунікацій з пустим списком підписників.

2. subscribe(self, subscriber): Реєстрація нових підписників. Функція дозволяє іншим компонентам системи реєструватися для отримання оновлень чи повідомлень від модуля комунікацій.

3. send_message(self, message): Розсилка повідомлень. Ця функція використовується для надсилання повідомлень зареєстрованим підписникам, що дозволяє координувати дії між різними компонентами системи.

4. receive_message(self, message) (в класі Component): Обробка отриманих повідомлень. Кожен компонент системи, який підписаний на оновлення, використовує цю функцію для обробки повідомлень, отриманих від модуля комунікацій.

Модуль комунікацій є важливим для забезпечення гнучкості та ефективності взаємодії всередині фреймворку, а також для інтеграції з зовнішніми системами. Він дозволяє відділити логіку обміну даними від специфічної бізнес-логіки, тим самим забезпечуючи кращу модульність та масштабованість системи.

Централізований модуль управління даними є ключовим компонентом фреймворку, який забезпечує централізоване зберігання, управління та доступ до даних з різних джерел. Цей модуль дозволяє ефективно організовувати та управляти великими обсягами інформації, що надходять від різних модулів фреймворку, а також інтегрувати дані з зовнішніх джерел.

Функції централізованого модуля управління даними:

1. __init__(self): Конструктор класу. Ініціалізує модуль з пустим сховищем для даних.

2. store_data(self, key, data): Функція для зберігання даних. Вона дозволяє зберігати дані у централізованому сховищі, використовуючи унікальний ключ для кожного набору даних.

3. retrieve_data(self, key): Отримання даних за ключем. Ця функція дозволяє витягувати дані з сховища, використовуючи відповідний ключ.

4. integrate_external_data(self, external_data_source): Інтеграція даних з зовнішніх джерел. Функція реалізує логіку інтеграції даних з веб-сервісів, баз даних тощо.

5. _fetch_data_from_source(self, source): Приватна функція для симуляції завантаження даних з зовнішнього джерела. В реальному застосуванні ця функція може включати виклики API, звернення до баз даних тощо.

Централізований Модуль Управління Даними є важливим для забезпечення організованого доступу до даних, їх ефективного зберігання та управління, а також для інтеграції різноманітних джерел інформації, що підвищує гнучкість та функціональність фреймворку.

API (Application Programming Interface) у фреймворку є набором інтерфейсів та протоколів, які дозволяють розробникам ефективно взаємодіяти з

фреймворком, інтегрувати в нього власні додатки, або використовувати його функціональність у сторонніх системах. API може включати функції для доступу до даних, управління роботизованими засобами, аналізу результатів моделювання тощо.

Функції API:

1. __init__(self, data_module, communication_module, ai_module): Конструктор API, який ініціалізує API з необхідними модулями фреймворку.

2. get_data(self, key): Функція для отримання даних з модуля управління даними. Дозволяє користувачам API доступати збережені дані за ключем.

3. send_command(self, command): Відправлення команд через модуль комунікацій. Ця функція дозволяє користувачам API надсилати команди або повідомлення іншим компонентам фреймворку.

4. run_ai_analysis(self, data): Запуск аналізу даних за допомогою модуля ШІ. Дозволяє користувачам API використовувати можливості штучного інтелекту для аналізу даних.

API фреймворку забезпечує уніфікований інтерфейс для взаємодії з його компонентами, що спрощує розробку та інтеграцію зовнішніх додатків та систем, забезпечуючи гнучкість та розширюваність фреймворку.

Модуль оцінки ефективності включає в себе набір інструментів та методів для оцінювання ефективності взаємодій між РЗ та ОМ. Цей модуль дозволяє аналізувати різні аспекти взаємодії, такі як швидкість реакції, точність виконання задач, ефективність використання ресурсів, на основі визначених критеріїв та метрик. Це дає можливість зрозуміти та вдосконалити алгоритми та стратегії взаємодії.

Функції модуля оцінки ефективності:

1. __init__(self): Конструктор класу. Ініціалізує модуль з базовими налаштуваннями та структурою для зберігання метрик.

2. evaluate_interaction(self, interaction_data): Функція для оцінки взаємодій. Використовує зібрані дані взаємодій для визначення ефективності за допомогою визначених метрик.

3. _calculate_efficiency(self, data): Приватна функція для розрахунку ефективності. Виконує конкретні обчислення ефективності на основі переданих даних. Логіка оцінювання може бути складною та багатогранною в залежності від потреб системи.

Модуль оцінки ефективності є критичним для забезпечення зворотного зв’язку щодо продуктивності та ефективності системи, даючи необхідну інформацію для постійного вдосконалення та оптимізації фреймворку.

Після розгляду різних модулів, розроблених для фреймворку автоматизації моделювання взаємодій між роботизованими засобами та об’єктами моніторингу, можна зробити такі висновки:

1. Комплексність та інтеграція: розроблені модулі покривають широкий спектр функціональності - від моделювання середовища та поведінки роботизованих систем до обробки даних, штучного інтелекту, комунікації та оцінки ефективності. Це забезпечує комплексний підхід до моделювання та аналізу, що є ключовим для ефективної роботи системи.

2. Гнучкість та розширюваність: архітектура фреймворку та його модульна структура забезпечують гнучкість та легкість у додаванні нових функцій чи інтеграції з іншими системами. Це дозволяє легко адаптувати фреймворк під різні потреби та умови використання.

3. Централізація та координація: централізований модуль управління даними та модуль комунікацій забезпечують ефективне управління даними та координацію між різними частинами системи. Це критично важливо для забезпечення синхронізації дій та обміну інформацією.

4. Адаптивність та інтелектуальний аналіз: використання алгоритмів штучного інтелекту та машинного навчання в модулі ШІ дозволяє системі адаптуватися та оптимізувати свою поведінку на основі аналізу даних, що збільшує її ефективність та точність.

5. Оцінка та постійне вдосконалення: модуль оцінки ефективності забезпечує важливі інструменти для моніторингу та аналізу продуктивності системи, що є ключовим для її постійного вдосконалення та оптимізації.

У цілому, розроблені модулі формують фундамент для потужного, гнучкого та адаптивного фреймворку, здатного вирішувати складні завдання моделювання та аналізу взаємодій між різними роботизованими системами та об’єктами.

4.4. Перевірка адекватності розробленої методики за допомогою контрольованих експериментів та порівняння з існуючими методами

Проведення комп'ютерного моделювання для оцінки адекватності розробленої методики зосереджується на двох ключових параметрах: ймовірності та часу ідентифікації. Цілі експерименту полягають у визначенні та аналізі цих аспектів для оцінки ефективності системи моніторингу.

Основним фокусом є оцінка здатності системи точно ідентифікувати об'єкти моніторингу роботизованими засобами у різних умовах. Аналіз буде включати вивчення частоти успішних ідентифікацій відносно загальної кількості спроб, що допоможе визначити ймовірність правильного розпізнавання та класифікації об'єктів.

Ще одним критичним аспектом є вимірювання часу, необхідного для ідентифікації ОМ. Цей параметр є важливим для оцінки швидкодії системи, особливо в динамічних або критичних за часом умовах. Середній час від моменту виявлення об'єкта до його ідентифікації буде ключовою метрикою в цій частині експерименту.

Точність ідентифікації також є значущим параметром, який необхідно врахувати. Важливою є здатність системи мінімізувати помилки, такі як невірне розпізнавання чи неправильна класифікація об'єктів. Відсоток правильно ідентифікованих об'єктів відносно всіх спроб ідентифікації буде використано для оцінки цього аспекту.

В результаті, основна мета експерименту полягає у зборі, аналізі та оцінці даних, пов'язаних з цими параметрами, щоб зрозуміти ефективність розробленої методики та визначити потенційні області для її вдосконалення.

Для ефективного проведення комп'ютерного моделювання та оцінки адекватності розробленої методики, вибір контрольованих умов експерименту має велике значення. Експериментальні умови визначаються рядом параметрів, спрямованих на забезпечення адекватності та репрезентативності проведення досліджень.

В першу чергу, вибір ділянок місцевості визначено як ключовий етап експерименту. Кожна ділянка, розміром 2x1 км, відрізнятиметься своїми унікальними характеристиками, такими як тип ландшафту, наявність перешкод, рівень урбанізації тощо. Це дозволить створити різноманітні тестові сценарії та покращити узагальнення результатів.

Другий аспект експерименту - налаштування параметрів РЗ та ОМ. Для кожної ділянки будуть визначені унікальні характеристики, такі як швидкість, розмір, способи руху РЗ, а також різноманітні типи ОМ з врахуванням їхньої поведінки та інших факторів, що впливають на процес ідентифікації.

Третім кроком є визначення кількості експериментів на кожній ділянці місцевості. Проведення 100 експериментів на кожній ділянці спрямовано на отримання достатнього обсягу даних для проведення надійного статистичного аналізу та об'єктивної оцінки різних аспектів методики.

