

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АЕРОКОСМІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Кваліфікаційна
наукова праця
на правах рукопису

Щеглов Владислав Романович

УДК 004.896:004.052:004.942(043)

ДИСЕРТАЦІЯ

**МОДЕЛІ, МЕТОД ТА ЗАСОБИ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ
ДІАГНОСТУВАННЯ ПРОМИСЛОВОГО ОБЛАДНАННЯ З
ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ І
ЦИФРОВИХ ДВІЙНИКІВ**

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. Щеглов

Науковий керівник Морозова Ольга Ігорівна, д.т.н., професор

Харків – 2026

АНОТАЦІЯ

Щеглов Владислав Романович. Моделі, метод та засоби інтелектуальної системи діагностування промислового обладнання з використанням індустріального інтернету речей і цифрових двійників – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 123 Комп'ютерна інженерія. – Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, 2026.

Дисертаційна робота присвячена розробленню моделей, методу та програмних засобів оцінювання і забезпечення готовності промислового обладнання систем індустріального інтернету речей в умовах виникнення прихованих відмов шляхом використання інтелектуальних систем діагностування на основі цифрових двійників з режимами донавчання.

Об'єктом дослідження є процеси технічного діагностування та оцінювання надійності промислового обладнання з використанням індустріального інтернету речей.

Вперше запропоновано теоретико-множинну модель та архітектуру інтелектуальної системи діагностування промислового обладнання систем індустріального інтернету речей з використанням цифрового двійника, яка, на відміну від існуючих, забезпечує локальне та федеративне донавчання, коригування тестових послідовностей і параметрів оперативного контролю на базі виявлених прихованих відмов, а також можливість динамічного обчислення функції готовності, що надає змогу знизити ймовірність прихованих відмов та підвищити керованість процесами обслуговування.

Удосконалено базові та мультифрагментні моделі готовності відновлювального промислового обладнання систем індустріального інтернету речей, які, на відміну від існуючих, утворюють множину, сформовану на основі розробленого онтологічного класифікатора, та враховують зміну достовірності діагностування, зумовлену донавчанням цифрового двійника після прояву різних

типів відмов, що дозволяє обирати структуру моделі залежно від особливостей об'єкта і процесів діагностування та підвищити достовірність оцінювання готовності системи.

Удосконалено метод верифікації та валідації мультифрагментних марковських моделей інтелектуальних систем діагностування промислового обладнання з використанням індустриального інтернету речей, який відрізняється використанням багатоступеневої перехресної перевірки результатів за допомогою двох незалежних методологій (на основі графів станів та структурно-автоматних моделей), а також врахуванням математичної «жорсткості» (stiffness) матриці інтенсивностей для вибору чисельного методу розв'язання, що дозволяє уникнути обчислювальних похибок та підвищити достовірність оцінювання готовності.

Ключові слова: інтелектуальна система діагностування, індустриальний інтернет речей, цифровий двійник, готовність, марковські моделі, приховані відмови, машинне навчання, донавчання, онтологічний класифікатор, підтримка прийняття рішень, технічне обслуговування за технічним станом, моделювання, федеративне навчання, верифікація та валідація.

Список публікацій здобувача за темою дисертації

1. Щеглов В. Р., Морозова О. І. Методи та технології розроблення цифрових двійників для гарантоздатних систем індустриального інтернету речей. *Системи управління, навігації та зв'язку*. 2022. Вип. 4 (70). С. 127–137. DOI: 10.26906/SUNZ.2022.4.127.
2. Kharchenko V., Shcheglov V., Ivasiuk O., Morozova O. Digital Twin-Based Lifecycle Methodology for Ensuring Safety of NPP/SMR I&C Systems. *Technologies*. 2026. Vol. 14, no. 1, article no. 46. P. 1–33. DOI: 10.3390/technologies14010046.
3. Щеглов В. Р. Інформаційна технологія оцінювання готовності інтелектуальних систем діагностування з донавчанням на основі цифрових

двійників. *Measuring and computing devices in technological processes*. 2026. № 1. С. 87–96. DOI: 10.31891/2219-9365-2026-85-11.

4. Shcheglov V., Kharchenko V., Borges J., Morozova O. Availability Markov Models of Complex Equipment with Relearning Diagnostic Means after Detecting Latent Failures. *Dependable Systems, Services and Technologies (DESSERT'2023)* : Proc. 13th IEEE Int. Conf., Athens, Greece, Oct. 13–15, 2023. P. 1–7. DOI: 10.1109/DESSERT61349.2023.10416539.

5. Shcheglov V., Kharchenko V., Volochiy B., Morozova O., Zaitseva E. Availability Assessing Intelligent Diagnostic System with Relearning after Complex Equipment Failures: Structural-Automata Model-based Methodology of Markov's Analysis. *Dependable Systems, Services and Technologies (DESSERT'2024)* : Proc. 14th IEEE Int. Conf., Athens, Greece, 2024. P. 1–8. DOI: 10.1109/DESSERT65323.2024.11122169.

6. Щеглов В. Р., Морозова О. І. Застосування технології цифрових двійників відносно безпілотних літальних апаратів. *Авіація, промисловість, суспільство* : матеріали наук.-практ. конф., м. Харків, 2024. Харків, 2024. С. 47–51.

7. Гаєвський В. В., Кунуп Т. В., Морозова О. І., Щеглов В. Р. Розділ 11. Інтелектуальні системи інтернету речей з використанням технологій цифрових двійників. *Methods and technologies of ensuring quality and safety of intelligent systems* : кол. монографія / за заг. ред. В. С. Харченка, О. І. Морозової. Харків : Міністерство освіти і науки України, Національний аерокосмічний університет «ХАІ», 2023. С. 193–224.

ABSTRACT

Shcheglov Vladyslav Romanovych. Models, a method, and means of an intelligent diagnostic system for industrial equipment using the Industrial Internet of Things and digital twins – Manuscript copyright.

Dissertation on competition for scientific degree of Doctor of Philosophy by specialty 123 Computer Engineering. – National Aerospace University «Kharkiv Aviation Institute», Kharkiv, 2026.

The dissertation is devoted to the development of models, a method, and software tools for assessing and ensuring the availability of industrial equipment in Industrial Internet of Things systems under conditions of latent failure occurrence, through the use of intelligent diagnostic systems based on digital twins with relearning modes.

The object of research is the processes of technical diagnostics and availability assessment of industrial equipment using the Industrial Internet of Things.

For the first time, a set-theoretic model and an architecture of an intelligent diagnostic system for industrial equipment in Industrial Internet of Things systems using a digital twin are proposed, which, unlike existing ones, ensure local and federated relearning, adjustment of test sequences and operational control parameters based on detected latent failures, as well as the capability of dynamic calculation of the availability function, enabling a reduction in the probability of latent failures and an increase in the controllability of maintenance processes.

The basic and multi-fragmental availability models for repairable industrial equipment in Industrial Internet of Things systems are improved, which, unlike existing ones, form a set generated based on the developed ontological classifier, and take into account the change in diagnostics trustworthiness caused by the relearning of the digital twin after the manifestation of various types of failures, allowing the selection of the model structure depending on the characteristics of the object and diagnostic processes, and improving the reliability of the system availability assessment.

The method of verification and validation of multi-fragmental Markov models of intelligent diagnostic systems for industrial equipment using the Industrial Internet of

Things is improved, which is distinguished by the use of multi-stage cross-verification of results using two independent methodologies (based on state transition graphs and structural-automata models), as well as the consideration of the mathematical «stiffness» of the transition rate matrix to select the numerical solution method, which allows avoiding computational errors and increases the reliability of the availability assessment.

Keywords: intelligent diagnostic system, Industrial Internet of Things, digital twin, availability, Markov models, latent failures, machine learning, relearning, ontological classifier, decision support, condition-based maintenance, modeling, federated learning, verification and validation.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	11
ВСТУП.....	12
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І ТЕХНОЛОГІЙ ОЦІНЮВАННЯ ТА ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ ПРОМИСЛОВОГО ОБЛАДНАННЯ З ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ ТА ІНТЕЛЕКТУАЛЬНОЮ СИСТЕМОЮ ДІАГНОСТУВАННЯ НА ОСНОВІ ЦИФРОВОГО ДВІЙНИКА. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	18
1.1 Аналіз особливостей функціонування та вимог до надійності промислового обладнання систем індустриального інтернету речей	18
1.1.1 Промислове обладнання з використанням індустриального інтернету речей. Визначення та класифікація	18
1.1.2 Аналіз вимог до надійності промислового обладнання з використанням індустріального інтернету речей	24
1.2 Аналіз методів і технологій забезпечення надійності промислового обладнання систем індустриального інтернету речей з використанням технічного діагностування та цифрових двійників.....	28
1.2.1 Аналіз існуючих методів і технологій технічного діагностування промислового обладнання з використанням індустриального інтернету речей. Обмеження та проблема прихованих відмов	29
1.2.2 Інтелектуальні системи діагностування промислового обладнання з використанням індустриального інтернету речей. Концепція донавчання в діагностиці	34
1.2.3 Аналіз технології Цифрових Двійників для забезпечення надійності промислового обладнання з використанням індустриального інтернету речей. Визначення, класифікація, зв'язок з життєвим циклом обладнання.....	38
1.2.4 Методи і обмеження використання Цифрових Двійників для забезпечення надійності промислового обладнання з використанням індустріального інтернету речей в інтелектуальних системах діагностування	44
1.3 Аналіз методів і моделей оцінювання надійності промислового обладнання систем індустриального інтернету речей з використанням технічного діагностування та цифрових двійників.....	47
1.3.1 Аналіз існуючих методів та математичних моделей оцінювання надійності промислового обладнання систем індустриального інтернету речей	47

1.3.2 Аналіз існуючих програмних засобів моделювання та методів верифікації результатів моделювання	48
1.3.3 Обґрунтування та вибір показників оцінювання надійності промислового обладнання з використанням індустріального інтернету речей	50
1.3.4 Формулювання науково-прикладного протиріччя	51
1.4 Обґрунтування задач та методики дослідження	52
1.4.1 Постановка загальної та часткових завдань дослідження	52
1.4.2 Етапи та методика дослідження	53
1.5 Висновки до першого розділу.....	55
Література до першого розділу.....	56
РОЗДІЛ 2. РОЗРОБЛЕННЯ АРХІТЕКТУРИ ТА МАТЕМАТИЧНИХ МОДЕЛЕЙ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ДІАГНОСТУВАННЯ ПРОМИСЛОВОГО ОБЛАДНАННЯ З ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ І ЦИФРОВИХ ДВІЙНИКІВ ІЗ ДОНАВЧАННЯМ	64
2.1 Розроблення теоретико-множинної моделі та архітектури ІСД промислового обладнання з використанням індустріального інтернету речей із донавчанням на основі цифрового двійника	64
2.1.1 Теоретико-множинна модель ІСД промислового обладнання з використанням індустріального інтернету речей із донавчанням на основі цифрового двійника	65
2.1.2 Архітектура ІСД промислового обладнання з використанням індустріального інтернету речей із донавчанням на основі цифрового двійника	70
2.1.3 Розроблення методу технічного діагностування промислового обладнання з використанням індустріального інтернету речей з оперативним та федеративним донавчанням на основі цифрового двійника	74
2.2 Розроблення комплексу базових моделей надійності промислового обладнання з використанням індустріального інтернету речей та ІСД із донавчанням.....	78
2.2.1 Обґрунтування ознак класифікації.....	79
2.2.2 Розроблення онтологічного класифікатора базових моделей.....	82
2.2.3 Розроблення комплексу базових моделей надійності.....	88
2.2.4 Багатокритеріальне ранжування та вибір цільової базової моделі.....	98
2.3. Обґрунтування припущень	101

2.4 Висновки до другого розділу	102
Література до другого розділу	104
РОЗДІЛ 3. РОЗРОБЛЕННЯ ТА ВЕРИФІКАЦІЯ МУЛЬТИФРАГМЕНТНИХ МАРКОВСЬКИХ МОДЕЛЕЙ ІСД ПРОМИСЛОВОГО ОБЛАДНАННЯ З ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ І ЦИФРОВИХ ДВІЙНИКІВ ІЗ ДОНАВЧАННЯМ	107
3.1 Послідовність розроблення та дослідження МФММ	107
3.2 Розроблення та дослідження МФММ1	114
3.2.1 Припущення	115
3.2.2 Розроблення	115
3.2.3 Дослідження	119
3.2.4 Попередні результати	126
3.3 Розроблення та дослідження МФММ2	127
3.3.1 Припущення	127
3.3.2 Розроблення вербальної моделі та формування базових подій	128
3.3.3 Розроблення ймовірнісного графу	135
3.3.4 Розроблення структурно-автоматної моделі	139
3.3.5 Дослідження	142
3.3.6 Попередні результати	145
3.4 Метод валідації та верифікації МФММ ІСД	147
3.5 Висновки до третього розділу	155
Література до третього розділу	157
РОЗДІЛ 4 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ТА ПРОГРАМНИЙ ЗАСІБ ДЛЯ СИСТЕМ ДІАГНОСТУВАННЯ ПРОМИСЛОВОГО ОБЛАДНАННЯ З ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ ІЗ ДОНАВЧАННЯМ НА ОСНОВІ ЦИФРОВИХ ДВІЙНИКІВ. ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ	160
4.1 Послідовність використання та особливості практичної реалізації інформаційної технології і розроблених моделей	160
4.2 Інформаційна технологія та архітектура розгортання ІСД на основі цифрового двійника для промислового обладнання із використанням індустріального інтернету речей	162
4.2.1 Концепція еталонного розгортання та зв'язок з теоретичною моделлю	162

	10
4.2.2 Структурна організація еталонної ІТ-архітектури	164
4.2.3 Модель даних та інтерфейси взаємодії.....	168
4.3 Програмний засіб «R-SIM» для оцінювання надійності промислового обладнання систем індустріального інтернету речей з використанням ІСД та донавчанням.....	170
4.3.1 Проектування архітектури та опис функціоналу.....	171
4.3.2 Приклади роботи та особливості застосування програмного засобу	175
4.4 Апробація МФММ4 та програмного засобу для обґрунтування надійності обладнання малих модульних реакторів із ІСД з донавчанням	180
4.5 Аналіз результатів впровадження	187
4.6 Висновки до четвертого розділу.....	188
Література до четвертого розділу.....	189
ВИСНОВКИ.....	191
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	194
ДОДАТОК А. АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ	204
ДОДАТОК Б. ЛІСТИНГИ КОДІВ ПРОГРАМНИХ ЗАСОБІВ	210

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БМ – Базова модель

ІСД – Інтелектуальна система діагностування

МФММ – Мультифрагментна марковська модель

НДР – Науково-дослідна робота

ОФМ – Однофрагментна модель

ПЗ – Програмне забезпечення

САМ – Структурно-автоматна модель

СТОіР – Служба технічного обслуговування і ремонту

ТО – Технічне обслуговування

ЦД – Цифровий двійник

ІІІ – Штучний інтелект

AI – Artificial Intelligence

BM – Base model

DT – Digital Twin

IDEF0 – Integration Definition for Function Modeling

IIoT – Industrial Internet of Things

IoT – Internet of Things

ML – Machine Learning

R-SIM – Reliability simulator

SMR – Small Modular Reactor

V&V – Verification and Validation

ВСТУП

Обґрунтування вибору теми дослідження. В останні роки впровадження технологій індустріального інтернету речей (IIoT) стрімко зростає на підприємствах різних галузей промисловості. Складне промислове обладнання та кіберфізичні системи вимагають безперервного моніторингу для забезпечення високих показників експлуатаційної готовності та надійності. Однак в умовах динамічної зміни режимів роботи та деградації матеріалів використання традиційних статичних засобів технічного діагностування є недостатньо ефективним, що призводить до накопичення прихованих відмов.

Зростаюча структурна складність обладнання вимагає пошуку нових адаптивних методів і засобів діагностування. Особливу увагу слід приділити розгортанню інтелектуальних систем діагностування на основі цифрових двійників, оскільки вони здатні забезпечити оперативне донавчання моделей контролю на базі виявлених неспецифікованих відмов. Таким чином, розроблення моделей, методу та засобів оцінювання і забезпечення надійності промислового обладнання систем IIoT із застосуванням цифрових двійників є актуальним науково-прикладним завданням, яке потребує подальшого вивчення.

Об'єкт дослідження – процеси технічного діагностування та оцінювання надійності промислового обладнання з використанням індустріального інтернету речей.

Предмет дослідження – моделі, метод та засоби забезпечення і оцінювання надійності промислового обладнання систем індустріального інтернету речей за допомогою інтелектуальних систем діагностування з використанням цифрових двійників та режимів донавчання.

Мета і завдання дослідження. Метою дисертаційного дослідження є підвищення надійності і достовірності її оцінювання у промисловому обладнанні систем індустріального інтернету речей шляхом розроблення і впровадження моделей, методу та засобів інтелектуального діагностування на основі цифрових двійників з режимами донавчання в умовах прихованих відмов.

Для досягнення мети дослідження необхідно вирішити наступні завдання:

- проаналізувати методи і технології забезпечення та оцінювання готовності промислового обладнання систем індустріального інтернету речей, виявити обмеження існуючих систем діагностування щодо врахування прихованих відмов та обґрунтувати мету, задачі і методику досліджень;
- розробити теоретико-множинну модель та архітектуру інтелектуальної системи діагностування промислового обладнання з використанням індустріального інтернету речей на основі цифрового двійника з функціями оперативного донавчання та коригування тестових послідовностей;
- розробити онтологічний класифікатор та комплекс базових і мультифрагментних марковських моделей готовності промислового обладнання, що враховують динаміку зміни достовірності діагностування при локальному та федеративному донавчанні цифрового двійника, з подальшим багатокритеріальним ранжуванням базових моделей;
- удосконалити метод верифікації та валідації мультифрагментних марковських моделей інтелектуальних систем діагностування шляхом використання багатоступеневої перехресної перевірки та врахуванні точності інструментів і математичної «жорсткості» матриці інтенсивностей переходів;
- розробити еталонну ІТ-архітектуру розгортання цифрового двійника інтелектуальної системи діагностування в середовищі індустріального інтернету речей, а також програмний засіб підтримки прийняття рішень для оцінювання готовності обладнання, та впровадити отримані результати в індустріальні проекти і навчальний процес.

Методи дослідження. У дисертаційній роботі використовувались методи системного аналізу, теорії ймовірностей та марковських випадкових процесів, теоретико-множинні моделі, теорії графів та структурно-автоматних моделей, методологія функціонального моделювання (IDEF0), методи онтологічного моделювання та теорії прийняття рішень.

Наукова новизна отриманих результатів:

- **вперше запропоновано** *теоретико-множинну модель та архітектуру* інтелектуальної системи діагностування промислового обладнання систем індустриального інтернету речей з використанням цифрового двійника, яка, **на відміну від існуючих**, забезпечує локальне та федеративне донавчання, коригування тестових послідовностей і параметрів оперативного контролю на базі виявлених прихованих відмов, а також можливість динамічного обчислення функції готовності, що надає змогу знизити ймовірність прихованих відмов та підвищити керованість процесами обслуговування;

- **удосконалено базові та мультифрагментні моделі** *готовності* відновлювального промислового обладнання систем індустриального інтернету речей, які, **на відміну від існуючих**, утворюють множину, сформовану на основі розробленого онтологічного класифікатора, та враховують зміну достовірності діагностування, зумовлену донавчанням цифрового двійника після прояву різних типів відмов, що дозволяє обирати структуру моделі залежно від особливостей об'єкта і процесів діагностування та підвищити достовірність оцінювання готовності системи;

- **удосконалено метод** верифікації та валідації мультифрагментних марковських моделей інтелектуальних систем діагностування промислового обладнання з використанням індустриального інтернету речей, **який відрізняється** використанням багатоступеневої перехресної перевірки результатів за допомогою двох незалежних методологій (на основі графів станів та структурно-автоматних моделей), а також врахуванням математичної «жорсткості» матриці інтенсивностей для вибору чисельного методу розв'язання, що дозволяє уникнути обчислювальних похибок та підвищити достовірність оцінювання готовності.

Особистий внесок здобувача полягає у розробленні теоретико-множинної моделі, архітектури, математичних моделей, методу верифікації та інструментальних засобів, які забезпечують вирішення поставлених задач

дисертаційного дослідження. Всі основні наукові та практичні результати отримані автором особисто та опубліковані у відповідних працях.

У працях, які опубліковані у співавторстві, автору належать:

- теоретико-множинна модель, структурна архітектура та спосіб технічного діагностування обладнання із використанням цифрового двійника з функцією оперативного донавчання, а також аналіз обмежень існуючих технологій ЦД в умовах індустріального інтернету речей;

- класифікатор моделей готовності інтелектуальних систем діагностування та синтез комплексу базових і мультифрагментних марковських моделей (лінійних та матричних), що враховують динаміку зміни достовірності діагностування при виникненні прихованих відмов та використанні режимів локального і федеративного донавчання;

- удосконалення методу верифікації та валідації (V&V) результатів марковського моделювання шляхом застосування перехресної перевірки на основі апарату структурно-автоматних моделей, а також розроблення алгоритму адаптивного вибору чисельного методу з урахуванням математичної «жорсткості» матриць інтенсивностей переходів;

- формалізація інформаційної технології оцінювання готовності за допомогою методології IDEF0, розроблення архітектури та написання програмного коду інструментального засобу підтримки прийняття рішень R-SIM, проведення симуляційного моделювання, візуалізація даних та аналіз результатів апробації на прикладі компонентів SMR.

Апробація матеріалів дисертації. Основні положення та ідеї дисертаційної роботи доповідалися та обговорювалися на конференціях: «Критичні комп'ютерні технології та системи» (м. Харків, 2022), «Перспективні мережеві та комп'ютерні технології» (м. Харків, 2023), «International Conference on Dependable Systems, Services and Technologies» (Athens, Greece, 2023, 2024), «Scientific International Conference on CBRNe» (Rome, Italy, 2023), «Авіація, Промисловість, Суспільство» (м. Харків, 2023), «International Workshop on Reliability Engineering and Computational Intelligence» (Zilina, Slovakia, 2024).

Зв'язок з науковими програмами, темами. Дисертаційна робота виконана у Національному аерокосмічному університеті «Харківський авіаційний інститут» відповідно з державними програмами та планами НДР:

- «Методи, засоби та технології моделювання, розроблення, розгортання та забезпечення гарантоздатності мобільних інтелектуальних систем для об'єктів критичної інфраструктури» (№ Д/Р 0124U003250, 2024-теперішній час);
- «Наукові засади і методи забезпечення гарантоздатності флотів БПЛА інтелектуальних систем моніторингу потенційно небезпечних і військових об'єктів» (№ Д/Р 0121U112172, 2021–2023 рр.);
- «Методи, моделі та інформаційні технології підвищення надійності та безпечності складних ІТ-систем на етапах розроблення та впровадження» (№ Д/Р 0121U113842, 2021–2023 рр.);
- «Методи, програмно-апаратні засоби та технології забезпечення гарантоздатності інтелектуальних систем індустріального інтернету речей» (№ Д/Р 0122U001065, 2022–2023 рр.).

Роль автора у зазначених НДР, в яких автор був безпосереднім виконавцем, полягає у розробленні моделей, методу та засобів оцінювання і забезпечення готовності промислового обладнання в умовах виникнення прихованих відмов шляхом використання інтелектуальних систем діагностування на основі цифрових двійників із донавчанням.

Практичне значення отриманих результатів. Практичні результати полягають у доведенні теоретичних положень дисертаційної роботи до конкретних методів, моделей, інформаційної технології та інструментальних програмних засобів (зокрема, ПЗ R-SIM) для підвищення надійності та точності його оцінювання у промисловому обладнанні систем індустріального інтернету речей та ІСД на основі ЦД. Результати дисертаційної роботи впроваджено (додаток А):

- у навчальному процесі Національного аерокосмічного університету «Харківський авіаційний інститут» (акт впровадження від 10 березня 2026);

- при виконанні науково-дослідних проєктів, що виконувались у Національному аерокосмічному університеті «Харківський авіаційний інститут» (акт впровадження від 09 березня 2026);

- на підприємстві ТОВ «НВП «РАДІКС» шляхом застосування методики використання цифрових двійників для підвищення ефективності процесів верифікації та валідації (V&V) інформаційно-керуючих систем і обґрунтування готовності обладнання у малих модульних реакторів з використанням засобів діагностування з донавчанням (акт впровадження від 26 лютого 2026).

Структура та обсяг дисертації. Дисертація складається із вступу, чотирьох розділів, висновку, списку виконаних джерел і додатків. Загальний обсяг дисертації складає 225 сторінок, з яких анотація на 5 сторінках, зміст на 4 сторінках, перелік умовних позначень на 1 сторінці, основний текст на 182 сторінках, список використаних джерел із 76 найменувань на 10 сторінках, додатки на 22 сторінках. Робота містить 17 таблиць та 41 рисунок.

Публікації. За темою дисертаційної роботи було опубліковано 7 наукових праць, серед яких: 2 статті у наукових фахових виданнях України категорії Б; 1 стаття у англomовних журналах категорії А, що індексовані у базі даних Scopus (квартіль Q1); 1 колективна монографія; 3 публікації у матеріалах наукових конференцій (з яких 2 індексуються у Scopus).

РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І ТЕХНОЛОГІЙ ОЦІНЮВАННЯ ТА ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ ПРОМИСЛОВОГО ОБЛАДНАННЯ З ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ ТА ІНТЕЛЕКТУАЛЬНОЮ СИСТЕМОЮ ДІАГНОСТУВАННЯ НА ОСНОВІ ЦИФРОВОГО ДВІЙНИКА. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз особливостей функціонування та вимог до надійності промислового обладнання систем індустриального інтернету речей

Протягом багатьох років світове виробництво характеризувалося безперервними інноваційними проривами, що призвели до радикальних змін у промисловості та пов'язаних з нею інформаційних процесах [1]. Від традиційної автоматизації замкнених систем (Індустрія 3.0) [2], розвинуті економіки переходять до Індустрії 4.0 та 5.0, формуються візії для Індустрії 6.0 [3].

Зі зростанням глобалізації та зміною споживчих очікувань, традиційні заводи, орієнтовані на масове виробництво ідентичних товарів, зіткнулися з вимогою масової кастомізації (партій розміром в одну одиницю), мінімізації часу виходу на ринок (time-to-market) та екстремального підвищення енергоефективності. Класичне виробництво не могло задовольнити ці вимоги через свою «локальність» – процеси були оптимізовані лише в межах одного цеху або однієї виробничої лінії, без врахування глобального контексту ланцюга постачання [4].

Тому в рамках Індустрії 4.0 та 5.0 виробництво переходить від автоматизації в замкнутих системах, відповідно до піраміди автоматизації або моделі ISA-95, до більш складної взаємодії – інтеграції та сумісності [3].

1.1.1 Промислове обладнання з використанням індустриального інтернету речей. Визначення та класифікація

Індустріальний інтернет речей (Industrial Internet of Things, IIoT) є одним із ключових проявлень Індустрії 4.0 у виробництві, що забезпечує глибоку

конвергенцію інформаційних технологій (IT) та операційних технологій (OT). ПоТ – це підрозділ Інтернету речей (Internet of Things, IoT), який зосереджується на покращенні доступності, продуктивності, масштабованості, економії часу та коштів виробничих підприємств [4]. Формальне визначення ПоТ – Система, що складається з мережевих інтелектуальних об'єктів, кіберфізичних активів, пов'язаних загальних інформаційних технологій та опціональних хмарних або периферійних обчислювальних платформ, які забезпечують інтелектуальний та автономний доступ, збір, аналіз, комунікацію та обмін інформацією про процеси, продукти та/або послуги в режимі реального часу в промисловому середовищі з метою оптимізації загальної вартості виробництва [5].