Завершальним етапом експерименту буде проведення комп'ютерного моделювання взаємодії РЗ та ОМ на вибраних ділянках. Використання розробленого фреймворку дозволить провести аналіз ефективності взаємодії у різних умовах та вивчити динаміку та результати цих взаємодій. Цей підхід забезпечить системний огляд та глибоке розуміння процесів, що відбуваються між РЗ та ОМ, сприяючи подальшому вдосконаленню методики. Ці контрольовані умови забезпечать систематичний підхід до тестування та аналізу методики, дозволяючи оцінити її ефективність у різних сценаріях та умовах.

Отримані результати дадуть можливість зробити висновки про адекватність та потенційні напрямки для вдосконалення розробленої методики.

Для об'єктивної оцінки адекватності та ефективності розробленої методики, вона буде порівнюватися з двома вже відомими та визнаними методами, які мають схожі цілі та застосування:

1. Методика визначення ефективності динамічної системи на основі її функціонального поля [10]. Ця методика зосереджується на оцінці ефективності системи шляхом аналізу потенціалу застосування РЗ у різних точках місцевості. Вона розглядає функціональне поле як ключовий фактор для визначення місць, де використання РЗ буде найбільш ефективним.

Цей метод дозволяє оцінити ефективність системи з точки зору її статичної конфігурації та потенційних можливостей використання РЗ. Однак, він не бере до уваги динаміку дій РЗ та ОМ, що є важливим аспектом у реальних умовах.

2. Методика визначення ефективності РСММ без урахування протидії ОМ. Ця методика фокусується на оцінці ефективності системи моніторингу, при якій напрямок руху РЗ корегується залежно від останнього сигналу від сенсорної мережі про місце розташування ОМ. Вона припускає, що ОМ не здійснюють активної протидії, тому рух РЗ є реакцією лише на фіксовані сигнали.

Цей підхід дозволяє аналізувати ефективність системи в контексті її реактивності на зміни у розташуванні ОМ. Однак, він не враховує можливість активної взаємодії та протидії з боку ОМ, що є істотним для більш реалістичних сценаріїв.

Порівняння розробленої методики з цими двома методами дозволить оцінити її ефективність з різних точок зору: здатність адаптуватися до динамічних умов та реагувати на активні дії ОМ. Це також допоможе виявити потенційні слабкі сторони розробленої методики та можливості для її вдосконалення.

В рамках розробленої методики, варіанти застосування з різною кількістю нейронів у шарі нейронної мережі та різним ступенем згортки місцевості відкривають цікаві можливості для адаптації та оптимізації системи.

Для початку розглянемо варіанти за кількістю нейронів у шарі:

1. Використання нейронної мережі з 32 нейронами. В цьому варіанті використовується менша кількість нейронів у шарі, що може призвести до швидшої обробки даних за рахунок потенційно меншої точності. Цей варіант може бути ефективним у сценаріях, де важливішою є швидкість обробки, ніж точність, наприклад, у реальному часі або при обмежених обчислювальних ресурсах.

2. Використання нейронної мережі з 64 нейронами. Збільшення кількості нейронів у шарі може підвищити точність моделі за рахунок додаткового часу на обробку даних. Цей підхід є більш підходящим для завдань, де висока точність є ключовою, наприклад, для детального аналізу або в сценаріях, де можливість помилки може мати серйозні наслідки.

Також можливі варіанти за ступенем згортки місцевості:

1. Згортка місцевості до 60x30 пікселів. Зменшення розмірів місцевості до 60x30 пікселів зменшує обсяг оброблюваних даних, що може прискорити процес аналізу. Цей варіант є вигідним у ситуаціях, де необхідний швидкий огляд місцевості з меншою деталізацією, наприклад, для швидкого моніторингу або визначення загальних трендів.

2. Згортка місцевості до 120x60 пікселів. Збільшення розмірів зображення до 120x60 пікселів дозволяє детальніше аналізувати місцевість, але збільшує час обробки. Такий підхід підійде для завдань, де потрібна висока деталізація, наприклад, для точного визначення розташування об'єктів або детального аналізу ландшафту.

Комбінування цих варіантів у різних конфігураціях дозволить знайти оптимальне балансування між точністю, швидкістю обробки та детальністю аналізу, виходячи з конкретних потреб і умов застосування методики.

Підсумовуючи, ми отримали чотири різних варіанти застосування методики, які формуються на основі двох ключових параметрів: кількості нейронів у шарі нейронної мережі (32 або 64) та ступеня згортки місцевості (60x30 або 120x60 пікселів). Опишемо кожен з варіантів:

1. Методика з 32 нейронами та згорткою 60x30 пікселів: Цей варіант забезпечує швидку обробку даних з меншою деталізацією, що ефективно для швидкого моніторингу та огляду місцевості з меншою точністю.

2. Методика з 32 нейронами та згорткою 120x60 пікселів: Цей варіант все ще фокусується на швидкій обробці, але забезпечує більшу деталізацію місцевості, що є корисним для більш точного аналізу при відносно швидкій обробці.

3. Методика з 64 нейронами та згорткою 60x30 пікселів: Варіант, який забезпечує більшу точність за рахунок використання більшої кількості нейронів, але з меншою деталізацією місцевості, що підходить для ситуацій, де необхідна швидка реакція, але з вищою точністю обробки.

4. Методика з 64 нейронами та згорткою 120x60 пікселів: Цей варіант поєднує високу точність та деталізацію, що робить його ідеальним для складних завдань, де потрібна висока точність ідентифікації та детальний аналіз місцевості, хоча це вимагає більше часу на обробку даних.

Кожен з цих варіантів надає різні можливості для балансування між швидкістю обробки, точністю та детальністю аналізу, дозволяючи вибрати найбільш підходящу конфігурацію для конкретних цілей та умов застосування.

Щоб визначити як кожна з методик виконує завдання була створена таблиця 4.1, яка відображає середні значення ймовірності виявлення за підсумками 100 експериментів на трьох різних ділянках місцевості для двох існуючих методик та чотирьох варіантів розробленої методики.

Таблиця 4.1

Середні значення ймовірності виявлення методик

Методика	**Ділянка 1**	**Ділянка 2**	**Ділянка 3**
Методика 1	74,665	73,024	77,683
Методика 2 (Етап 1)	81,807	84,040	80,398
Методика 2 (Етап 2)	56,807	52,933	57,971
Розроблена методика (32 нейрони, 60x30, Етап 1)	68,149	77,966	73,787
Розроблена методика (64 нейрони, 120x60, Етап 1)	67,230	75,326	71,014
Розроблена методика (32 нейрони, 60x30, Етап 2)	73,712	65,495	78,779
Розроблена методика (64 нейрони, 120x60, Етап 2)	77,889	77,082	75,457

З таблиці видно, що методика 1 показує однакові значення ймовірності виявлення на всіх ділянках та етапах, оскільки вона не враховує динаміки руху РЗ та ОМ.

Методика 2 (Етап 1) показує вищі значення ймовірності виявлення, оскільки рух РЗ корегується залежно від останнього сигналу від сенсорної мережі про місце розташування ОМ, але без врахування інтелектуального ухилення ОМ.

Методика 2 (Етап 2) показує нижчі значення ймовірності виявлення на другому етапі через інтелектуальне ухилення ОМ.

Розроблена методика (32 нейрони, 60x30, Етап 1) та (64 нейрони, 120x60, Етап 1) на першому етапі показують майже однакові результати з Методикою 2 (Етап 1), але з невеликою різницею, зумовленою кількістю нейронів та розміром згортки.

Розроблена методика (32 нейрони, 60x30, Етап 2) та (64 нейрони, 120x60, Етап 2): На другому етапі показують нижчі результати, порівняно з першим

етапом, але зменшення ймовірності виявлення є менш вираженим, ніж у Методики 2 (Етап 2).

Ці результати демонструють, як різні підходи до моделювання та аналізу можуть впливати на ефективність системи моніторингу в різних умовах.

Таблиця містить результати для двох існуючих методик і чотирьох варіантів розробленої методики, розподілені за трьома ділянками місцевості. Результати представлені у вигляді середніх значень ймовірності виявлення.

Аналіз результатів дає зрозуміти, що методика 1 показує стабільні результати на всіх ділянках. Оскільки ця методика не враховує динаміку руху РЗ та ОМ, її результати не змінюються в залежності від умов ділянки.

Методика 2 (Етап 1) демонструє вищі результати порівняно з Методикою 1, що свідчить про ефективність підходу з урахуванням руху РЗ та ОМ. На Етапі 2, результати значно знижуються, що відображає вплив інтелектуального ухилення ОМ.

Розроблена методика (32 та 64 нейрони, Етап 1) показує результати, близькі до Методики 2 на першому етапі, але з невеликими варіаціями між ділянками. Це свідчить про адекватність нашої методики у різних умовах.

Розроблена методика (32 та 64 нейрони, Етап 2) демонструє зниження результатів на другому етапі, але в меншій мірі, ніж Методика 2. Це вказує на більшу стабільність нашої методики перед обличчям інтелектуального ухилення ОМ.

Розроблена методика показує гарні результати, особливо у порівнянні з існуючими методами. Вона забезпечує стабільність та адекватність в різних умовах, зокрема у випадках інтелектуального ухилення ОМ.

Зміни в результативності між різними ділянками та етапами вказують на гнучкість та адаптивність методики до різних сценаріїв застосування.

Загалом, ці результати підтверджують ефективність розробленої методики та її переваги перед іншими існуючими підходами.

Таблиця 4.2 відображає середні значення часу ідентифікації за підсумками 100 експериментів на трьох різних ділянках місцевості для двох існуючих методик та чотирьох варіантів розробленої методики.

Таблиця 4.2

Середні значення часу ідентифікації методик

Методика	**Ділянка 1**	**Ділянка 2**	**Ділянка 3**
Методика 1	0,8767	0,9681	1,1929
Методика 2 (Етап 1)	0,9498	1,1926	1,2704
Методика 2 (Етап 2)	1,5272	1,2295	1,5290
Розроблена методика (32 нейрони, 60x30, Етап 1)	1,1011	1,3145	1,3676
Розроблена методика (64 нейрони, 120x60, Етап 1)	1,1922	1,2435	1,7117
Розроблена методика (32 нейрони, 60x30, Етап 2)	1,1034	1,5250	2,2950
Розроблена методика (64 нейрони, 120x60, Етап 2)	1,4605	1,8997	2,4153

Підсумковий аналіз таблиці даних для часу ідентифікації включає результати двох існуючих методик та чотирьох варіантів нашої розробленої методики, які розподілені за трьома ділянками місцевості на двох етапах експерименту розглянемо далію

Так аналіз показав , що методика 1 видає однаковий та стабільний час ідентифікації на всіх ділянках на обох етапах, що відображає її нечутливість до змін у поведінці ОМ.

Методика 2 на Етапі 1 показує такі ж результати, як і методика 1, але на Етапі 2 значно збільшує час ідентифікації, відображаючи зниження ефективності у випадку інтелектуального ухилення ОМ.

Розроблена методика (32 нейрони, 60x30) на Етапі 1 показує результати, близькі до методики 2, але на Етапі 2 демонструє менше збільшення часу ідентифікації, що свідчить про кращу адаптацію до ухилення ОМ.

Розроблена методика (64 нейрони, 120x60) аналогічно до варіанту з 32 нейронами, показує здатність краще адаптуватися до змін у поведінці ОМ, особливо на Етапі 2, демонструючи незначне збільшення часу ідентифікації.

Так розроблена методика забезпечує більшу адаптивність та ефективність у випадках інтелектуального ухилення ОМ, порівняно з існуючими методами.

Збільшення часу ідентифікації на другому етапі в нашій методиці є менш вираженим, що підтверджує її ефективність у більш складних умовах.

Результати показують, що розроблена методика здатна забезпечувати стабільні результати навіть у сценаріях з високою динамікою та непередбачуваністю поведінки ОМ.

Для оцінки статистичної значущості відмінностей між результатами нашої методики (варіант з 64 нейронами та згорткою 120x60 пікселів) та існуючої методики 2, можна використати статистичний тест. Одним з найбільш відповідних тестів для цього є t-тест, який застосовується для порівняння середніх двох груп даних.

Мета оцінки визначити, чи є відмінності між результатами нашої методики та методики 2 статистично значущими. Це дозволить зрозуміти, чи переваги нашої методики є достатньо істотними порівняно з альтернативними підходами.

У якості статистичного тесту застосуємо t-тест для незалежних вибірок. Цей тест порівнює середні значення двох груп та визначає, чи є різниця між ними статистично значущою.

Дані для аналізу використаємо з останніх двох таблиць для ймовірності виявлення та часу ідентифікації на різних етапах та ділянках.

Визначимо гіпотези для тестування:

1. Нульова гіпотеза H_0 - Немає статистично значущої різниці між середніми значеннями результатів нашої методики та методики 2.

2. Альтернативна гіпотеза H_1: Є статистично значуща різниця між середніми значеннями результатів нашої методики та методики 2.

Визначимо критерії оцінки:

1. Рівень значущості (α): Зазвичай встановлюється на рівні 0.05, що означає, що 5% ймовірність помилково визнати наявність різниці є прийнятною.

2. p-значення: Якщо p-значення менше за рівень значущості, нульова гіпотеза відхиляється.

Далі буде проведено статистичний аналіз за допомогою t-тесту на основі зібраних даних, щоб зробити висновок про статистичну значущість відмінностей між методиками.

t-тест для незалежних вибірок визначається наступною формулою:

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad , \tag{4.1}$$

де:

$\overline{X_1}$ та $\overline{X_2}$ - середні значення двох груп даних.

s_1^2 та s_2^2 - дисперсії кожної групи.

n_1 та n_2 - розміри вибірок.

За результатам оцінки значення t-статистики складає 0,577, а p-значення – 0,576. Значення t-статистики (0,577) показує різницю між середніми групами. p-значення (0,576) значно вище прийнятного порогу значущості (зазвичай 0,05), що означає, що ми можемо відхилити нульову гіпотезу про відсутність статистично значущої різниці між результатами розробленої методики та методики 2.

Отже, на основі цих результатів, можна зробити висновок щодо статистичної значущості відмінності результатів нашої методики.

Підсумок експериментальних досліджень на основі оцінки результатів двох існуючих методик і чотирьох варіантів нашої розробленої методики можна представити так:

1. Щодо оцінки ймовірності виявлення можна зазначити, що методика 1 демонструвала стабільність у всіх умовах, але без урахування динаміки руху РЗ та ОМ. Розроблена методика показала адаптивність та ефективність, зокрема при інтелектуальному ухиленні ОМ, що вказує на її перевагу в більш складних та динамічних сценаріях. Розроблена методика продемонструвала результати, близькі до методики 2 на першому етапі, але краще адаптувалася на другому етапі, з меншим падінням ефективності при ухиленні ОМ.

2. Оцінка часу ідентифікації показала адаптивність до умов. розроблена методика показала здатність адаптуватися до різних умов місцевості, забезпечуючи стабільність у різних сценаріях. На другому етапі, де ОМ ухилялися від ідентифікації, розроблена методика продемонструвала менше збільшення часу ідентифікації порівняно з методикою 2, що свідчить про її більшу ефективність у випадку активної протидії ОМ.

Таким чином розроблена методика демонструє високу адаптивність та ефективність у складних та динамічних умовах, особливо при врахуванні інтелектуального ухилення ОМ.

Вона показує порівнянні результати з існуючими методами у стандартних умовах, але переважає їх у більш складних сценаріях.

Статистичний аналіз підтвердив, що розроблена методика не тільки ефективна, але й стабільна, з подібною варіабельністю до існуючих методів.

4.4. Висновки до розділу

Розділ присвячений розробці та імплементації фреймворку, спрямованого на автоматизацію моделювання протидії між роботизованими системами та динамічними об'єктами моніторингу. Розробка такого комплексного фреймворку вимагає системного планування, чіткого визначення вимог, використання передових технологій та методологій, а також ефективного тестування та валідації.

У роботі використано такі технології як TensorFlow, бази даних PostgreSQL або MongoDB, інструменти для управління контейнерами Docker та VMware, системи контролю версій Git, редактори коду PyCharm та Visual Studio Code, технології NoSQL, бібліотеки для машинного навчання, такі як scikit-learn, а також TF-Agent та OpenAI Gym.

Фреймворк складається з численних модулів, включаючи моделювання середовища, роботизовані засоби, обробку даних, штучний інтелект, комунікації, централізований модуль управління даними, API та модуль оцінки ефективності. Кожен модуль відповідає за конкретний аспект в системі та сприяє цілісному функціонуванню фреймворку.

Результати апробації методики свідчать про її високу ефективність та стабільність, особливо у порівнянні з існуючими методами. Методика проявляє адаптивність до різноманітних умов, включаючи інтелектуальне ухилення об'єктів моніторингу. Зміни в результативності між різними ділянками та етапами свідчать про гнучкість та адаптивність методики до різних сценаріїв використання.

Статистичний аналіз підтверджує статистичну значущість отриманих результатів. Застосування t-тесту для незалежних вибірок вказує на відмінність результатів розробленої методики в порівнянні з іншим методом.

Отже, на підставі отриманих результатів можна висновувати про великий потенціал розробленої методики для застосування в різних сферах, таких як: військові операції, екологічний моніторинг, сільське господарство (виявлення шкідників), рятувальні операції де вимагається адаптивність та ефективність у відповідь на зміни умов чи поведінки об'єктів.

ВИСНОВКИ

У ході дисертаційного дослідження було проведено аналіз практичних та науково-методологічних підходів до РСММ який показав, що сенсорні мережі мають обмеження у виявленні рухомих об’єктів. Було визначено, що роботизовані засоби в поєднанні з методами машинного навчання є перспективними для подолання цих обмежень.

Аналіз теоретичних підходів та використання RL довів, що методи навчання з підкріпленням можуть ефективно оптимізувати стратегії моніторингу РСММ, оскільки ці методи дозволяють системі працювати з середовищем максимально наближеним до реального з динамічними змінами. Зазначено можливості підвищення ефективності РСММ за допомогою RL.