Відмінності між ПоТ та стандартним споживчим IoT є фундаментальними та зумовлені безпрецедентно високою ціною помилки у промисловому середовищі. Отже, ПоТ висуває жорсткі вимоги до детермінованості зв'язку, мінімальної затримки (субмілісекундного рівня) та надвисокої надійності у складних промислових умовах. Так, наприклад, у споживчому сегменті затримка пакетів даних або їх тимчасова втрата часто призводить лише до погіршення користувацького досвіду. В індустріальних системах керування рухом або роботизованих лініях відхилення у часі доставки пакету навіть на мілісекунду призводить до розсинхронізації конвеєра, руйнування дороговартісного обладнання, екологічних катастроф або загрози життю персоналу. Для забезпечення абсолютного детермінізму ПоТ-архітектури вимагають імплементації спеціалізованих мережевих протоколів, таких як Time-Sensitive Networking (TSN), що гарантують строге впорядкування пакетів та часову синхронізацію, а також механізмів Time Scheduled Channel Hopping (TSCH) у стандарті IEEE 802.15.4 для забезпечення надійності в середовищах із високим рівнем перешкод [6].

Систематизація рішень ПоТ є необхідною для структурування підходів до проектування, стандартизації та впровадження. Класифікація проводиться за декількома вимірами: галузевим спрямуванням, функціональними завданнями,

бізнес-моделями, еталонними архітектурами, семантичними онтологіями, а також технологічним стеком.

Зокрема, з огляду на галузеву специфіку, технології ПоТ сьогодні інтегруються у більшість критично важливих секторів економіки (від розумного виробництва та енергетики до логістики й видобувної промисловості) [7, 8], де кожна сфера висуває власні вимоги до безвідмовності сенсорних мереж та протоколів обміну даними. За функціональним призначенням такі інфраструктури системно охоплюють як завдання базового віддаленого моніторингу, так і складні процеси прогнозованого технічного обслуговування (Predictive Maintenance), управління енергоспоживанням та гарантування безпеки експлуатації [7, 9]. Як наслідок, відбувається глобальна трансформація бізнес-моделей підприємств [10]: від традиційного продажу фізичних активів індустрія переходить до сервіс-орієнтованих (Service-centered) та керованих даними (Data-driven) платформ. У таких економічних умовах головною цінністю стає не стільки саме обладнання, скільки гарантована безперервність надання послуг, що робить завдання інтелектуального технічного діагностування фундаментальною вимогою для будь-якої сучасної ПоТ-системи.

Для забезпечення міжгалузевої сумісності та стандартизації складних ПоТ-рішень розроблено низку фундаментальних еталонних архітектур, серед яких найзначущими є RAMI 4.0 (Reference Architecture Model Industrie 4.0) та IIRA (Industrial IoT Reference Architecture) [11]. Замість фокусування на окремих апаратних вузлах, ці багатовимірні моделі системно формалізують увесь життєвий цикл об'єкта, його функціональні домени та шари інтеграції. Водночас, згідно з базовою архітектурою за стандартом ДСТУ ISO/IEC 30141 [12], центральним концептом побудови будь-якої ПоТ-екосистеми визнається гарантування довіри (trustworthiness), що нерозривно об'єднує вимоги до функційної безпеки (safety), надійності та стійкості (resilience), оскільки відмови у таких мережах здатні спричинити критичні фізичні наслідки для персоналу та довкілля. З аналітичної точки зору, така регламентація на рівні міжнародних стандартів доводить: просте мережеве об'єднання промислового обладнання є

недостатнім. Сучасна індустріальна архітектура повинна на фундаментальному рівні містити інтегровані механізми безперервного технічного діагностування для підтримання заявлених показників експлуатаційної готовності та безпеки.

Однією з головних перешкод на шляху до повної інтеграції IoT-рішень є фундаментальна гетерогенність пристроїв, форматів даних та протоколів обміну. Ефективним інструментом подолання цієї проблеми є застосування семантичних онтологій – формальних представлень знань у вигляді концептів та зв'язків, які надають спільний словник і дозволяють системам однозначно інтерпретувати телеметричну інформацію без участі людини [13]. Для забезпечення такої семантичної сумісності індустрія спирається на низку міжнародних стандартів (зокрема, базові архітектури SSN і SOSA для сенсорних мереж, SAREF для розумних приладів та SemIoT для інтелектуальних агентів) [14, 15]. Незважаючи на об'єктивні технічні обмеження щодо вербозності даних та обчислювальної складності, застосування онтологічного підходу виходить далеко за межі простого опису датчиків. У контексті складного середовища IoT онтології є ефективним засобом формалізації предметної галузі та систематизації інженерного досвіду. Це дозволяє використовувати їх як базис для систем підтримки прийняття рішень, зокрема у задачах структурного синтезу та багатокритеріального вибору математичних моделей для інтелектуального діагностування обладнання [16].

Технологічний стек та обчислювальні парадигми. Проте, якщо семантичні онтології вирішують проблему логічної сумісності даних та формалізації знань, то практична реалізація систем безперервного моніторингу стану обладнання вимагає створення надійної фізичної та мережевої інфраструктури для збору цих даних. Архітектура IoT реалізується через багат шаровий технологічний стек, де кожен шар відповідає за специфічні завдання. Можна виділити кілька рівнів обробки даних [17].

1. Шар сприйняття (Perception Layer). Фізичні датчики, RFID-мітки та камери, що збирають дані з реального світу.
2. Мережевий шар (Network Layer). Забезпечує передачу даних через шлюзи, роутери та комутатори.

3. Шар проміжного ПЗ (Middleware/Processing Layer). Відповідає за агрегацію, фільтрацію та зберігання даних.

4. Шар додатків (Application Layer). Специфічне програмне забезпечення для моніторингу, аналітики та управління бізнес-процесами.

5. Шар бізнес-логіки. Верхній рівень, де приймаються стратегічні рішення на основі обробленої інформації.

Cloud, Fog та Edge Computing в IoT. З точки зору технічного діагностування, агрегована на нижніх рівнях телеметрія вимагає значних обчислювальних ресурсів для виявлення прихованих дефектів. Тому вибір місця обробки даних є критичним компромісом між обчислювальною потужністю, автономністю, безпекою та часом реакції [18].

1. Cloud Computing (Централізована модель). Хмара пропонує необмежені ресурси для довгострокового аналізу та зберігання великих масивів даних. Проте вона має високу затримку та залежність від стабільного інтернету.

2. Fog Computing (Туманні обчислення). Це децентралізована інфраструктура, де обробка відбувається на проміжних вузлах (туманних вузлах), розташованих близько до джерел даних. Це дозволяє зменшити обсяг даних, що передаються в хмару, та знизити затримку.

3. Edge Computing (Периферійні обчислення). Обробка відбувається безпосередньо на кінцевих пристроях (сенсорах, контролерах). Це критично для систем реального часу, таких як робототехніка або автономні транспортні засоби, де рішення мають прийматися за мілісекунди.

Протоколи комунікації. Для гарантування безперервності процесу збору телеметрії та діагностування між розподіленими обчислювальними рівнями, IoT використовує специфічний набір протоколів, оптимізованих для енергоефективності та надійності в умовах завад [17]. На рівні мережі можна виділити наступні протоколи.

1. Wi-Fi. Універсальний бездротовий стандарт для локальних мереж, який забезпечує високу швидкість передачі даних, але має відносно високе енергоспоживання і менший радіус дії.

2. Industrial Ethernet. Адаптований для промислових умов стандарт мережових технологій (IEEE 802.3), що забезпечує надійний обмін даними в реальному часі між компонентами автоматизації, контролерами та робочими станціями.

3. Zigbee. Бездротовий стандарт з низьким енергоспоживанням, призначений для організації сенсорних мереж, що передають дані на короткі відстані (до 200 м).

4. LoRaWAN. Тип мережі LPWAN (енергоефективна мережа дальнього радіуса дії), оптимізована для пристроїв із живленням від батарейок; дозволяє передавати невеликі обсяги даних на значні відстані (в тому числі в сільському господарстві чи логістиці).

5. LTE-M / NB-IoT. Стандарти мобільного зв'язку, спеціально адаптовані для IoT. Вони забезпечують широке покриття, глибоке проникнення сигналу в будівлі та енергоефективність при підключенні величезної кількості пристроїв.

6. Bluetooth / BLE (Bluetooth Low Energy). Використовується для економічного бездротового обміну даними між пристроями на дуже коротких відстанях.

На прикладному рівні можна виділити наступні протоколи [17].

1. MQTT (Message Queuing Telemetry Transport). Легкий протокол, що використовує модель «видавець-підписник». Він ідеально підходить для вузьких каналів зв'язку та пристроїв з обмеженим живленням.

2. CoAP (Constrained Application Protocol). Орієнтований на роботу через UDP, що робить його дуже швидким. Він легко інтегрується з веб-службами (HTTP) та підтримує багатоадресні запити.

3. OPC UA (OPC Unified Architecture). Промисловий стандарт для обміну даними між пристроями різних виробників. Забезпечує високий рівень безпеки та вбудовану інформаційну модель.

4. DDS (Data Distribution Service). Використовується в критично важливих системах реального часу, де необхідна гарантована доставка даних без центрального сервера.

Отже, ПоТ – це не просто мережа датчиків, а комплексна інтелектуальна екосистема, яка трансформує фізичний світ у цифровий формат і дозволяє керувати виробничими процесами з високою точністю та ефективністю. Успішне функціонування та масштабування таких рішень безпосередньо залежить від гармонійного поєднання архітектурних стандартів, семантичних описів та надійних технологій зв'язку. У перспективі подальший розвиток ПоТ буде нерозривно пов'язаний із переходом до парадигми Industry 5.0 та 6.0, де головними пріоритетами стануть стійкий розвиток, орієнтація на потреби людини та широке впровадження штучного інтелекту для автономного управління.

Разом із тим, зростаюча структурна складність ПоТ-інфраструктур, багат шаровість протоколів та висока ціна помилки у промисловому середовищі висувають жорсткі вимоги до експлуатаційної готовності такого обладнання. З іншого боку, генерація величезних масивів телеметричних даних на польовому та граничному рівнях створює інформаційний базис для розгортання сучасних систем діагностування промислового обладнання. З огляду на це, фундаментальною передумовою життєздатності сучасних ПоТ-екосистем стає безумовне забезпечення експлуатаційної готовності промислового обладнання та мінімізація ризиків прихованих відмов, вимоги до яких детально розглядаються у наступному підрозділі.

1.1.2 Аналіз вимог до надійності промислового обладнання з використанням індустріального інтернету речей

Основою для формування вимог до сучасного промислового обладнання з використанням ПоТ є класична теорія надійності та безпеки комп'ютерних систем. Відповідно до базового стандарту ДСТУ 2860-94 [19], надійність (dependability) визначається як властивість об'єкта зберігати у часі в установлених межах значення всіх параметрів, які характеризують здатність виконувати потрібні функції в заданих режимах та умовах застосування, технічного обслуговування, зберігання та транспортування. Надійність є складною комплексною властивістю,

яка, залежно від призначення об'єкта, включає такі підвластивості як безвідмовність (властивість об'єкта виконувати потрібні функції в певних умовах протягом заданого інтервалу часу чи наробітку), ремонтпридатність (властивість об'єкта бути пристосованим до підтримання та відновлення стану, в якому він здатний виконувати потрібні функції за допомогою технічного обслуговування та ремонту), довговічність (властивість об'єкта виконувати потрібні функції до переходу у граничний стан при встановленій системі технічного обслуговування та ремонту) та зберезуваність (властивість об'єкта зберігати в заданих межах значення параметрів, що характеризують здатність об'єкта виконувати потрібні функції, під час і після зберігання та (чи) транспортування) [19].

Більшість сучасного промислового обладнання, що інтегрується в середовище індустріального інтернету речей (ІІІТ), класифікується як відновлювані системи. Для такого класу систем, де ключовою вимогою є безперервність надання сервісів та виконання технологічних процесів, одним із найбільш важливих комплексних показників стає готовність (availability). Готовність за ДСТУ [19] визначається як властивість об'єкта бути здатним виконувати потрібні функції в заданих умовах у будь-який час чи протягом заданого інтервалу часу за умови забезпечення необхідними зовнішніми ресурсами. В межах теорії надійності, готовність розглядається як інтегральна властивість, яка об'єднує безвідмовність та ремонтпридатність (властивість об'єкта бути пристосованим до підтримання та відновлення стану, в якому він здатний виконувати потрібні функції за допомогою технічного обслуговування та ремонту).

Кількісною мірою цієї властивості є коефіцієнт готовності (КГ) – ймовірність того, що система опиниться в працездатному стані у будь-який час, крім планових періодів, протягом яких її використання за призначенням не передбачається. Сучасне критичне промислове обладнання часто належить до систем високої готовності, рівень яких вимірюється кількістю «дев'яток» після коми (наприклад, 0.999 або 0.99999). Забезпечення такого рівня означає жорсткі

обмеження на сумарний час простою системи (до кількох хвилин або навіть секунд на рік).

Класифікація систем за критичністю. Сучасні комп'ютерні системи класифікують за критичністю залежно від наслідків їхнього відмовлення.

1. Safety-critical (небезпечні при відмовах) системи – це ті, у яких відмова може призвести до катастрофічних наслідків (загибелі людей, екологічної катастрофи, великих матеріальних втрат). До таких систем належать, наприклад, автоматизовані системи управління на АЕС та малих модульних реакторах (SMR), навігаційні та авіаційні системи.

2. Mission-critical (критичні для виконання місії) системи забезпечують виконання конкретної стратегічної задачі, наприклад, космічні комплекси.

3. Security-critical (безпеково-критичні) системи охоплюють інфраструктури, де пріоритетом є захист від кібератак (енергетичні мережі, військові об'єкти).

4. Business-critical (бізнес-критичні) системи забезпечують безперервність бізнес-процесів (банківські і фінансові платформи, IT-інфраструктура).

Через це вимоги до експлуатації й сертифікації таких систем строго регламентовані міжнародними стандартами з функційної безпеки. Базовим стандартом є ІЕС 61508 (серія стандартів для електричних/електронних програмованих систем з безпеки), який задає вимоги до аналізу ризиків, показників довірчої придатності і організації процесу розробки. У багатьох країнах ІЕС 61508 прийнятий у вигляді національних стандартів. Зокрема, в Україні ці норми гармонізовано через ДСТУ EN 61508-1:2019 Функційна безпечність електричних, електронних, програмованих електронних систем, пов'язаних із безпекою. Частина 1. Загальні вимоги [20]. Наведені стандарти визначають методи оцінювання надійності, процедури верифікації та валідації, а також вимоги до організації супровідних процесів (менеджмент життєвого циклу).

Причини неготовності: проблема прихованих відмов. Неготовність (відсутність належної працездатності) промислового обладнання може виникати з різних причин. До основних факторів належать апаратні дефекти (вихід з ладу електроніки, механічні зношування, деградація сенсорів) та програмні збої (помилки програмного забезпечення, збої комунікації тощо). Також актуальними є проблеми через інтерференцію та завади у бездротових каналах зв'язку [21], відмови через кібератаки [22]. Особливо небезпечними є приховані відмови (латентні відмови), які залишаються непоміченими звичайними процедурами контролю. Вони можуть відноситись як до програмних та і до апаратних. У складних розподілених системах ІоТ звичайні операційні моніторингові інструменти можуть не виявити негайно дефект. Латентна помилка може тривалий час не впливати на очевидні показники роботи, тому вона накопичується в системі й залишається до того моменту, коли відбувається її екстремальне спрацювання. Згодом така прихована відмова може поєднатися з іншими несприятливими факторами і раптово ескалювати у критичну аварійну ситуацію.

Рекомендації з експлуатації вказують, що саме латентні відмови найбільше знижують реальну готовність системи. Як зазначає відомий дослідник Дж. Різон (1990) [23], приховані помилки являють собою найбільшу загрозу безпеці в складній системі, оскільки часто залишаються непоміченими і здатні викликати низку різноманітних активних помилок. Якщо зосереджувати увагу лише на явних (активних) помилках, латентні несправності залишаються в системі, і їх накопичення фактично робить систему більш схильною до майбутніх відмов. В контексті ІоТ зростання складності архітектури, великий обсяг даних і мережеві затримки ще більше ускладнюють раннє виявлення прихованих відмов. Таким чином, вони підривають реальний рівень безпеки та готовності промислових систем, і саме боротьба з латентними відмовами є важливою задачею аналізу надійності в середовищі Industrie 4.0/5.0.

При розгортанні ІоТ-інфраструктури, підтримання високих показників готовності вимагає переходу від класичних методів проєктування до проактивної

експлуатації. Якщо традиційно висока надійність забезпечувалася переважно апаратним резервуванням (використанням надмірності), то для розподілених систем ПоТ цей метод має суттєві економічні та структурні обмеження. Відповідно, головний акцент зміщується на етап експлуатації: висока готовність досягається через неперервний контроль працездатності, своєчасне технічне діагностування та оперативне відновлення функцій. Водночас, ефективність процесів відновлення прямо залежить від того, наскільки повно і достовірно система здатна ідентифікувати власні дефекти до того, як вони призведуть до критичних наслідків.

1.2 Аналіз методів і технологій забезпечення надійності промислового обладнання систем індустріального інтернету речей з використанням технічного діагностування та цифрових двійників

У класичній інженерії надійність описується ймовірністю безвідмовної роботи протягом визначеного часу. Для складних систем ПоТ цей показник доповнюється параметрами цілісності даних та доступності сервісів. Математична модель надійності системи, що складається з багатьох інтелектуальних вузлів, базується на інтенсивності відмов λ та середньому часі між відмовами (mean time between failure, MTBF). Впровадження технологій ПоТ спрямоване на мінімізацію λ шляхом виявлення перехідних процесів від нормальної роботи до стадії зносу задовго до фактичної відмови обладнання, тобто фактично впроваджуючи підхід прогностного (предиктивного) обслуговування [24]. Так у роботі [25] автори дослідили вплив від впровадження прогностного обслуговування у виробництві на економічні та операційні показники. Згідно з case-study, економія для їх сценарію становить до 27,05% порівняно з профілактичним технічним обслуговуванням і 63,43% порівняно з коригувальним технічним обслуговуванням, що виконується після виходу з ладу, є суттєвою фінансовою перевагою для компаній. Прогнозне обслуговування може запобігти суттєву частину помилок на виробничій лінії, що призводить до значного зменшення виробничих відходів, спричинених

проблемами з технічним обслуговуванням обладнання [25] і підвищити готовність виробничої лінії завдяки уникненню незапланованих простоїв виробництва.

1.2.1 Аналіз існуючих методів і технологій технічного діагностування промислового обладнання з використанням індустріального інтернету речей. Обмеження та проблема прихованих відмов

Забезпечення високої готовності складного промислового обладнання в середовищі індустріального інтернету речей неможливе без ефективних засобів визначення його технічного стану. Згідно з базовим стандартом ДСТУ 2389-94 [26], технічний стан – це стан, який характеризується в певний момент часу, за певних умов зовнішнього середовища значеннями параметрів, установлених технічною документацією на об'єкт. Базовою процедурою для його визначення є контроль технічного стану – перевірення відповідності значень параметрів об'єкта вимогам технічної документації та визначення на цій основі одного із заданих видів технічного стану (наприклад, працездатний чи непрацездатний) у даний момент часу. Ширшим поняттям є технічне діагностування – процес визначення технічного стану об'єкта з заданою точністю, основними завданнями якого є безпосередньо сам контроль технічного стану, пошук місця та виявлення причин відмови (несправності), а також прогнозування технічного стану на наступний інтервал часу [26].

Залежно від умов проведення та характеру впливів, розрізняють системи робочого та тестового технічного діагностування. Системи робочого (апаратного) діагностування забезпечують безперервний контроль об'єкта під час подачі на нього робочих впливів у процесі виконання функцій за призначенням. Натомість методи тестового діагностування передбачають подачу спеціально сформованих тестових впливів, що зазвичай вимагає переривання нормальної роботи системи, але забезпечує вищу глибину пошуку несправностей [26].

Завдяки широкому впровадженню ПоТ стало можливим безперервне віддалене отримання експлуатаційних даних у реальному часі, що забезпечило

перехід від реактивного обслуговування до прогнозової діагностики та управління надійністю. У сучасній науковій літературі методи виявлення та діагностики несправностей (Fault Detection and Diagnostics, FDD) зазвичай класифікують на сигнал-орієнтовані, модель-орієнтовані та знання-/дані-орієнтовані підходи [27-29].

1. Сигнал-орієнтовані методи базуються на аналізі фізичних вимірюваних параметрів, таких як вібрація, акустичні сигнали або температурні профілі. Вони традиційно використовуються для діагностики обертових машин і енергетичного обладнання. Сучасні дослідження демонструють, що поєднання аналізу сигналів із методами глибокого навчання дозволяє автоматично виявляти приховані дефекти та деградаційні процеси на ранніх стадіях розвитку.

2. Модель-орієнтовані підходи ґрунтуються на використанні аналітичних або математичних моделей об'єкта. Вони передбачають порівняння реальної поведінки системи з її очікуваною моделлю, що дозволяє виявляти відхилення, пов'язані з несправностями. До таких методів належать алгоритми оцінювання стану, зокрема фільтри Калмана, а також концепція цифрових двійників (Digital Twin) або цифрових тіней (Digital Shadow). Останні створюють динамічне відображення фізичного активу та дозволяють виконувати перевірку достовірності даних у реальному часі.

3. Знання- та дані-орієнтовані методи активно розвиваються у зв'язку з доступністю великих обсягів експлуатаційних даних. Знання-орієнтовані системи використовують експертні правила для діагностики відомих сценаріїв відмов, тоді як дані-орієнтовані підходи застосовують машинне навчання та аналітику для виявлення нових закономірностей і прогнозування майбутніх збоїв. Такі методи демонструють високу ефективність у складних багатопараметричних системах, характерних для сучасної енергетики та промисловості.

У середовищі ІІoТ ці методи в рамках систем діагностування функціонують у замкненому діагностичному контурі з об'єктом діагностування. На першому етапі сенсори збирають телеметричні дані в реальному часі. Далі на рівні граничних обчислень (edge computing) виконується первинна обробка: фільтрація

шумів, виділення ознак та локальна оцінка стану. Далі більш складні цифрові моделі або алгоритми машинного навчання аналізують відхилення між нормальним і фактичним режимами роботи. Результати аналізу передаються до систем прийняття рішень, де формуються рекомендації щодо технічного обслуговування або здійснюється автоматичне коригування режимів роботи обладнання. Хмарні платформи (cloud computing) при цьому забезпечують централізований моніторинг і довгострокове прогнозування на основі накопичених даних

Оцінювання ефективності систем технічного діагностування, що функціонують у такому замкненому контурі, здійснюється за допомогою показників повноти, тривалості, глибини та достовірності [26]. Однак ключовою проблемою при впровадженні проактивного діагностування є проблема достовірності. Для забезпечення достовірного розпізнавання стану обладнання в основі таких систем лежить використання діагностичних моделей – моделі, в яких для певних входних (тестових або робочих) впливів визначені очікувані еталонні результати. Діагностична модель формалізує еталонну поведінку об'єкта та чітко визначає межу між нормальним (працездатним) станом та відмовою. Найбільш розповсюдженим підходом у промисловості є побудова діагностичної моделі на базі допускового контролю параметрів. Наприклад, для промислового насоса така модель спирається на контроль фізичних характеристик (тиску на вході та виході, температури тощо), для яких встановлюються межі допусків, що залежать від специфіки конкретного обладнання. Проте, в реальних умовах експлуатації жорсткі допуски можуть вичерпати свій потенціал через змінність середовища, старіння компонентів та зашумленість сенсорних даних. Ця неідеальність аналітичних алгоритмів і сенсорів при розпізнаванні істинного технічного стану в єдиній системі «об'єкт–засоби діагностування» призводить до появи трьох типів помилок [26].

1. Помилка першого роду (ω_1): об'єкт є працездатним, але розпізнається системою як непрацездатний (ризик поставника, хибне бракування). Це відбувається, коли параметри виходять за встановлені межі допускового

контролю через нестандартні, але безпечні режими роботи обладнання. А також призводить до необґрунтованих зупинок виробництва та зниження функції готовності та економічних втрат.

2. **Помилка другого роду (ω_2):** об'єкт є непрацездатним, але розпізнається як працездатний (ризик замовника, хибне визнання норми).

3. **Помилка третього роду (ω_3):** об'єкт непрацездатний, відмовив елемент І, проте розпізнається як непрацездатний внаслідок відмови іншого елемента J (ризик діагноста, хибне визначення причини або місця відмови). Призводять до некоректного відновлення або збільшення часу його виконання.

Саме помилки другого (ω_2) та третього (ω_3) роду є найнебезпечнішими для складного критичного обладнання, оскільки вони формують базис для виникнення прихованих відмов (latent failures), що може переводити систему у небезпечний стан. Як вже було зазначено у попередньому підрозділі, приховані відмови можуть локалізуватися в певних компонентах об'єкта діагностування, залишаючись непоміченими засобами оперативного контролю. Помилки третього роду (ω_3) додатково унеможливають коректне застосування процедур відновлення, оскільки система починає відновлювати не той вузол. А за певних умов прихована відмова ескалює у критичну аварійну ситуацію.

Таким чином, сучасні системи технічної діагностування в ПОТ поєднують фізичні вимірювання, математичне моделювання та інтелектуальну аналітику, проте жорсткі детерміновані алгоритми вичерпують свій потенціал в умовах змінності середовища і умов експлуатації. Необхідність мінімізації помилок 2-го та 3-го роду і виявлення прихованих відмов вимагає переходу до інтелектуальних систем діагностування (ІСД) на основі алгоритмів машинного навчання, здатних адаптуватися до нових шаблонів деградації шляхом безперервного донавчання в процесі експлуатації. У цьому контексті концепція донавчання ІСД полягає саме в динамічному коригуванні меж допускового контролю діагностичної моделі на основі нових експлуатаційних даних. Адаптуючи цю межу між нормою та відмовою під індивідуальні умови роботи конкретного обладнання, система здатна безперервно еволюціонувати: звужуючи або розширюючи межі допусків,

ІСД мінімізує хибні зупинки (помилки першого роду) та ефективно усуває небезпечні приховані відмови (помилки другого та третього роду).

Таким чином, традиційні сигнал- та модель-орієнтовані методи не втрачають своєї актуальності, залишаючись надійним фундаментом для первинного збору телеметрії, вилучення базових ознак та контролю параметрів на нижньому рівні. Проте їх використання як єдиного інструменту для діагностування складного промислового обладнання в середовищі ІоТ стикається з низкою критичних обмежень. Комплексна мотивація переходу до нових методів зумовлена кількома факторами. По-перше, це стрімке зростання структурної складності кіберфізичних систем, через що побудова точних аналітичних моделей фізики процесів стає обчислювально надскладною задачею. По-друге, жорсткі детерміновані алгоритми не здатні адаптуватися до так званого «дрейфу концепту» (concept drift) – динамічної зміни умов експлуатації, режимів навантаження та природного старіння компонентів. По-третє, критичною залишається проблема достовірності: традиційні засоби часто не здатні розпізнати неочевидні патерни деградації, генеруючи помилки другого та третього роду, які призводять до накопичення прихованих відмов.

Отже, забезпечення високої готовності вимагає еволюційного переходу до інтелектуальних систем діагностування на основі алгоритмів машинного навчання чи інтеграції методів адаптації у сигнал- та модель-орієнтовані методи. Сучасні ІСД не відкидають класичні напрацювання, а інтегрують їх у гібридні архітектури, де data-driven методи надбудовуються над фізичними вимірюваннями. Це дозволяє знаходити складні нелінійні залежності у великих масивах даних, мінімізувати ризики хибного діагностування та забезпечувати проактивне управління надійністю завдяки здатності системи безперервно еволюціонувати шляхом оперативного донавчання.

1.2.2 Інтелектуальні системи діагностування промислового обладнання з використанням індустріального інтернету речей. Концепція донавчання в діагностиці

Як було показано у попередньому підрозділі, головним недоліком традиційних детермінованих систем діагностування та контролю є їхня жорсткість та нездатність виявляти неочевидні патерни деградації. Це призводить до генерації помилок другого та третього роду і, як наслідок, накопичення прихованих відмов. Вирішення проблеми лежить у площині переходу до інтелектуальних систем діагностування (ІСД) – систем діагностування, що здатні адаптуватися до нових умов експлуатації на основі інтелектуального аналізу даних. У контексті надійності промислового обладнання ПоТ, їхня «інтелектуальність» полягає у здатності не лише фіксувати вихід параметрів за встановлені допуски, але й виявляти аномалії, прогнозувати залишковий ресурс та адаптуватися до нових умов шляхом машинного навчання (із донавчанням моделі).