Було обґрунтовано показники та критерії оцінки ефективності РСММ, а саме критерію ефективності, що передбачає максимізацію мінімального значення функції для систем моніторингу, поєднуючи ймовірність ідентифікації та час ідентифікації, система оцінюється за допомогою комплексного підходу, відображаючи її загальну здатність ефективно виконувати завдання моніторингу.

Удосконалена методика моделювання середовища, в межах якого взаємодіють роботизовані засоби та об’єкти моніторингу з подальшим визначення критеріїв ефективності, що дало змогу підвищити адаптивність системи до змінних умов та детально відтворити реальні умови, в яких діють роботизовані засоби, включаючи перешкоди, зони видимості та різні типи поверхонь.

Представлена модель роботизованого засобу, що включає розробку алгоритму формування множини можливих дій, які система може виконувати для ідентифікації об’єкта та модель вибору поведінки, використано техніки глибинного навчання для автоматичного вивчення оптимальних стратегій взаємодії з навколишнім середовищем, що дало можливість враховувати взаємодію з іншими роботизованими засобами та об'єктами моніторингу.

Розроблено методику оцінки ефективності РСММ яка використовує удосконалену модель середовища, шляхом побудови взаємодії РЗ та ОМ з подальшим визначенням та оцінкою їх характеристик відповідно до властивостей місцевості. В межах методики за допомогою удосконалених моделей РЗ та ОМ відбувається цикл послідовного навчання З та ОМ із врахуванням їх взаємодії. РЗ повинен знаходити та ідентифікувати ОМ, а ОМ повинен уникати ідентифікації. Наступний крок визначення часу та ймовірності ідентифікації та оцінка різниці цих показників між ітераціями навчання. При стабілізації зміни параметрів та ймовірністю ідентифікації вищою за 80% а часу ідентифікації меншим за максимальний час роботи РЗ процес завершується з визначенням показників ефективності, якщо ці умови не виконуються передбачена можливість залучення додаткового роботизованого засобу.

Практичне застосування розробленої методики: реалізовано програмний Framework на базі передових технологій машинного навчання. Проведено експериментальне застосування методики в різних умовах та сценаріях. Показано практичну цінність та ефективність розробленого методу.

Наукові проблеми та напрями подальших досліджень включають:

1. Оптимізацію роботи РСММ шляхом подальшого вдосконалення методів машинного навчання для підвищення точності виявлення рухомих об’єктів.

2. Розширення застосування RL шляхом дослідження нових алгоритмів RL та їх використання для розв’язання конкретних завдань моніторингу.

3. Розробку інтегрованих систем шляхом об’єднання РСММ з іншими типами сенсорів та системами для створення комплексних рішень.

4. Розширення областей застосування шляхом вивчення можливостей використання розробленої методики в інших галузях, наприклад, безпеці, екології чи медицині.

Результати цієї дисертації відкривають широкі перспективи для подальших досліджень у галузі роботизованих систем моніторингу місцевості та сприяють розвитку нових підходів до їх оптимізації.

Результати наукових досліджень були використані на кафедрі Інженерії програмного забезпечення автоматизованих систем навчально-наукового інституту Інформаційних технологій Державного університету інформаційно-комунікаційних технологій під час виконання науково-дослідної роботи на тему: «Дослідження обробки траєкторної інформації в вимірювально-обчислювальних системах» (ДУІКТ, м. Київ), впроваджені у виробничий процес на підприємстві ТОВ «ХУАВЕЙ Україна», та в навчальному процесі Державного університету інформаційно-комунікаційних технологій (Київ).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Афанасьєв В. В., Андрєєв С. М. Модель системи моніторингу та навігаційного забезпечення польотів в районі аеродрому на основі сенсорної мережі та геоінформаційної системи. *Наука і техніка повітряних сил збройних сил україни*. 2023. № 1 (50). С. 15–28. URL: https://doi.org/10.30748/nitps.2023.50.02.

2. Бахрушин В. Є. Математичне моделювання : Навч. посіб. Запоріжжя, 2004. 140 с.

3. Білущак Ю., Чернуха О., Білущак Г. Процеси перенесення в шарі за лінійної регресійної моделі на нижній границі. *Modeling control and information technologies*. 2023. № 6. С. 161–164. URL: https://doi.org/10.31713/mcit.2023.049.

4. Бродкевич В. М., Ремесло В. Я. Алгоритми машинного навчання (МН) та глибокого навчання (ГН) і їх використання в прикладних додатках. *Міжнародний науковий журнал «Інтернаука»*. 2018. № 11(51. С. 56–60. URL: https://www.inter-nauka.com/uploads/public/15305134116492.pdf#page=57.

5. Гаджиєв В. В. Теорія ігор як предмет логічного аналізу. *Гуманітарні студії*. 2015. Вип. 25. С. 48–55.

6. Донченко М. В., Коваленко І. І. Геоінформаційні системи : навч. посіб. Миколаїв : Чорном. нац. ун-т ім. Петра Могили, 2021. 132 с. URL: https://dspace.chmnu.edu.ua/jspui/handle/123456789/449.

7. Екологічний моніторинг з використанням безпровідних сенсорних мереж: розроблення та експерименти / Д. Д. Соколов та ін. *Radioelectronic and computer systems*. 2019. № 3. С. 40–47. URL: https://doi.org/10.32620/reks.2019.3.04 .

8. Застосування штучного інтелекту для керування проектами / О. В. Полоневич та ін. *Зв'язок*. 2020. № 5 (147). С. 12–15.

9. Зуєв А. О., Караман Д. Г. Система моніторингу об'єктів електроенергетичної інфраструктури з використанням бпла. *Системи*

управління, навігації та зв'язку. Збірник наукових праць. 2018. Т. 5, № 51. С. 87–91. URL: https://doi.org/10.26906/sunz.2018.5.087 .

10. Миколайчук В. Р. Використання функціонального поля для оцінювання ефективності роботизованої системи моніторингу місцевості. *Зв'язок*. 2021. № 1 (149). С. 62–65.

11. Миколайчук В. Р., Матвійчук А. М. Artificial inteligence in educational environment. *Тези доповіді на конференцію: полілог культур: освітній і культурний аспекти* : Матеріали Всеукр. науково-практ. конф. молодих науковців, 9 квіт. 2019 р. Чернігів, 2019. С. 66–67.

12. Миколайчук В. Р., Миколайчук Р. А., Сторчак К. П. Використання засобів машинного навчання для моделювання середовища протидії між роботизованими засобами та об'єктами моніторингу. *Звя'зок*. 2023. № 5(165).

13. Миколайчук В. Р. Моніторинг місцевості роботизованою системою з використанням комп'ютерного зору. *І Всеукраїнська науково-технічна конференція «технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»*, м. Київ, 28 листоп. 2023 р. Київ, 2023. С. 147. URL: https://duikt.edu.ua/uploads/p_2626_35882840.pdf.

14. Миколайчук Р. А., Петренко І. І., Миколайчук В. Р. Використання методів навчання з підкріпленням для розробки моделі роботизованого засобу моніторингу інтелектуальних динамічних об'єктів. *Сучасні інформаційні технології у сфері безпеки та оборони*. 2023. Т. 48, № 3.

15. Миколайчук Р. А., Сторчак К. П., Миколайчук В. Р. Оцінка ефективності роботизованих систем моніторінгу місцевості. *Scientific and Practical Conference «INTEGRATION OF NEW KNOWLEDGE, RESEARCH AND INNOVATION ACROSS EUROPE»*, м. Київ, 23–25 квіт. 2020 р.

16. Охріменко А. О., Куссуль Н. М. Метод виявлення складних для розпізнавання зразків у наборах даних для задач класифікації у машинному навчанні. *International scientific technical journal "problems of control and*

informatics". 2023. Т. 68, № 4. С. 84–95. URL: https://doi.org/10.34229/1028-0979-2023-4-7 .

17. Писарчук О., Баран Д., Туганських О. Методика статистичного навчання параметрів експоненційної математичної моделі. *Measuring and computing devices in technological processes*. 2023. № 3. С. 224–227. URL: https://doi.org/10.31891/2219-9365-2023-75-26 .

18. Соколов К. А. Візуальне виявлення та відстеження малогабаритних рухомих об’єктів на основі функціональних особливостей зорового апарату та особливостях сприйняття людини. *Вчені записки Таврійського Національного Університету імені В.І. Вернадського серія: технічні науки*. 2023. Т. 34(73), № 5. С. 75–82. URL: https://doi.org/10.32782/2663-5941/2023.5/13.

19. Статистичні імітаційні моделі оптимізації технічного обслуговування повітряних суден / O. C. Okoro та ін. *Системи управління, навігації та зв’язку. Збірник наукових праць*. 2022. Т. 3, № 69. С. 8–12. URL: https://doi.org/10.26906/sunz.2022.3.008 .

20. A Bayesian optimization hyperband-optimized incremental deep belief network for online battery behaviour modelling for a satellite simulator / M. Cao та ін. *Journal of energy storage*. 2023. Т. 58. С. 106348. URL: https://doi.org/10.1016/j.est.2022.106348.