Основою сучасних інтелектуальних систем діагностування (ІСД) є методи глибокого навчання, зокрема згорткові (CNN) та рекурентні (RNN/LSTM) нейронні мережі. Вони забезпечують автоматичне виявлення складних нелінійних деградаційних патернів у багатовимірних сенсорних даних, що демонструє значно вищу ефективність порівняно з традиційними ML-алгоритмами. Ефективність застосування таких моделей для точної класифікації технічних станів у ПоТ-системах підтверджується сучасними дослідженнями [30].

Проте впровадження статичних моделей штучного інтелекту (ШІ) для критичного промислового обладнання стикається з фундаментальними викликами.

1. Дефіцит даних (Data Scarcity) та незбалансованість класів. У промислових системах (особливо критичного призначення) відмови є рідкісними подіями. Відсутність повноцінних прикладів, патернів і траєкторій відмов обмежує можливості навчання моделей та знижує їхню здатність до узагальнення

у реальних умовах експлуатації. Необхідність створення ефективних моделей навіть за умов мінімальних навчальних вибірок є однією з головних проблем сучасної діагностики [31]. Частковим вирішенням цієї проблеми є генерація синтетичних даних або застосування нейромереж [31], що враховують фізичні закони (Physics-Informed Neural Networks, PINN) [32], які дозволяють компенсувати брак даних використанням фізичних моделей як регуляризуючого фактора.

2. Дрейф концепту (Concept Drift) – у процесі експлуатації фізичні властивості обладнання неминуче змінюються внаслідок зносу та деградації матеріалів, або певним чином можуть змінюватись умови експлуатації. Статична модель ШІ, навчена на даних нового обладнання по одних умовах, з часом втрачає свою релевантність. Цей ефект призводить до різкого падіння достовірності діагностування [33].

Для подолання цих обмежень архітектура сучасних ІСД повинна відмовитися від парадигми одноразового навчання на етапі проектування. Згідно із запропонованим у дослідженнях підходом [33, 34], ІСД має функціонувати на основі концепції безперервного донавчання (Incremental / Continuous Learning) в процесі експлуатації. У межах дисертації під терміном «донавчання» розуміється процес адаптації існуючої діагностичної моделі за допомогою нових експлуатаційних даних з метою підвищення адекватності моделі, компенсації змін у середовищі функціонування (дрейфу концепту), врахування раніше невідомих закономірностей (прихованих відмов). Донавчання засобів діагностування дозволяє поступово підвищувати показник достовірності діагностування.

Практична доцільність донавчання підтверджується експериментальними дослідженнями. У роботі [33] показано, що класичні моделі діагностування, побудовані на фіксованій множині станів, демонструють суттєве зниження точності при появі нових, раніше невідомих відмов, що зумовлено відсутністю механізмів адаптації до змін розподілу даних. Натомість застосування інкрементального підходу, зокрема еволюційних класифікаторів (eClass), дозволяє інтегрувати нові класи без повного перенавчання моделі та забезпечує

стабільні показники розпізнавання в умовах розширення простору станів. Аналогічно, у дослідженні [31] продемонстровано, що використання адаптивних методів навчання байєсівських мереж (зокрема QMAP) є більш ефективним у задачах діагностування при обмежених вибірках порівняно з класичними методами оцінювання параметрів (MLE), які потребують значно більшого обсягу даних для досягнення співставної точності. Це експериментально обґрунтовує теоретичне припущення даного дослідження про зростання показника достовірності діагностування D після кожного циклу донавчання.

В архітектурі розподілених IoT-систем реалізація концепції донавчання в тому числі визначається топологією збору даних та місцем проведення обчислень. З огляду на це, можна виділити три основні підходи до адаптації моделей.

1. Локальне донавчання (Edge/Local Learning): здійснюється безпосередньо на рівні граничних обчислень конкретного вузла після виявлення локальної прихованої відмови або під час планового технічного обслуговування. Цей підхід забезпечує максимальну автономність та конфіденційність, проте модель залишається обмеженою індивідуальним досвідом лише одного екземпляра обладнання [35].

2. Централізоване донавчання на сервері (Cloud/Server Learning): традиційний підхід, за якого сирі масиви телеметрії з усього парку обладнання передаються до центрального сервера для офлайн- чи онлайн-перенавчання загальної моделі. Хоча це дозволяє використовувати потужні обчислювальні ресурси, такий підхід створює надмірне навантаження на канали зв'язку та порушує вимоги конфіденційності даних критичної інфраструктури [33, 34].

3. Федеративне навчання (Federated Learning): виступає сучасним компромісом між локальними та централізованими архітектурами. Впроваджує децентралізований метод навчання моделей, який є особливо корисним у середовищах граничного навчання, де дані розподілені між декількома пристроями (наприклад сукупність IoT пристроїв). Воно навчає спільну модель шляхом ітеративних оновлень з використанням локальних даних, уникаючи централізації конфіденційної інформації. Ця техніка сприяє спільній навчальній

роботі, одночасно забезпечуючи захист конфіденційності даних і мінімізуючи накладні витрати на комунікацію [36].

Системні обмеження донавчання та вимоги до довірчоздатності. Незважаючи на очевидні переваги, процес оперативного донавчання моделей ШІ вносить нові ризики в експлуатацію критичних систем. Спроби адаптації моделей через повторне навчання часто супроводжуються явищем *catastrophic forgetting* [34], коли нові знання витісняють раніше засвоєні патерни відмов. В тій самій роботі пропонується вирішення проблеми. Також відповідно до сучасних моделей якості систем штучного інтелекту, ключовими вимогами є довірчоздатність (Trustworthiness) та пояснюваність (Explainability). Для безпекокритичних систем важливою є інтерпретованість моделей (Explainable AI, XAI), що дозволяє експертам обґрунтовувати діагностичні рішення та сприяє їх інтеграції у процеси технічної експлуатації [37]. Крім того, динамічна зміна параметрів ІСД під час донавчання потребує витрат часу, а впровадження може вимагати простою системи. Практична реалізація також ускладнюється обмеженими обчислювальними ресурсами edge-пристроїв.

Таким чином, сучасні інтелектуальні системи діагностування в середовищі ІоТ забезпечують суттєво вищу ефективність у виявленні деградаційних процесів порівняно з традиційними підходами. Водночас їх застосування в умовах дефіциту даних, дрейфу характеристик обладнання та змінних режимів експлуатації виявляє обмеженість статичних моделей і обумовлює необхідність використання безперервного донавчання. А федеративне навчання дозволяє масштабувати цей ефект на сукупність пристроїв.

Разом із тим, адаптація моделей супроводжується новими викликами, пов'язаними з довірчоздатністю, пояснюваністю та ризиком втрати раніше засвоєних знань. Це свідчить про потребу переходу від простої адаптивності до керованого перенавчання (relearning), яке дозволяє зберігати узгодженість знань у процесі експлуатації та забезпечувати довготривалу достовірність діагностування складного ІоТ-обладнання. Часткові вирішення цих проблем та підкреслення

сильних сторін ІСД можливо забезпечити за допомогою інтеграції технології цифрових двійників в ІСД, що і буде розглянуто далі.

1.2.3 Аналіз технології Цифрових Двійників для забезпечення надійності промислового обладнання з використанням індустріального інтернету речей. Визначення, класифікація, зв'язок з життєвим циклом обладнання

Як було визначено у попередньому підрозділі, інтелектуальні системи діагностування потребують інструментарію для безпечного донавчання, генерування розмічених даних та прогнозування станів. Хоча технологія цифрових двійників (ЦД, Digital Twins, DT) не є єдиним інструментом для вирішення зазначених вище проблем, вона є одним з перспективних напрямків для такого роду задач: тестування гіпотез без ризику для реального обладнання, подолання дефіциту даних та забезпечення платформи для безперервного навчання моделей.

Варто зазначити, що детальний огляд концепції цифрових двійників, їх класифікація та варіанти застосування у різних доменах гарантоздатних систем індустріального інтернету речей (включаючи мобільні кіберфізичні платформи на прикладі безпілотних літальних апаратів) був попередньо проведений та опублікований автором у наукових працях [38, 39], а також у колективній монографії [40]. Спираючись на ці результати і додатково проведений огляд публікацій, у даному підрозділі наведено загальний аналіз технології ЦД, тоді як у наступному підрозділі увагу буде зосереджено безпосередньо на методах її інтеграції в ІСД для забезпечення надійності промислового обладнання з ІоТ.

Визначення та історія. Досить часто початком формування концепції і терміну ЦД вважають виступ у 2002 році Майкла Грівза на університетському курсі по темі створення центру управління життєвим циклом продукту (PLM) [41]. Хоча й до цього були певні напрацювання з ідеєю віддзеркалення фізичного світу на цифровий. Причому формалізація технології ЦД хоча й мала кілька важливих етапів, але все ще продовжується досі. Існує багато визначень від різних

установ, компаній та консорціумів, а стандарти все ще перебувають у процесі активної розробки. Наприклад, в стандарті ISO 23247-1:2021(en) Digital twin framework for manufacturing [42] визначення звучить так (у перекладі з англ): «відповідне призначенню цифрове представлення спостережуваного виробничого елемента із синхронізацією між елементом та його цифровим представленням». У Глосарії від Digital Twin Consortium написано наступне [43] (з англ.): «Цифровий двійник – це інтегроване віртуальне представлення реальних об'єктів і процесів на основі даних, із синхронізованою взаємодією із заданою частотою та точністю». Але тут все ще необхідно чітко враховувати такі важливі аспекти, як життєвий цикл і можливість використання інтегрованих фізичних, заснованих на правилах та інших моделей. Таким чином, узгоджене визначення сформоване в статті автора дисертації [44] звучить так: цифровий двійник можна розглядати як цифрове представлення фізичного об'єкта або процесу на основі інтеграції різних статистичних, заснованих на даних і фізичних моделей, які синхронізуються з фізичним об'єктом у режимі реального часу через двосторонній канал і використовуються для моніторингу, прогнозування різних типів відмов, оптимізації експлуатаційних витрат і сценаріїв «що-якщо» (what-if) на різних етапах життєвого циклу фізичного об'єкта.

Суміжні поняття. Для задач діагностування важливо розмежовувати ЦД від суміжних понять за рівнем інтеграції та напрямком потоків даних [45] (рис. 1.1).

1. Цифрова модель (Digital Model): цифрове представлення існуючого або запланованого об'єкта без автоматичного обміну даними між фізичною та віртуальною частинами. Усі зміни вносяться вручну (наприклад, статичні CAD/CAE моделі).

2. Цифрова тінь (Digital Shadow): наявний автоматизований односторонній потік даних від фізичного об'єкта до цифрового. Зміна стану фізичного об'єкта автоматично відображається у віртуальному, але не навпаки.

3. Цифровий двійник (Digital Twin): характеризується повністю автоматизованим двостороннім потоком даних. Віртуальна модель не лише

відображає стан фізичного об'єкта, але й здатна генерувати керуючі або коригувальні впливи на нього (зворотний зв'язок).

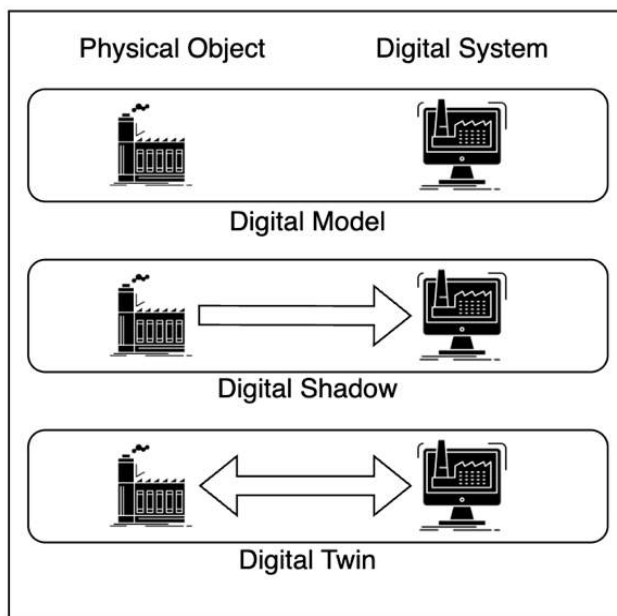


Рисунок 1.1 – Суміжні поняття ЦД за рівнем інтеграції цифрової моделі

Класифікація цифрових двійників. Оскільки ЦД працюють з гетерогенними кіберфізичними системами, їх класифікують за кількома базовими ознаками.

За ієрархією (рис 1.2). Розрізняють двійники компонентів (part/component twin), майна або активів (asset twin), систем (system twin) та процесів (process twin). У контексті ІІоТ, ЦД можуть представляти як прості датчики чи насоси, так і об'єднувати кілька виробничих підсистем у єдину «систему систем» (System of Systems) [38].

За методологією моделювання. Виділяють три основні підходи: фундаментальне) моделювання (базується на законах прикладних наук); моделювання на основі даних або data-driven (використання алгоритмів машинного навчання виключно на масивах телеметрії); гібридне моделювання, що поєднує фізичні закони із ML-алгоритмами [38].

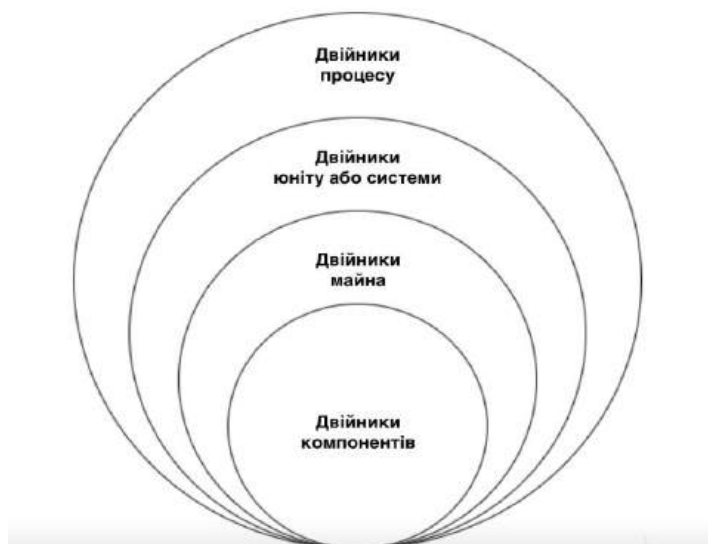


Рисунок 1.2 – Класифікація ЦД за ієрархією

За рівнем агрегації екземплярів класифікують ЦД наступним чином [46].

1. Екземпляр ЦД (Digital Twin Instance, (DTI): цифровий аналог одного конкретного існуючого об'єкта, який допомагає у моніторингу та прогнозуванні його індивідуальної поведінки.
2. Прототип ЦД (Digital Twin Prototype, (DTP): цей тип цифрового двійника описує прототипний фізичний артефакт. Він містить набори інформації, необхідні для опису та створення фізичної версії, яка дублює або є двійником віртуальної версії. Ці набори інформації включають, але не обмежуються, вимогами, повністю анотованою 3D-моделлю, специфікацією матеріалів, специфікацією процесів, специфікацією послуг та специфікацією утилізації.
3. Агрегат ЦД (Digital Twin Aggregator, (DTA): агреговане представлення всіх фізичних двійників одного типу (флоту). Використовується для загальних тестів, оцінювання та обміну досвідом, що є основою для реалізації федеративного навчання
4. Середовище ЦД (Digital Twin Environment): це інтегрований багатодоменний фізичний простір для роботи з цифровими двійниками з різними цілями (передбачення, дослідження, «що-якщо» сценарії).

Зрілість ЦД. Цифрові двійники мають певну градацію інтелекту або зрілості. Така зрілість залежить від точності моделі, якості даних, інтеграції, призначення тощо. Розглянемо рівні зрілості цифрових двійників [47]:

1. Описовий – візуальна та інша описова інформація про стан продукту, процесу. Це можуть бути різні інформаційні панелі, графіки, 3D-моделі.
2. Інформативний – такий ЦД має покращену інтеграцію з даними. Модель допомагає зробити певні висновки та знайти певні закономірності.
3. Прогнозний – історичні та поточні дані дають змогу таким ЦД прогнозувати певні події, умови, несправності.
4. Рекомендаційний – вдосконалена версія прогнозного ЦД. На основі зроблених прогнозів можна надавати певні рекомендації.
5. Розумний/автономний – може працювати автономно, реагуючи на дані в режимі реального часу, роблячи прогнози та надаючи відповідні команди фізичному аналогу.

Цифрова зрілість організації. Для створення повноцінного ЦД, необхідний достатній рівень цифровізації або зрілості (Digital Maturity) фізичної системи (цифрова зрілість виробництва, організації, фізичного об'єкта, тощо). Тому такі фізичні системи або об'єкти можна розділити на ті, що відповідають рівню цифровізації, і ті, що все ще потребують підготовчих робіт для цифровізації. Крім того, організації, які активно використовують ЦД, повинні не тільки пройти цифрову трансформацію, але й трансформувати деякі бізнес-процеси, корпоративну культуру – від вивчення працівників і завоювання їхньої довіри до ЦД до перетворення на гнучку компанію з більш мобільними командами. Більше того, процес створення ЦД слід розпочинати з детальної підготовчої роботи щодо встановлення цілей та ключових показників ефективності, оцінки цифрових прогалин, побудови дорожньої карти. Відповідні дослідження, документи та рамки були розроблені організацією Change2Twin та Acatech у їхньому Індексі зрілості Індустрії 4.0 [48]. А на рисунку 1.3 показано 7 рівнів цифровізації, які були запропоновані організацією Change2Twin відповідно до Maturity Index [47]. Перші 3 рівні (відсутність цифровізації, комп'ютеризація, рівень підключення)

можна віднести до процесу поступової цифровізації системи, де компоненти можуть використовувати ІТ-технології, але не працювати як інтелектуальна система в поєднанні з людським аналізом. Останні 4 рівні вже відповідають необхідному рівню цифровізації – вони характеризуються інтелектуальним, автоматизованим взаємозв'язком людей, пристроїв і систем в єдину систему.

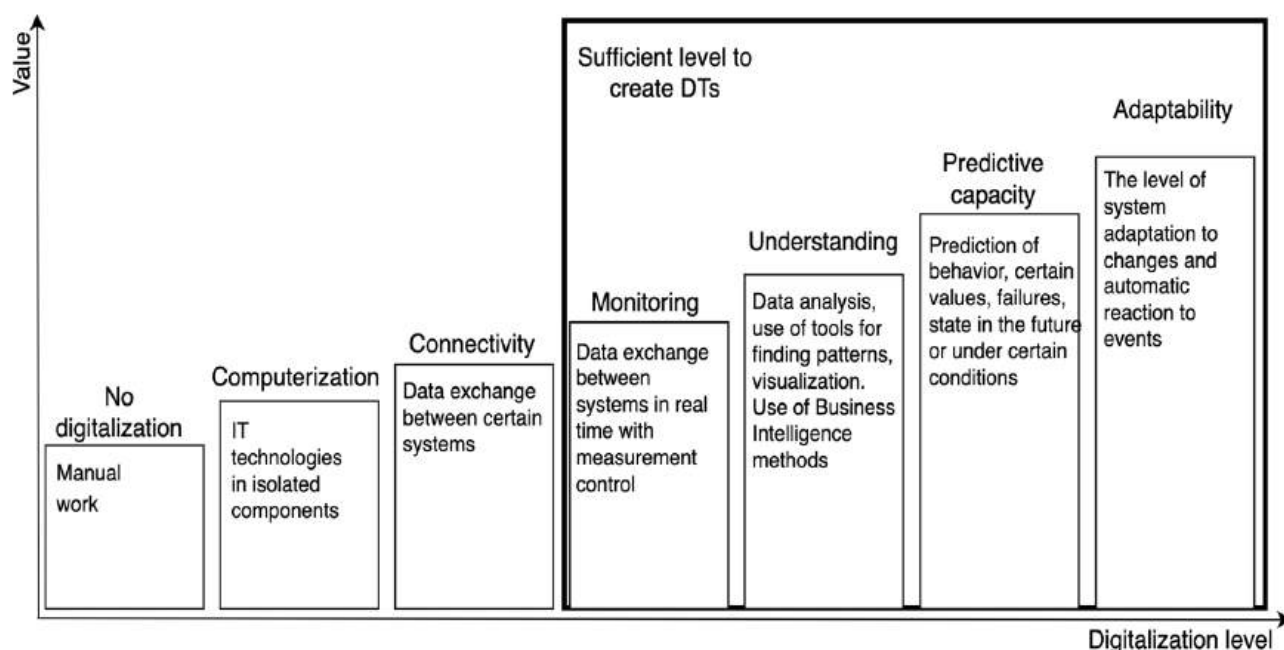


Рисунок 1.3 – Необхідний рівень цифровізації фізичного об'єкта та організації, що відповідає за створення ЦД [47]

Зв'язок ЦД із життєвим циклом фізичного об'єкта. Ключовою властивістю цифрового двійника, що відрізняє його від звичайної одноразової симуляції, є безперервний супровід фізичного об'єкта на всіх етапах його життєвого циклу (Product Lifecycle Management, PLM) [43]. Відповідно до класичного життєвого циклу розроблення та експлуатації систем (відомого як V-подібна модель), технологія цифрових двійників може застосовуватися на кожному з етапів: від початкового проєктування до безпосередньої експлуатації, виконуючи специфічні функції моніторингу та технічного діагностування.

1. Етап проектування: ЦД існує як «прототип» (DTP). Він використовується як віртуальне середовище для тестування нових концептів та попереднього оцінювання надійності до створення фізичного виробу.

2. Етап верифікації та валідації (V&V): ЦД відіграє критичну роль віртуального випробувального стенда. Це особливо важливо для унікального або безпекокритичного обладнання (наприклад, інформаційно-керуючих систем малих модульних реакторів, SMR), де проведення фізичних експериментів із введенням несправностей є небезпечним [43]

3. Етап експлуатації: ЦД перетворюється на «екземпляр» (DTI), що працює синхронно з фізичним об'єктом. Саме тут реалізуються завдання моніторингу, прогностного обслуговування та динамічного донавчання діагностичних моделей на основі даних телеметрії.

Таким чином, проведений огляд дозволяє систематизувати понятійний апарат, класифікаційні ознаки та рівні зрілості цифрових двійників, а також окреслити їх місце в життєвому циклі фізичних об'єктів і кіберфізичних систем. Розмежування суміжних концепцій, типів та архітектур ЦД створює методологічну основу для подальшого аналізу їх практичного застосування в задачах діагностування. Це, у свою чергу, дозволяє перейти до розгляду механізмів інтеграції цифрових двійників в інтелектуальні системи діагностування ІоТ.

1.2.4 Методи і обмеження використання Цифрових Двійників для забезпечення надійності промислового обладнання з використанням індустріального інтернету речей в інтелектуальних системах діагностування

Інтеграція технології цифрових двійників (ЦД) у контур інтелектуальних систем діагностування (ІСД) відкриває нові можливості для управління надійністю складного промислового обладнання в середовищі ІоТ.

Методи забезпечення надійності на основі ЦД. Одним із ключових методів, що реалізується за допомогою ЦД, є концепція віртуальних сенсорів

(Virtual Sensing). У складних або небезпечних умовах експлуатації (зокрема в енергетичних установках, таких як малі модульні реактори, чи у високоточному виробництві) встановлення додаткових фізичних датчиків часто є технічно неможливим або економічно недоцільним. У таких випадках цифрові двійники застосовують підхід віртуального сенсингу, поєднуючи фундаментальні фізичні закони з методами машинного навчання, орієнтованого на фізику (Physics-Informed Machine Learning, PIML). Це дозволяє реконструювати внутрішні параметри об'єкта на основі непрямой телеметрії. Саме гібридна інтеграція моделей у ЦД забезпечує підвищення точності оцінювання стану при обмеженій спостережуваності фізичної системи [49].

Другим важливим методом є безпечне тестування гіпотез. Цифрові двійники виступають у ролі віртуальних випробувальних стендів обладнання. Це дозволяє моделям в ІСД безпечно тестувати сценарії типу «що-якщо» та вивчати нові патерни прихованих відмов у віртуальному середовищі. Завдяки цьому ЦД виступає простором для реалізації локального та федеративного донавчання: виявлення неспецифікованої відмови на фізичному об'єкті ініціює процес оновлення віртуальної моделі, після чого нові вагові коефіцієнти безпечно валідуються і лише потім розгортаються на рівні граничних обчислень фізичного активу [50].

Забезпечення процесів донавчання ІСД та генерація синтетичних даних. Виявлення неспецифікованої (прихованої) відмови на фізичному об'єкті ініціює процес генерації відповідних симуляційних сценаріїв (What-If analysis) у цифровому двійнику. Саме у віртуальному середовищі відбувається процес локального донавчання моделі на нових патернах деградації. Після того як ЦД адаптувався до нових умов, розраховані оновлені параметри оперативного контролю та скориговані тестові послідовності розгортаються на рівні граничних обчислень фізичного активу. Крім того ЦД може й виступати інструментом підтримки, бо генерація синтетичних даних необхідна й для створення початкової версії III моделі, що буде ядром ІСД [47, 52].

Виявлення першопричин та причинно-наслідкове моделювання. За рахунок комбінації фундаментальних і статистичних моделей цифрові двійники дають можливість виявляти першопричини відмов у складних системах, виходячи із багатовимірних даних сенсорів та симуляційних трас деградації. Або ж підтримувати проведення післяаварійних розслідувань. Такий причинно-орієнтований підхід значно підвищує інтерпретованість діагностичних висновків і підтримує кваліфіковані технічні рішення в IoT-інфраструктурах [36].

Обмеження використання ЦД в інтелектуальних системах діагностування. По-перше, критичним обмеженням є обчислювальна складність та проблема затримок через опрацюванням даних у реальному часі за допомогою ЦД. Забезпечення двосторонньої синхронізації між фізичним об'єктом і ЦД у режимі реального часу вимагає надзвичайно високої пропускну здатності IoT-мереж. Водночас виконання ресурсоємних предиктивних моделей на хмарному або навіть граничному рівнях створює значні затримки у прийнятті рішень, що є неприпустимим фактором для безпекокритичних систем автоматики (I&C) [51]. Причому не завжди граничні вузли матимуть достатньо обчислювальної потужності.

Впровадження двосторонньої синхронізації між фізичним об'єктом та ЦД безпрецедентно розширює поверхню кібератак. Компрометація ЦД або комунікаційного каналу зловмисником може призвести до надсилання хибних діагностичних або керуючих команд безпосередньо на фізичний актив. З огляду на це, при проєктуванні ІСД необхідно керуватися парадигмою Security-Informed Safety (SIS) – враховувати не лише фізичну надійність компонентів, але й імовірність кібератак, розглядаючи безпеку та надійність як єдину комплексну властивість гарантоздатності [49].

Обмеження точності моделей залишаються критичним фактором. Цифровий двійник завжди є наближеним відображенням фізичного об'єкта, а похибки фізичних, статистичних або гібридних моделей можуть накопичуватися в процесі синхронізації з реальним обладнанням. Це особливо важливо враховувати коли ЦД використовується як еталонна модель [49].

Нарешті, слід враховувати загальні обмеження як обсяги даних, складність та вартість побудови точної моделі ЦД.

Висновки. Аналіз методів та обмежень використання ЦД дозволяє сформулювати ключовий підсумок щодо технологічного забезпечення надійності складного обладнання. Інтеграція ЦД у контур інтелектуальних систем діагностування є ефективним шляхом для реалізації концепції безперервного донавчання. Використання методів віртуального зондування (Virtual Sensing) та генерації синтетичних даних перетворює ЦД на безпечний випробувальний полігон. Це вирішує проблему дефіциту даних та «дрейфу концепту», дозволяючи локально та федеративно адаптувати моделі контролю до нових патернів деградації без ризику для фізичного активу, а також валідувати зміни у ІСД.

Водночас виявлені системні обмеження розкривають головне науково-прикладне протиріччя: прагнення мінімізувати приховані відмови та підвищити достовірність діагностування у довгостроковій перспективі стикається з витратами часу на саму адаптацію алгоритмів, застосуванням результатів донавчання, проблемами обчислювальної складності, кібер ризиками. Оскільки неконтрольоване або занадто часте донавчання може не підвищити, а навпаки – знизити загальний рівень готовності критичної системи, виникає потреба у розробленні спеціалізованого математичного апарату. Такий апарат, має забезпечити кількісне оцінювання та верифікацію впливу динаміки донавчання на показники готовності, що є предметом аналізу в наступному підрозділі.