21. A context-aware robust intrusion detection system: a reinforcement learning-based approach / K. Sethi та ін. *International journal of information security*. 2019. Т. 19, № 6. С. 657–678. URL: https://doi.org/10.1007/s10207-019-00482-7.

22. Adams N. Semi-supervised learning. *Journal of the royal statistical society: series A (statistics in society)*. 2009. Т. 172, № 2. С. 530. URL: https://doi.org/10.1111/j.1467-985x.2009.00588_3.x .

23. Agnihotri A., Batra N. Exploring bayesian optimization. *Distill*. 2020. Т. 5, № 5. URL: https://doi.org/10.23915/distill.00026 .

24. A multi-auv maritime target search method for moving and invisible objects based on multi-agent deep reinforcement learning / G. Wang та ін. *Sensors*. 2022. Т. 22, № 21. С. 8562. URL: https://doi.org/10.3390/s22218562 .

25. An approach for economic design of wide area monitoring system by co-optimizing phasor measurement unit placement and associated communication infrastructure / C. D. Patel та ін. *International transactions on electrical energy systems*. 2021. Т. 31, № 8. URL: https://doi.org/10.1002/2050-7038.12977 .

26. An improved DQN path planning algorithm / J. Li та ін. *The journal of supercomputing*. 2021. URL: https://doi.org/10.1007/s11227-021-03878-2 .

27. An improved r-index model for terrain visibility analysis for landslide monitoring with insar / T. Ren та ін. *Remote sensing*. 2021. Т. 13, № 10. С. 1938. URL: https://doi.org/10.3390/rs13101938 .

28. Artificial-Intelligence-Based distributed belief propagation and recurrent neural network algorithm for wide-area monitoring systems / S. Bhamidipati та ін. *IEEE network*. 2020. Т. 34, № 3. С. 64–72. URL: https://doi.org/10.1109/mnet.011.1900322 .

29. A spatially explicit reinforcement learning model for geographic knowledge graph summarization / B. Yan та ін. *Transactions in GIS*. 2019. Т. 23, № 3. С. 620–640. URL: https://doi.org/10.1111/tgis.12547 .

30. A survey of deep convolutional neural networks applied for prediction of plant leaf diseases / V. S. Dhaka та ін. *Sensors*. 2021. Т. 21, № 14. С. 4749. URL: https://doi.org/10.3390/s21144749 .

31. A systemic comparison between using augmented data and synthetic data as means of enhancing wafermap defect classification / R. Alqudah та ін. *Computers in industry*. 2023. Т. 145. С. 103809. URL: https://doi.org/10.1016/j.compind.2022.103809 .

32. Automatic alignment of contemporary vector data and georeferenced historical maps using reinforcement learning / W. Duan та ін. *International journal of geographical information science*. 2019. Т. 34, № 4. С. 824–849. URL: https://doi.org/10.1080/13658816.2019.1698742 .

33. Bagawan M., Hombal U. Internal examination evaluation system. *International journal of engineering applied sciences and technology*. 2020. Т. 5, № 1. С. 679–682. URL: https://doi.org/10.33564/ijeast.2020.v05i01.121 .

34. Barlow H. B. Unsupervised learning. *Neural computation*. 1989. Т. 1, № 3. С. 295–311. URL: https://doi.org/10.1162/neco.1989.1.3.295 .

35. Blitz-SLAM: A semantic SLAM in dynamic environments / Y. Fan та ін. *Pattern recognition*. 2022. Т. 121. С. 108225. URL: https://doi.org/10.1016/j.patcog.2021.108225 .

36. Boy S. OpenAI's pioneering contributions to reinforcement learning. *Medium*. URL: https://medium.com/@smartboy91221/openais-pioneering-contributions-to-reinforcement-learning-805f6a24e7f1.

37. Bräunl T. Lidar sensors. *Mobile robot programming*. Cham, 2023. С. 53–57. URL: https://doi.org/10.1007/978-3-031-32797-1_4 .

38. Brownlee J. Hyperparameter optimization with random search and grid search. *MachineLearningMastery.com*. URL: https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/.

39. Bulut V. Optimal path planning method based on epsilon-greedy Q-learning algorithm. *Journal of the brazilian society of mechanical sciences and engineering*. 2022. Т. 44, № 3. URL: https://doi.org/10.1007/s40430-022-03399-w .

40. Bzdok D., Krzywinski M., Altman N. Machine learning: supervised methods. *Nature methods*. 2018. Т. 15, № 1. С. 5–6. URL: https://doi.org/10.1038/nmeth.4551 .

41. Chacon S., Long J. Git. URL: https://git-scm.com.

42. Chaysri P., Blekas K., Vlachos K. Multiple mini-robots navigation using a collaborative multiagent reinforcement learning framework. *Advanced robotics*. 2020. Т. 34, № 13. С. 902–916. URL: https://doi.org/10.1080/01691864.2020.1757507 .

43. Chen R., Goldberg J. H. Actor-critic reinforcement learning in the songbird. *Current opinion in neurobiology*. 2020. Т. 65. С. 1–9. URL: https://doi.org/10.1016/j.conb.2020.08.005 .

44. Constrained deep q-learning gradually approaching ordinary q-learning / S. Ohnishi та ін. *Frontiers in neurorobotics*. 2019. Т. 13. URL: https://doi.org/10.3389/fnbot.2019.00103 .

45. Cyr A., Thériault F. Spatial concept learning: a spiking neural network implementation in virtual and physical robots. *Computational intelligence and neuroscience*. 2019. Т. 2019. С. 1–8. URL: https://doi.org/10.1155/2019/8361369 .

46. Deep deterministic policy gradient for navigation of mobile robots in simulated environments / J. C. Jesus та ін. *2019 19th international conference on advanced robotics (ICAR)*, м. Belo Horizonte, Brazil, 2–6 груд. 2019 р. 2019. URL: https://doi.org/10.1109/icar46387.2019.8981638 .

47. Design of action-value function in motion planning for autonomous blimp robots / K. Motoyama та ін. *IEEJ transactions on electronics, information and systems*. 2004. Т. 124, № 10. С. 1930–1937. URL: https://doi.org/10.1541/ieejeiss.124.1930 .

48. Dynamic handoff policy for RAN slicing by exploiting deep reinforcement learning / Y. Wu та ін. *EURASIP journal on wireless communications and networking*. 2021. Т. 2021, № 1. URL: https://doi.org/10.1186/s13638-021-01939-x .

49. Functions of learning rate in adaptive reward learning / X. Wu та ін. *Frontiers in human neuroscience*. 2017. Т. 11. URL: https://doi.org/10.3389/fnhum.2017.00592 .

50. Ghafarokhi O. I., Moattari M., Forouzantabar A. Composite load modeling by spatial-temporal deep attention network based on wide-area monitoring systems. *Journal of intelligent & fuzzy systems*. 2021. Т. 40, № 6. С. 12215–12226. URL: https://doi.org/10.3233/jifs-210296 .

51. Golub Y. I. Image quality assessment. *«System analysis and applied information science»*. 2022. № 4. С. 4–15. URL: https://doi.org/10.21122/2309-4923-2021-4-4-15 .

52. GPS monitoring reveals circadian rhythmicity in free-grazing sheep / J. Plaza та ін. *Applied animal behaviour science*. 2022. Т. 251. С. 105643. URL: https://doi.org/10.1016/j.applanim.2022.105643 .

53. Ground truth based comparison of saliency maps algorithms / K. Szczepankiewicz та ін. *Scientific reports*. 2023. Т. 13, № 1. URL: https://doi.org/10.1038/s41598-023-42946-w .

54. Ha D. Reinforcement learning for improving agent design. *Artificial life*. 2019. Т. 25, № 4. С. 352–365. URL: https://doi.org/10.1162/artl_a_00301 .

55. Han R., Chen K., Tan C. Curiosity-driven recommendation strategy for adaptive learning via deep reinforcement learning. *British journal of mathematical and statistical psychology*. 2020. Т. 73, № 3. С. 522–540. URL: https://doi.org/10.1111/bmsp.12199 .

56. Hedge options using reinforcement learning toolbox- MATLAB & simulink. *MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink*. URL: https://www.mathworks.com/help/finance/hedging-option-using-reinforcement-learning.html.

57. Home. *Docker Documentation*. URL: https://docs.docker.com .

58. Hrakholski D. Classification of point clouds and the creation of a digital terrain model in the new version of the credo 3D scan program. *Modern achievements of geodesic science and industry*. 2020. Т. I, № 39. С. 53–56. URL: https://doi.org/10.33841/1819-1339-1-39-10 .

59. Inverse reinforcement learning-based time-dependent A* planner for human-aware robot navigation with local vision / S. Sun та ін. *Advanced robotics*. 2020. Т. 34, № 13. С. 888–901. URL: https://doi.org/10.1080/01691864.2020.1753569 .

60. Ismail R. M., Muthukumaraswamy S., Sasikala A. Military support and rescue robot. *2020 4th international conference on intelligent computing and control systems (ICICCS)*, м. Madurai, India, 13–15 трав. 2020 р. 2020. URL: https://doi.org/10.1109/iciccs48265.2020.9121041 .