1.3 Аналіз методів і моделей оцінювання надійності промислового обладнання систем індустріального інтернету речей з використанням технічного діагностування та цифрових двійників

1.3.1 Аналіз існуючих методів та математичних моделей оцінювання надійності промислового обладнання систем індустріального інтернету речей

Оцінювання надійності складних кіберфізичних систем (до яких належить промислове обладнання ІоТ) традиційно базується на двох класах математичних

моделей: статичних (логіко-ймовірнісних) та динамічних (моделях простору станів). До статичних моделей належать структурні схеми надійності (CSN, Reliability Block Diagrams) та дерева відмов (Fault Tree Analysis). Вони є ефективними для розрахунку показників невідновлюваних систем з жорсткою апаратною архітектурою. Проте головним недоліком логіко-ймовірнісних методів є їхня нездатність адекватно описувати процеси в часі: вони не враховують послідовність виникнення відмов, черги на обслуговування та переходи системи між станами з різним рівнем деградації або ефективності діагностування [54, 55].

До динамічних моделей належать марковські ланцюги з неперервним часом (Continuous-Time Markov Chains, CTMC), напівмарковські процеси та мережі Петрі. Для систем з відновленням і технічним діагностуванням марковські моделі є інструментом світового стандарту (IEC 61508 [20]), оскільки дозволяють формалізувати будь-які стратегії обслуговування у вигляді графа станів і переходів. Однак використання класичних марковських моделей для систем із функцією донавчання стикається з проблемою необхідності зміни параметрів у графі. Вирішенням цієї проблеми є перехід до мультифрагментних марковських моделей (МФММ) [43] та використання апарату структурно-автоматних моделей (САМ) [53]. Такий підхід дозволяє декомпонувати складну поведінку ІСД на окремі фрагменти (наприклад, базовий режим, режим дрейфу концепту, режим локального донавчання) з власними матрицями інтенсивностей переходів.

1.3.2 Аналіз існуючих програмних засобів моделювання та методів верифікації результатів моделювання

Для оцінювання показників надійності та готовності складних систем традиційно застосовуються аналітичні та імітаційні методи. Сучасні спеціалізовані інструменти моделювання надійності (наприклад, комерційні комплекси ReliaSoft, Isograph) та загальні математичні пакети (MATLAB, Python SciPy) зазвичай використовують класичні чисельні методи (наприклад, явні методи Рунге–Кутти) для розв'язання систем диференціальних рівнянь

Колмогорова-Чепмена. Водночас дослідження показують, що жорсткість моделей (проблема математичної «жорсткості», коли швидкості процесів відрізняються на кілька порядків) спричиняє може знижувати точність результатів вже на 4-5 знаку після коми, що особливо важливо для критичного обладнання з високими вимогами до точності [56]. Адаптивні системи діагностування характеризуються наявністю швидкоплинних процесів оновлення моделей машинного навчання, тривалість якого може складати години на тлі надзвичайно повільних процесів фізичної деградації промислового обладнання. Дослідження показують, що в таких умовах стандартні солвери накопичують чисельні помилки і можуть давати нестабільні, ненадійні результати [56, 57].

Обмеження методів верифікації та валідації (V&V) Окрім обчислювальних труднощів, критичним викликом є забезпечення довіри до отриманих результатів. Традиційні методи верифікації та валідації моделей надійності зазвичай обмежуються перевіркою коректності реалізації алгоритмів і порівнянням вихідних даних із простими тестовими сценаріями. Однак такі підходи часто не охоплюють усіх значущих станів складної динамічної системи, внаслідок чого критичні сценарії можуть залишатися непоміченими. Сучасні складні системи (зокрема, ІСД) вимагають удосконалених схем V&V, зокрема перехресної верифікації – перевірки результатів за допомогою двох незалежних методологій моделювання. Такий підхід дозволяє виявити розбіжності (пропущені стани чи переходи) та підтвердити коректність моделі [58], але все ще може бути недостатнім у контексті моделей з динамічним донаванням.

Таким чином, існуючі інструментальні засоби для деяких випадків моделювання жорстких систем для критичного обладнання можуть бути недостатніми, має бути певна методологія правильного обґрунтування та вибору інструментів. Крім того, переважна більшість існуючих методів верифікації та валідації орієнтована на перевірку реалізації моделі, а не на гарантування повноти простору станів та відсутності пропущених переходів у складних динамічних структурах із можливістю структурної еволюції. І також не адаптована під задачі донавчання (динамічної зміни одного з параметрів). Це обґрунтовує необхідність:

1. у створенні спеціалізованої інформаційної технології та розроблення власного програмного забезпечення, яке підтримуватиме симулювання МФММ та базуватиметься на толерантних до жорсткості методах розв'язання;

2. удосконалення методів верифікації та валідації шляхом інтеграції багатоступеневої перехресної перевірки (зокрема із залученням апарату структурно-автоматних моделей), що дозволить забезпечити високу достовірність результатів моделювання.

1.3.3 Обґрунтування та вибір показників оцінювання надійності промислового обладнання з використанням індустріального інтернету речей

Оскільки промислове обладнання є об'єктом тривалого багаторазового використання, що підлягає відновленню та технічному обслуговуванню, його надійність не може оцінюватися лише ймовірністю безвідмовності $P(t)$. Відповідно до ДСТУ 2860-94 [19] та загальної теорії надійності, базовим комплексним показником, що поєднує властивості безвідмовності та ремонтпридатності, є Коефіцієнт готовності (Availability, K_g або $A(t)$).

Функція готовності $A(t)$ визначає ймовірність того, що система буде здатна виконувати необхідну функцію у заданий момент часу t . Однак, в умовах впровадження концепції донавчання ІСД (через виникнення прихованих відмов) функція готовності набуває складної нелінійної форми, перенавчання моделі та застосування результатів навчання може вимагати простою, але й підвищити достовірність діагностування у перспективі. Тому для повноцінного аналізу розроблюваних моделей доцільно ввести три специфічні показники оцінювання:

1. A_{min} – мінімальне (найгірше) значення готовності, що виникає в момент на початковому етапі донавчання;

2. T_{loss} – час, протягом якого система функціонує зі зниженим рівнем готовності до моменту її перетину з базовою кривою (готовність системи діагностування без донавчання);

3. Asteady – стаціонарна готовність, на яке виходить система після завершення процесу донавчання.

Використання цих показників дозволить кількісно обґрунтувати доцільність впровадження ІСД.

1.3.4 Формулювання науково-прикладного протиріччя

На основі проведеного в розділі аналізу методів, моделей та інструментальних засобів оцінювання надійності встановлено, що подальший розвиток систем діагностування гальмується наявністю конфлікту між потребами сучасної індустрії та наявними методологічними обмеженнями.

Протиріччя полягає між об'єктивною необхідністю забезпечення високого рівня готовності та безпечності промислового обладнання систем індустріального інтернету речей, параметри якого динамічно змінюються під впливом деградації та умов експлуатації, та обмеженими можливостями існуючих систем діагностування, які базуються на статичних моделях без використання технології цифрових двійників, механізмів оперативного донавчання і врахування досвіду аналогічних систем, що призводить до накопичення прихованих відмов та зниження точності оцінювання готовності.

Наявність сформульованого протиріччя ускладнює обґрунтоване проектування адаптивних систем діагностування промислового обладнання з використанням ІоТ і визначає необхідність розв'язання актуальної науково-прикладної задачі дисертаційної роботи – розроблення моделей, методу та засобів оцінювання і забезпечення надійності промислового обладнання систем індустріального інтернету речей на основі інтелектуальних систем діагностування з використанням цифрових двійників та режимів донавчання.

1.4 Обґрунтування задач та методики дослідження

1.4.1 Постановка загальної та часткових завдань дослідження

Загальне завдання: розроблення моделей, методу та засобів оцінювання і забезпечення надійності промислового обладнання систем індустриального інтернету речей на основі інтелектуальних систем діагностування з використанням цифрових двійників та режимів донавчання.

Часткові завдання:

1) проаналізувати методи і технології забезпечення та оцінювання готовності промислового обладнання систем індустриального інтернету речей, виявити обмеження існуючих систем діагностування щодо врахування прихованих відмов та обґрунтувати мету, задачі і методику досліджень;

2) розробити теоретико-множинну модель та архітектуру інтелектуальної системи діагностування промислового обладнання з використанням індустриального інтернету речей на основі цифрового двійника з функціями оперативного донавчання та коригування тестових послідовностей;

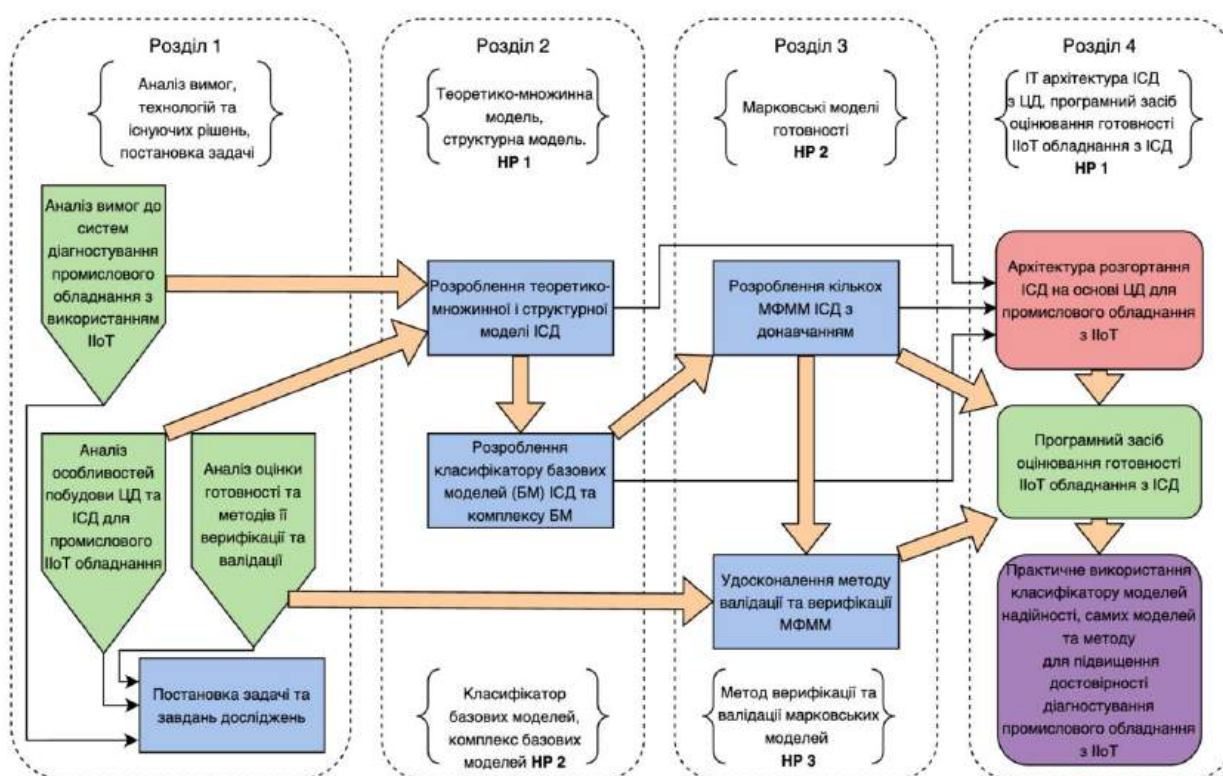
3) розробити онтологічний класифікатор та комплекс базових і мультифрагментних марковських моделей готовності промислового обладнання, що враховують динаміку зміни достовірності діагностування при локальному та федеративному донавчанні цифрового двійника, з подальшим багатокритеріальним ранжуванням базових моделей;

4) удосконалити метод верифікації та валідації мультифрагментних марковських моделей інтелектуальних систем діагностування шляхом використання багатоступеневої перехресної перевірки та врахуванні точності інструментів і математичної «жорсткості» матриці інтенсивностей переходів;

5) розробити еталонну ІТ-архітектуру розгортання цифрового двійника інтелектуальної системи діагностування в середовищі індустриального інтернету речей, а також програмний засіб підтримки прийняття рішень для оцінювання готовності обладнання, та впровадити отримані результати в індустриальні проекти і навчальний процес.

1.4.2 Етапи та методика дослідження

Під час дослідження буде проведено комплекс робіт для формування заявлених наукових результатів. На підставі вхідних даних буде проаналізовано, розроблено та досліджено компоненти моделі, методи і засоби, що представлені загальною схемою взаємозв'язку результатів дослідження на рисунку 1.4.



Рисунки 1.4 – Схема взаємозв'язку результатів дослідження

Розглянуті у першому розділі вимоги до систем діагностування промислового обладнання з використанням індустріального інтернету речей (ІІоТ) та технології цифрових двійників формують базу для постановки задачі дослідження. На основі цього аналізу, у другому розділі об'єкт дослідження буде представлено у вигляді теоретико-множинної моделі та розробленої архітектури розгортання інтелектуальної системи діагностування (ІСД). Теоретико-множинна модель описуватиме взаємодію об'єкта діагностування, блоку цифрового двійника

зі сховищами, середовищем розгортання, засобами федеративного донавчання та оцінки готовності, функції, що виконуватиме така ІСД. Додатково ІСД буде представлено у вигляді структурної діаграми (архітектури). Це формує базу для отримання першого наукового результату (НР1).

Для подальшого математичного моделювання процесів ІСД розробляється онтологічний класифікатор базових моделей. Він дозволяє декомпонувати множину станів та ознак системи залежно від внутрішньофрагментних (тип контролю, врахування обслуговування, кібератак і т.д.) та мультифрагментних властивостей (зміна параметрів достовірність діагностування, цільове донавчання після різних типів відмов, тощо). На основі класифікатора синтезується комплекс базових моделей (БМ) надійності.

Згідно з розробленими БМ, у третьому розділі формується множина мультифрагментних марковських моделей (МФММ). Перехід від БМ до МФММ дозволяє врахувати динаміку зміни достовірності діагностування, зумовлену локальним або федеративним донавчанням цифрового двійника. Сукупність розробленого класифікатора та комплексу МФММ відповідає другому науковому результату дослідження (НР2).