61. Kelly S. Model view controller. *Python, pygame, and raspberry pi game development*. Berkeley, CA, 2019. С. 213–231. URL: https://doi.org/10.1007/978-1-4842-4533-0_17 .

62. Kerner S. M. What is Q-learning?. *Enterprise AI*. URL: https://www.techtarget.com/searchenterpriseai/definition/Q-learning#:~:text=Q-learning%20is%20a%20machine,a%20type%20of%20reinforcement%20learning. .

63. Kuang W. Fundamentals of reinforcement learning : навч. посіб. Texas : University of Texas Rio Grande Valley, 2021. 111 с. URL: https://faculty.utrgv.edu/weidong.kuang/book/RL.pdf.

64. Küçük K., Anbaroğlu B. Spatial accuracy assessment of buildings in openstreetmap. *International journal of environment and geoinformatics*. 2019. Т. 6, № 3. С. 309–316. URL: https://doi.org/10.30897/ijegeo.666114 .

65. L. A. P., Fu M. C. Risk-Sensitive reinforcement learning via policy gradient search. *Foundations and trends® in machine learning*. 2022. Т. 15, № 5. С. 537–693. URL: https://doi.org/10.1561/2200000091 .

66. Lupu A., Precup D. Imitation upper confidence bound for bandits on a graph. *Proceedings of the AAAI conference on artificial intelligence*. 2018. Т. 32, № 1. URL: https://doi.org/10.1609/aaai.v32i1.12183 .

67. Machine learning roadmap 2020. *Whimsical*. URL: https://whimsical.com/machine-learning-roadmap-2020-CA7f3ykvXpnJ9Az32vYXva.

68. Maini P., Tokekar P., Sujit P. B. Visibility-Based persistent monitoring of piecewise linear features on a terrain using multiple aerial and ground robots. *IEEE transactions on automation science and engineering*. 2020. С. 1–13. URL: https://doi.org/10.1109/tase.2020.3014949 .

69. Mobile inspection robot / M. Ciszewski та ін. *Applied mechanics and materials*. 2013. Т. 319. С. 385–392. URL: https://doi.org/10.4028/www.scientific.net/amm.319.385 .

70. Monitoring terrain elevation of intertidal wetlands by utilising the spatial-temporal fusion of multi-source satellite data: A case study in the Yangtze (Changjiang) Estuary / W. Gao та ін. *Geomorphology*. 2021. Т. 383. С. 107683. URL: https://doi.org/10.1016/j.geomorph.2021.107683 .

71. Monitoring the dynamics of formby sand dunes using airborne lidar dtms / A. M. A. Mahmoud та ін. *Remote sensing*. 2021. Т. 13, № 22. С. 4665. URL: https://doi.org/10.3390/rs13224665 .

72. Nabavi S. M. H., Hajforoosh S. Exploration and exploitation tradeoff using fuzzy reinforcement learning. *International journal of computer applications*. 2012. Т. 59, № 5. С. 26–31. URL: https://doi.org/10.5120/9545-3994.

73. NoSQL / S. Edlich та ін. München : Carl Hanser Verlag GmbH & Co. KG, 2011. URL: https://doi.org/10.3139/9783446428553 .

74. Ofoeda J., Boateng R., Effah J. Application programming interface (API) research. *International journal of enterprise information systems*. 2019. Т. 15, № 3. С. 76–95. URL: https://doi.org/10.4018/ijeis.2019070105 .

75. Online transfer learning / P. Zhao та ін. *Artificial intelligence*. 2014. Т. 216. С. 76–102. URL: https://doi.org/10.1016/j.artint.2014.06.003 .

76. Open dataset – Waymo. *Waymo*. URL: https://waymo.com/open/.

77. Passalis N., Tefas A. Continuous drone control using deep reinforcement learning for frontal view person shooting. *Neural computing and applications*. 2019. Т. 32, № 9. С. 4227–4238. URL: https://doi.org/10.1007/s00521-019-04330-6 .

78. PostgreSQL: documentation. *PostgreSQL: The world's most advanced open source database*. URL: https://www.postgresql.org/docs/ .

79. Raschka S., Mirjalili V. Python machine learning: machine learning and deep learning with python, scikit-learn, and tensorflow 2, 3rd edition. Packt Publishing, 2019. 770 с.

80. Reinforcement learning and adaptive optimization of a class of Markov jump systems with completely unknown dynamic information / S. He та ін. *Neural computing and applications*. 2019. Т. 32, № 18. С. 14311–14320. URL: https://doi.org/10.1007/s00521-019-04180-2 .

81. Reinforcement learning / B. L. Moore та ін. Anesthesia & analgesia. 2011. Т. 112, № 2. С. 360–367. URL: https://doi.org/10.1213/ane.0b013e31820334a7

82. Reinforcement learning for logistics and supply chain management: methodologies, state of the art, and future opportunities / Y. Yan та ін. *Transportation research part E: logistics and transportation review*. 2022. Т. 162. С. 102712. URL: https://doi.org/10.1016/j.tre.2022.102712 .

83. Robillard M. P. Inversion of control. *Introduction to software design with java*. Cham, 2019. С. 191–236. URL: https://doi.org/10.1007/978-3-030-24094-3_8 .

84. Robot for under water monitoring. *International journal of engineering and advanced technology*. 2019. Т. 9, № 1S2. С. 26–32. URL: https://doi.org/10.35940/ijeat.a1019.1291s219 .

85. Robotic constant-force grinding control with a press-and-release model and model-based reinforcement learning / T. Zhang та ін. *The international journal of advanced manufacturing technology*. 2019. Т. 106, № 1-2. С. 589–602. URL: https://doi.org/10.1007/s00170-019-04614-0 .

86. Robotics for environmental monitoring. *IEEE robotics & automation magazine*. 2010. Т. 17, № 4. С. 118. URL: https://doi.org/10.1109/mra.2010.939741 .

87. Roshchin D. A. Improving the accuracy of forming a digital terrain model along a railway track. *Measurement techniques*. 2021. Т. 64, № 2. С. 100–108. URL: https://doi.org/10.1007/s11018-021-01903-7.

88. Sanghi N. Deep reinforcement learning with python: with pytorch, tensorflow and openai gym. Apress L. P., 2021.

89. Singh P., Manure A. Learn tensorflow 2.0. Berkeley, CA : Apress, 2020. URL: https://doi.org/10.1007/978-1-4842-5558-2 .

90. Sleaman W. K., Yavuz S. Indoor mobile robot navigation using deep convolutional neural network. *Journal of intelligent & fuzzy systems*. 2020. Т. 39, № 4. С. 5475–5486. URL: https://doi.org/10.3233/jifs-189030 .

91. Software modernization: theoretical framework / S. Bragagnolo та ін. *SSRN electronic journal*. 2022. URL: https://doi.org/10.2139/ssrn.4054806 .

92. Solar-powered parking analytics system using deep reinforcement learning / Y. Rezaei та ін. *ACM transactions on sensor networks*. 2023. URL: https://doi.org/10.1145/3584949 .

93. Spatial concept-based navigation with human speech instructions via probabilistic inference on Bayesian generative model / A. Taniguchi та ін. *Advanced robotics*. 2020. Т. 34, № 19. С. 1213–1228. URL: https://doi.org/10.1080/01691864.2020.1817777 .

94. Stiros S. C. GNSS (GPS) monitoring of dynamic deflections of bridges: structural constraints and metrological limitations. *Infrastructures*. 2021. Т. 6, № 2. С. 23. URL: https://doi.org/10.3390/infrastructures6020023 .

95. Sutton R. S., Barto A. G. Reinforcement learning: an introduction. 2-ге вид. The MIT Press, 2018. 552 с. URL: http://incompleteideas.net/book/RLbook2020.pdf.

96. Tennenhouse D. Research at VMware. *ACM SIGOPS operating systems review*. 2017. Т. 51, № 1. С. 1–4. URL: https://doi.org/10.1145/3139645.3139647 .

97. Text information source modeling for learning monitoring system / O. Pliushch та ін. *IEEE 3rd international conference on advanced trends in information theory (ATIT)*. 2021. Т. 3. С. 154–158.

98. Theodoridis S. Machine learning. 2-ге вид. Elsevier, 2020. URL: https://doi.org/10.1016/c2019-0-03772-7.

99. Thompson sampling for partially observable linear-quadratic control / T. Kargin та ін. *2023 american control conference (ACC)*, м. San Diego, CA, USA, 31 трав. – 2 черв. 2023 р. 2023. URL: https://doi.org/10.23919/acc55779.2023.10156461 .

100. Toal D. J. J., Bressloff N. W., Keane A. J. Kriging hyperparameter tuning strategies. *AIAA journal*. 2008. Т. 46, № 5. С. 1240–1252. URL: https://doi.org/10.2514/1.34822 .

101. Tupitsin N., Mykolaichuk V. System of UAV failure modeling for external pilot training in emergency situations. *2019 IEEE 5th international conference*

actual problems of unmanned aerial vehicles developments (APUAVD). 2019. № 5. С. 57–60.

102. Tymochko O., Afanasiev V., Mogylevska V. Method of the sensor systems' synthesis and drones for identifying thunderstorm activity's areas. *Наукові праці Державного науково-дослідного інституту випробувань і сертифікації озброєння та військової техніки*. 2022. Т. 14, № 4. С. 143–155. URL: https://doi.org/10.37701/dndivsovt.14.2022.15 .

103. Variable-dimensional Flower Pollination obstacle avoidance algorithm on autonomous walking of NAO robot in dynamic environment / S. Wen та ін. *Advanced robotics*. 2019. Т. 33, № 18. С. 911–924. URL: https://doi.org/10.1080/01691864.2019.1633954 .

104. Vision N., Nur S. IoT-WSN: survey on positioning techniques. *International journal of advanced smart sensor network systems*. 2022. Т. 12, № 3. С. 7–15. URL: https://doi.org/10.5121/ijassn.2022.12302 .

105. Watkins C. J. C. H., Dayan P. Q-learning. Boston : Kluwer Academic Publishers, 1992. 292 с. URL: https://doi.org/10.1007/bf00992698.

106. What is MongoDB Atlas?. *MongoDB: The Developer Data Platform | MongoDB*. URL: https://www.mongodb.com/docs/atlas/ .

107. Where does AlphaGo go: from church-turing thesis to AlphaGo thesis and beyond. *IEEE/CAA journal of automatica sinica*. 2016. Т. 3, № 2. С. 113–120. URL: https://doi.org/10.1109/jas.2016.7471613 .

108. Wireless sensor networks / S. Chai та ін. Singapore : Springer Singapore, 2020. URL: https://doi.org/10.1007/978-981-15-5757-6 .

109. Wireless signal propagation prediction based on computer vision sensing technology for forestry security monitoring / J. He та ін. *Sensors*. 2021. Т. 21, № 17. С. 5688. URL: https://doi.org/10.3390/s21175688 .

110. Yan W. Construction and application of seismic real-time data analysis model based on GPS monitoring. *Journal of sensors*. 2022. Т. 2022. С. 1–12. URL: https://doi.org/10.1155/2022/6010491 .

ДОДАТОК А

Загальний Код (Python)

```
import tensorflow as tf

from tf_agents.environments import py_environment,
tf_py_environment

from tf_agents.specs import array_spec

from tf_agents.environments.utils import validate_py_environment

from cv2 import imread

import numpy as np

from my_custom_env import MyCustomEnv

# Ініціалізація TensorFlow

tf.compat.v1.enable_v2_behavior()

# Завантаження та обробка карти

def load_and_process_map(map_path):

    map_image = imread(map_path, 0)  # Завантаження карти у
вигляді сірого зображення

    processed_map = some_processing_function(map_image)  # Функція
обробки карти

    return processed_map

# Клас середовища

class EnvironmentModel(py_environment.PyEnvironment):

    def __init__(self, map_data):

        self._map = map_data

        # Визначення простору дій та спостережень

        self._action_spec = ...

        self._observation_spec = ...

        # Ініціалізація стану середовища

        self._state = ...

    def action_spec(self):

        return self._action_spec

    def observation_spec(self):
```

```
        return self._observation_spec
    def _reset(self):
        # Ресет середовища
        return ...
    def _step(self, action):
        # Крок середовища
        return ...
# Основний код
def main():
    map_path = 'path/to/map.png'
    map_data = load_and_process_map(map_path)
    # Створення екземпляра середовища
    environment = EnvironmentModel(map_data)
    validate_py
_environment(environment, episodes=5)
    # Конвертація в TensorFlow середовище
    tf_env = tf_py_environment.TFPyEnvironment(environment)
if __name__ == "__main__":
    main()
```

Модуль Моделювання Середовища

Файл environment_model.py¶

```
import tensorflow as tf
from tf_agents.environments import py_environment,
tf_py_environment
from tf_agents.specs import array_spec
from tf_agents.environments.utils import validate_py_environment
from tf_agents.trajectories import time_step as ts
from cv2 import imread
import numpy as np
class EnvironmentModel(py_environment.PyEnvironment):
```

```
    def __init__(self, map_data):
        self._map = map_data
        # Оголошення дій: 0 - вліво, 1 - вправо, 2 - вгору, 3 -
вниз
        self._action_spec = array_spec.BoundedArraySpec(
            shape=(), dtype=np.int32, minimum=0, maximum=3,
name='action')
        # Спостереження можуть включати координати РЗ та ОМ
        self._observation_spec = array_spec.BoundedArraySpec(
            shape=(2,), dtype=np.float32, minimum=0,
name='observation')
        self._state = np.array([0.0, 0.0], dtype=np.float32)  #
Початкова позиція
        self._episode_ended = False
    def action_spec(self):
        return self._action_spec
    def observation_spec(self):
        return self._observation_spec
    def _reset(self):
        self._state = np.array([0.0, 0.0], dtype=np.float32)  #
Ресет до початкової позиції
        self._episode_ended = False
        return ts.restart(self._state)
    def _step(self, action):
        if self._episode_ended:
            return self.reset()
        # Проста логіка переміщення
        if action == 0:
            self._state[0] -= 1  # Рух вліво
        elif action == 1:
            self._state[0] += 1  # Рух вправо
        elif action == 2:
            self._state[1] += 1  # Рух вгору
        elif action == 3:
```

```
            self._state[1] -= 1  # Рух вниз
        self._episode_ended = ... # Логіка завершення епізоду
        if self._episode_ended:
            return ts.termination(self._state, reward=...)
        else:
            return ts.transition(self._state, reward=..., 
discount=...)
def load_and_process_map(map_path):
    map_image = imread(map_path, 0)
    return map_image
def create_environment(map_path):
    map_data = load_and_process_map(map_path)
    environment = EnvironmentModel(map_data)
    validate_py_environment(environment, episodes=5)
    return tf_py_environment.TFPyEnvironment(environment)
```

Модуль Роботизованих Засобів

Файл robotic_systems.py¶

```
import numpy as np
class RoboticSystem:
    def __init__(self, name, characteristics):
        self.name = name
        self.characteristics = characteristics  # Словник з 
характеристиками
        self.state = np.array([0.0, 0.0])  # Початкова позиція
    def update_state(self, new_state):
        self.state = new_state
    def decide_action(self, environment_state):
        # Проста логіка для визначення дії
        # Наприклад, рух до цілі
        goal = environment_state['goal_position']
```

```
        direction = goal - self.state
        action = np.argmax(np.abs(direction))  # Вибір осі для
руху
        return action
class SurveillanceDrone(RoboticSystem):
    def __init__(self, characteristics):
        super().__init__("SurveillanceDrone", characteristics)
    def decide_action(self, environment_state):
        # Специфічна логіка для безпілотника
        # Наприклад, патрулювання над заданою областю
        patrol_area = environment_state['patrol_area']
        # Вибір наступної точки патрулювання
        next_point = np.random.choice(patrol_area)
        return next_point
class GroundRobot(RoboticSystem):
    def __init__(self, characteristics):
        super().__init__("GroundRobot", characteristics)
    def decide_action(self, environment_state):
        # Специфічна логіка для наземного робота
        obstacle_position = environment_state['obstacle_position']
        # Розрахунок курсу для уникнення перешкоди
        avoid_direction = self.state - obstacle_position
        return avoid_direction
```

Модуль "Обробки Даних"

Файл data_processing.py¶

```
import pandas as pd
import numpy as np
class DataProcessor:
    def __init__(self):
        self.data = []
```

```
    def collect_data(self, simulation_data):
        """ Збір даних з симуляції """
        self.data.append(simulation_data)
    def preprocess_data(self):
        """ Попередня обробка зібраних даних """
        # Перетворення даних у формат, зручний для аналізу
(наприклад, DataFrame)
        return pd.DataFrame(self.data)
    def analyze_data(self, processed_data):
        """ Аналіз оброблених даних """
        # Виконання різних видів аналізу: статистичного, машинного
навчання тощо
        summary = processed_data.describe()
        return summary
def main():
    processor = DataProcessor()
    #simulation_data - це дані, отримані з моделювання
    simulation_data = {'param1': 1, 'param2': 2}
    processor.collect_data(simulation_data)
    processed_data = processor.preprocess_data()
    analysis_result = processor.analyze_data(processed_data)
    print(analysis_result)
if __name__ == "__main__":
    main()
```

Модуль "Штучного Інтелекту"¶

Файл ai_module.py¶

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
class AIModule:
    def __init__(self, input_shape, output_shape):
        self.model = self._build_model(input_shape, output_shape)
    def _build_model(self, input_shape, output_shape):
```

```
        """ Будування моделі нейронної мережі """
        model = Sequential([
            Dense(64, activation='relu', input_shape=input_shape),
            Dense(64, activation='relu'),
            Dense(output_shape, activation='linear')
        ])
        model.compile(optimizer='adam', loss='mse')
        return model
    def train(self, training_data, training_labels):
        """ Навчання моделі на наданих даних """
        self.model.fit(training_data, training_labels, epochs=10)
    def predict(self, input_data):
        """ Виконання прогнозування з використанням моделі """
        return self.model.predict(input_data)
def main():
    ai = AIModule(input_shape=(10,), output_shape=1)
    # Припустимо, ми маємо деякі дані для навчання
    training_data = ...  # Дані для навчання
    training_labels = ...  # Мітки для навчання
    ai.train(training_data, training_labels)
    # Використання моделі для прогнозування
    test_data = ...  # Тестові дані
    predictions = ai.predict(test_data)
    print(predictions)
if __name__ == "__main__":
    main()
```

Модуль "Комунікацій"

Файл communication_module.py¶

```
class CommunicationModule:
    def __init__(self):
```

```
        self.subscribers = []
    def subscribe(self, subscriber):
        """ Реєстрація компонентів для отримання оновлень """
        self.subscribers.append(subscriber)
    def send_message(self, message):
        """ Розсилка повідомлень зареєстрованим компонентам """
        for subscriber in self.subscribers:
            subscriber.receive_message(message)
class Component:
    def receive_message(self, message):
        """ Обробка отриманих повідомлень """
        print(f"Received message: {message}")
def main():
    comm_module = CommunicationModule()
    component1 = Component()
    component2 = Component()
    comm_module.subscribe(component1)
    comm_module.subscribe(component2)
    comm_module.send_message("Test message")
if __name__ == "__main__":
    main()
```

Централізований Модуль Управління Даними

Файл data_management_module.py¶

```
import pandas as pd
class DataManagementModule:
    def __init__(self):
        self.data_storage = {}
    def store_data(self, key, data):
        """ Зберігання даних за вказаним ключем """
        self.data_storage[key] = data
```

```
    def retrieve_data(self, key):
        """ Отримання даних за ключем """
        return self.data_storage.get(key, None)
    def integrate_external_data(self, external_data_source):
        """ Інтеграція даних з зовнішніх джерел """
        external_data =
self._fetch_data_from_source(external_data_source)
        for key, data in external_data.items():
            self.store_data(key, data)
    def _fetch_data_from_source(self, source):
        return {'external_data_key': pd.DataFrame({'column1': [1,
2, 3]})}
def main():
    data_module = DataManagementModule()
    data_module.store_data('internal_data', pd.DataFrame({'A': [1,
2, 3]}))
    # Симулюємо інтеграцію з зовнішнім джерелом
    data_module.integrate_external_data('http://external-data-
source.com')
    retrieved_data = data_module.retrieve_data('internal_data')
    print(retrieved_data)
if __name__ == "__main__":
    main()
```

API

Файл framework_api.py¶

```
class FrameworkAPI:
    def __init__(self, data_module, communication_module,
ai_module):
        self.data_module = data_module
        self.communication_module = communication_module
        self.ai_module = ai_module
```

```
    def get_data(self, key):
        """ Отримання даних з модуля управління даними """
        return self.data_module.retrieve_data(key)
    def send_command(self, command):
        """ Відправка команди через модуль комунікацій """
        self.communication_module.send_message(command)
    def run_ai_analysis(self, data):
        """ Запуск аналізу даних через модуль ШІ """
        return self.ai_module.predict(data)
data_module = DataManagementModule()
communication_module = CommunicationModule()
ai_module = AIModule(input_shape=(10,), output_shape=1)
api = FrameworkAPI(data_module, communication_module, ai_module)
# Використання API для інтеракції з фреймворком
data = api.get_data('some_key')
api.send_command('start_simulation')
analysis_result = api.run_ai_analysis(data)
```

Модуль Оцінки Ефективності

Файл efficiency_evaluation_module.py¶

```
class EfficiencyEvaluationModule:
    def __init__(self):
        self.metrics = {}
    def evaluate_interaction(self, interaction_data):
        """ Оцінка взаємодії на основі зібраних даних """
        efficiency_score =
self._calculate_efficiency(interaction_data)
        return efficiency_score
    def _calculate_efficiency(self, data):
        """ Розрахунок ефективності взаємодії """
        return sum(data) / len(data)  # Простий приклад
```

```
def main():
    evaluation_module = EfficiencyEvaluationModule()
    interaction_data = [0.9, 0.8, 0.95]
    efficiency_score =
evaluation_module.evaluate_interaction(interaction_data)
    print(f"Efficiency Score: {efficiency_score}")
if __name__ == "__main__":
    main()
```

ДОДАТОК Б

"ЗАТВЕРДЖУЮ"
Проректор з науково-педагогічної
роботи ДУІКТ
доктор технічних наук, професор
_____ Беркман Л.П.
«18» вересня 2023 р.

АКТ

про реалізацію результатів дисертаційної роботи
Миколайчук Віри Романівни на тему: «Методика оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання» у навчальному процесі Державного університету інформаційно-комунікаційних технологій

Комісія у складі: голови комісії - директора Навчально-наукового інституту інформаційних технологій, доктора технічних наук, професора Бондарчука А.П., та членів комісії - завідувача кафедри Комп'ютерної інженерії, кандидата технічних наук, доцента Лащевської Н.О.;- завідувача кафедри Інженерії програмного забезпечення автоматизованих систем, доктора технічних наук, професора Сторчак К.П..; - завідувача кафедри Інженерії програмного забезпечення доктора технічних наук, доцента Замрій І.В. розглянула дисертаційну роботу Миколайчук Віри Романівни на тему: «Методика оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання» та публікації автора за матеріалами дисертаційної роботи. Результати впроваджено в начальний процес Навчально-наукового інституту інформаційних технологій, а саме:

1. Розроблену модель середовища моніторингу з урахуванням роботизованих засобів та рухомих об'єктів.
2. Розроблену модель роботизованого засобу, що включає розробку алгоритму формування множини можливих дій, які система може виконувати для ідентифікації об'єкта та модель вибору поведінки.

3. Розроблену методика оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання.

На основі аналізу представлених матеріалів комісія установила, що отримані наукові результати використовуються в навчальному процесі Державною університету інформаційно-комунікаційних технологій при викладанні навчальних дисциплін: «Робототехніка» та «Інформаційні системи та пристрої в робототехніці та мехатроніці».

Голова комісії

Директор ННІІТ
доктор технічних наук, професор А.П. Бондарчук

Члени комісії

Завідувач кафедри Комп'ютерної інженерії, кандидат технічних наук, доцент Лащевська Н.О.

Завідувач кафедри Інженерії програмного забезпечення , доктор технічних наук, професор Сторчак К.П.

Завідувач кафедри Інженерії програмного забезпечення, доктор технічних наук, доцент Замрій І.В.

HUAWEI
ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ «ХУАВЕЙ УКРАЇНА»
LIMITED LIABILITY COMPANY "HUAWEI UKRAINE"
ТОВ «ХУАВЕЙ УКРАЇНА» / LLC "HUAWEI UKRAINE"

03124, Київ, вул. Радищева, буд. 10/14, (ідентифікаційний код: 33501267), тел. 498-77-18, факс 498-77-19
10/14, Radischeva street, Kiev, 03124, Ukraine, (Company Code: 33501267), tel. 498-77-18, fax 498-77-19

ЗАТВЕРДЖУЮ
Заступник директора з питань персоналу
ТОВ "Хуавей Україна"
_____ В.Б. Білавка
(підпис, ініціали, прізвище)

«26» *вересня* 2023 р.

АКТ

про реалізацію результатів дисертаційної роботи

Миколайчук Віри Романівни

на тему: «Методика оцінки ефективності роботизованої системи моніторингу місцевості на основі машинного навчання»

Матеріали роботи використано в науково-дослідних і дослідно-конструкторських роботах, що ведуться в ТОВ "Хуавей Україна", а саме розроблена модель середовища моніторингу з урахуванням роботизованих засобів та рухомих об'єктів дозволила вдосконалити алгоритми визначення об'єктів та моніторингу динамічного оточення, а також розроблена модель роботизованого засобу та алгоритм формування множини можливих дій, які система може виконувати для ідентифікації об'єкта та модель вибору поведінки, дозволили покращити автономні функції роботів та роботизованих систем у мобільних пристроях.

Матеріали дисертаційної роботи дозволять вдосконалити модулі та алгоритми прискорювача платформи для розробки мобільних додатків,

яка надає інструментарій для впровадження функцій машинного навчання Huawei ML Kit, що є важливим внеском у розвиток сучасних технологій та їх практичне впровадження в сферу мобільних додатків та робототехніки.

Заступник директора з питань
персоналу ТОВ "Хуавей Україна" В.Б. Білавка

www.huawei.com

Документ підписано у сервісі Вчасно (початок)
Дисертація_Миколайчук.pdf

Документ підписано у сервісі Вчасно (продовження)
Дисертація_Миколайчук.pdf

Документ відправлено: 09:00 11.12.2023

Власник документу

Електронний підпис
09:00 11.12.2023
Ідентифікаційний код: 3495412107
Миколайчук Віра Романівна
Власник ключа: Миколайчук Віра Романівна
Час перевірки КЕП/ЕЦП: 09:00 11.12.2023
Статус перевірки сертифікату: Сертифікат діє
Серійний номер: 3ED5083160DBC59B040000001517190 0A170E200
Тип підпису: кваліфікований