З огляду на структурну складність МФММ та ефект математичної «жорсткості» систем диференціальних рівнянь особливо для гарантоздатного обладнання (розрив між швидким навчанням і рідкісними відмовами), паралельно розробляється та удосконалюється метод верифікації і валідації (V&V). Цей метод базується на багатоступеневій перехресній перевірці результатів за допомогою двох незалежних методологій: на основі графів станів та структурно-автоматних моделей. А також метод включає алгоритм вибору інструменту для підрахунку готовності з врахуванням жорсткості матриці інтенсивностей переходів. Це зменшує обчислювальні похибки та формує третій науковий результат (НР3).

У четвертому розділі теоретичні положення інтегруються у конкретні практичні рішення. На базі розроблених моделей (НР2) та методу їх перевірки (НР3) формалізується еталонна ІТ-архітектура розгортання ІСД. На основі логіки взаємодії класифікатора та розрахункового ядра окремо створено

інструментальний програмний засіб підтримки прийняття рішень «R-SIM». Він призначений для автоматизованого оцінювання готовності промислового обладнання ІоТ з донавчанням.

1.5 Висновки до першого розділу

На підставі проведеного аналізу методів і технологій оцінювання та забезпечення надійності промислового обладнання в середовищі індустриального інтернету речей (ІоТ) сформульовано наступні висновки.

1. За результатами аналізу особливостей функціонування промислового обладнання в середовищі ІоТ встановлено, що забезпечення необхідного рівня готовності складних і критичних систем ускладнюється неминучими процесами фізичного старіння матеріалів та зміною умов експлуатації. Виявлено, що використання традиційних систем з жорстко заданими параметрами діагностування призводить до накопичення прихованих відмов. Це обґрунтовує необхідність переходу до адаптивних рішень та впровадження інтелектуальних систем діагностування (ІСД) з донавчанням

2. Дослідження методів забезпечення надійності в умовах прихованих відмов показало, що подолання проблеми дрейфу концепту та старіння моделей штучного інтелекту вимагає реалізації парадигми безперервного локального та федеративного донавчання. Виявлено, що ефективним та безпечним середовищем для такого донавчання є технологія цифрових двійників.

3. Здійснено аналіз існуючих методів, математичних моделей і програмних засобів оцінювання надійності. Встановлено, що класичні статичні моделі (дерева відмов, структурні схеми надійності) та комерційні програмні комплекси мають обмеження чи не підходять для моделювання динаміки донавчання систем. Це зумовлено неможливістю підтримки змінності параметрів та математичною «жорсткістю» процесів. Для вирішення цієї проблеми обґрунтовано необхідність використання мультифрагментних марковських

моделей (МФММ), розроблення методу їх перехресної верифікації та застосування спеціалізованих показників оцінювання.

4. На підставі проведеного аналізу сформульовано науково-прикладне протиріччя між зростаючими вимогами до готовності та гарантоздатності промислового обладнання систем ІоТ в умовах прихованих відмов і динамічних змін середовища функціонування, з одного боку, та обмеженими можливостями існуючих методів і моделей діагностування та оцінювання готовності, які не враховують адаптивне донавчання цифрових двійників і зміну достовірності діагностики, з іншого боку.

5. Виходячи з виявленого протиріччя, обґрунтовано мету дисертаційного дослідження та сформульовано загальне й часткові завдання, що передбачають розроблення архітектури інтелектуальної системи діагностування на основі цифрового двійника з режимами донавчання, створення онтологічного класифікатора та комплексу базових і мультифрагментних моделей готовності з їх багатокритеріальним ранжуванням, а також удосконалення методу їх верифікації та валідації.

Література до першого розділу

1. Грищук А. М. Необхідність зміни технологічного укладу на основі драйверів економічного зростання. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Економіка і управління*. 2020. Т. 31 (70), № 2 (1). С. 81–88. URL: [http://nbuv.gov.ua/UJRN/UZTNU_econ_2020_31\(70\)_2\(1\)__16](http://nbuv.gov.ua/UJRN/UZTNU_econ_2020_31(70)_2(1)__16) (дата звернення: 01.02.2026).

2. Bigliardi B., Bottani E., Casella G. Enabling technologies, application areas and impact of industry 4.0: a bibliographic analysis. *Procedia Manufacturing*. 2020. Vol. 42. P. 322–326. DOI: 10.1016/j.promfg.2020.02.086.

3. Islam M. T., Sepanloo K., Woo S., Woo S. H., Son Y.-J. A Review of the Industry 4.0 to 5.0 Transition: Exploring the Intersection, Challenges, and Opportunities

of Technology and Human–Machine Collaboration. *Machines*. 2025. Vol. 13, no. 4, article no. 267. DOI: 10.3390/machines13040267.

4. Alabadi M., Habbal A., Wei X. Industrial Internet of Things: Requirements, Architecture, Challenges, and Future Research Directions. *IEEE Access*. 2022. Vol. 10. P. 66374–66400. DOI: 10.1109/ACCESS.2022.3185049.

5. Boyes H., Hallaq B., Cunningham J., Watson T. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry*. 2018. Vol. 101. P. 1–12. DOI: 10.1016/j.compind.2018.04.015.

6. Hermeto R. T., Gallais A., Theoleyre F. Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey. *Computer Communications*. 2017. Vol. 114. P. 84–105. DOI: 10.1016/j.comcom.2017.10.004.

7. What is IIoT and which industries does it impact? *CCLAB*. URL: <https://www.cclab.com/news/what-is-iiot-and-which-industries-does-it-impact> (дата звернення: 01.02.2026).

8. Afrin S., Rafa S., Kabir M., Farah T., Alam M. S., Lameesa A., Ahmed S., Gandomi A. Industrial Internet of Things: Implementations, challenges, and potential solutions across various industries. *Computers in Industry*. 2025. Vol. 170, article no. 104317. DOI: 10.1016/j.compind.2025.104317.

9. Jarwar M. A., Watson J., Ali S. Modeling Industrial IoT Security Using Ontologies: A Systematic Review. *IEEE Open Journal of the Communications Society*. 2025. DOI: 10.1109/OJCOMS.2025.3532224.

10. Endres H., Indulska M., Ghosh A., Baiyere A., Broser S. Industrial Internet of Things (IIoT) *Business Model Classification*. *40th International Conference on Information Systems (ICIS)* : Proc. Int. Conf., Munich, Germany, 2019. URL: https://www.researchgate.net/publication/342261722_Industrial_Internet_of_Things_IIoT_Business_Model_Classification (дата звернення: 01.02.2026).

11. Lin S.-W., Mellor S., Munz H., Barnstedt E. Architecture Alignment and Interoperability. An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper. URL:

https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf (дата звернення: 01.02.2026).

12. ДСТУ ISO/IEC 30141:2019. Інтернет речей. Еталонна архітектура (ISO/IEC 30141:2018, IDT). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=83590 (дата звернення: 01.02.2026).

13. Bochra G., Abderrahim S. The Importance of Semantic Interoperability in The Internet of Things (IOT). *SWIoT+MSW@KGSWC* : Proc. Conf., 2023. URL: <https://api.semanticscholar.org/CorpusID:269043727> (дата звернення: 01.02.2026).

14. Moreira J. L., Daniele L. M., Ferreira Pires L., van Sinderen M. J., Wasielewska K., Szmeja P., Pawlowski W., Ganzha M., Paprzycki M. Towards IoT platforms' integration: Semantic Translations between W3C SSN and ETSI SAREF. *SEMANTiCS conference 2017* : Proc. Int. Conf., Amsterdam, Netherlands, 2017. URL: <http://ceur-ws.org/Vol-2063/sisiot-paper3.pdf> (дата звернення: 01.02.2026).

15. Cornejo Lupa M., Aguilera A., Dongo I. Developing an Ontology to Enhance Semantic Interoperability in IoT Environment. *CLEI Electronic Journal*. 2025. Vol. 28. DOI: 10.19153/cleiej.28.3.4.

16. Stryzhak O., Horborukov V., Prychodniuk V., Franchuk O., Chepkov R. Decision-making System Based on The Ontology of The Choice Problem. *Journal of Physics: Conference Series*. 2021. Vol. 1828, no. 1. P. 012007. DOI: 10.1088/1742-6596/1828/1/012007.

17. Mrabet H., Belguith S., Alhomoud A., Jemai A. A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. *Sensors*. 2020. Vol. 20, no. 13, article no. 3625. DOI: 10.3390/s20133625.

18. Gautam V. L., Lanjewar U. A. Cloud, Fog, and Edge Computing: A Comparative Analysis of Architectures, Applications, Security Challenges and Performance. *International Journal of Computer Science Trends and Technology (IJCST)*. 2025. P. 130–137. URL: <https://www.ijcstjournal.org/conference-issue/ICAMT2025/23.pdf> (дата звернення: 01.02.2026).

19. ДСТУ 2860-94. Надійність техніки. Терміни та визначення. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=25034 (дата звернення: 01.02.2026).

20. ДСТУ EN 61508-1:2019. Функційна безпечність електричних, електронних, програмованих електронних систем, пов'язаних із безпекою. Частина 1. Загальні вимоги (EN 61508-1:2010, IDT; IEC 61508-1:2010, IDT). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=84383 (дата звернення: 01.02.2026).

21. Owen B., Smith J. Signal Integrity Challenges in Harsh Industrial Environments Under Cyber Interference. 2025. URL: https://www.researchgate.net/publication/398394492_Signal_Integrity_Challenges_in_Harsh_Industrial_Environments_Under_Cyber_Interference (дата звернення: 01.02.2026).

22. Securing IIoT Systems: An Analysis of Key Vulnerabilities and Threats in Manufacturing Industries. CCLAB. URL: <https://www.cclab.com/news/securing-iiot-systems-an-analysis-of-key-vulnerabilities-and-threats-in-manufacturing-industries> (дата звернення: 01.02.2026).

23. Kohn L. T., Corrigan J. M., Donaldson M. S. Why Do Errors Happen? *Institute of Medicine (US) Committee on Quality of Health Care in America*. 2000. URL: <https://www.ncbi.nlm.nih.gov/books/NBK225171/> (дата звернення: 01.02.2026).

24. Industrial IoT Enhancing Equipment Reliability and Uptime. 2025. URL: <https://oxmaint.com/blog/post/iiot-equipment-reliability-uptime> (дата звернення: 02.02.2026).

25. Benhanifia A., Cheikh Z. B., Oliveira P. M., Valente A., Lima J. Systematic review of predictive maintenance practices in the manufacturing sector. *Intelligent Systems with Applications*. 2025. Vol. 26, article no. 200501. DOI: 10.1016/j.iswa.2025.200501.

26. ДСТУ 2389-94. Технічне діагностування та контроль технічного стану. Терміни та визначення. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=104134 (дата звернення: 02.02.2026).
27. Gonzalez-Jimenez D., del-Olmo J., Poza J., Garramiola F., Madina P. Data-Driven Fault Diagnosis for Electric Drives: A Review. *Sensors*. 2021. Vol. 21, no. 12, article no. 4024. DOI: 10.3390/s21124024.
28. Yan W., Wang J., Lu S., Zhou M., Peng X. A Review of Real-Time Fault Diagnosis Methods for Industrial Smart Manufacturing. *Processes*. 2023. Vol. 11, no. 2, article no. 369. DOI: 10.3390/pr11020369.
29. Leite D., Andrade E., Rativa D., Maciel A. M. A. Fault Detection and Diagnosis in Industry 4.0: A Review on Challenges and Opportunities. *Sensors*. 2025. Vol. 25, no. 1, article no. 60. DOI: 10.3390/s25010060.
30. Rodriguez-Aguilar R., Marmolejo-Saucedo J.-A., Köse U. Development of a Digital Twin Driven by a Deep Learning Model for Fault Diagnosis of Electro-Hydrostatic Actuators. *Mathematics*. 2024. Vol. 12, no. 19, article no. 3124. DOI: 10.3390/math12193124.
31. Ademujimi T., Prabhu V. Model-Driven Bayesian Network Learning for Factory-Level Fault Diagnostics and Resilience. *Sustainability*. 2024. Vol. 16, no. 2, article no. 513. DOI: 10.3390/su16020513.
32. Kumar S., Raj K. K., Cirrincione M., Cirrincione G., Franzitta V., Kumar R. R. A Comprehensive Review of Remaining Useful Life Estimation Approaches for Rotating Machinery. *Energies*. 2024. Vol. 17, no. 22, article no. 5538. DOI: 10.3390/en17225538.
33. Carino J. A. та ін. Fault Detection and Identification Methodology Under an Incremental Learning Framework Applied to Industrial Machinery. *IEEE Access*. 2018. Vol. 6. P. 49755–49766. DOI: 10.1109/ACCESS.2018.2868430.
34. Chen B., Shen C., Shi J., Kong L., Tan L., Wang D., Zhu Z. Continual learning fault diagnosis: A dual-branch adaptive aggregation residual network for fault diagnosis with machine increments. *Chinese Journal of Aeronautics*. 2023. Vol. 36, no. 6. P. 361–377. DOI: 10.1016/j.cja.2022.08.019.

35. Chen X., Gao C., Delbruck T., Liu S.-C. EILE: Efficient Incremental Learning on the Edge. *Artificial Intelligence Circuits and Systems (AICAS)* : Proc. 3rd IEEE Int. Conf., Washington DC, USA, 2021. P. 1–4. DOI: 10.1109/AICAS51828.2021.9458554.
36. Neupane D., Bouadjenek M. R., Dazeley R., Aryal S. Data-driven machinery fault diagnosis: A comprehensive review. *Neurocomputing*. 2025. Vol. 627, article no. 129588. DOI: 10.1016/j.neucom.2025.129588.
37. Savaglio C., Barbuto V., Mangione F., Fortino G. Generative Digital Twins: A Novel Approach in the IoT Edge-Cloud Continuum. *IEEE Internet of Things Magazine*. 2025. Vol. 8, no. 1. P. 42–48. DOI: 10.1109/IOTM.001.2400035.
38. Щеглов В. Р., Морозова О. І. Методи та технології розроблення цифрових двійників для гарантоздатних систем індустриального інтернету речей. *Системи управління, навігації та зв'язку*. 2022. Вип. 4 (70). С. 127–137. DOI: 10.26906/SUNZ.2022.4.127.
39. Щеглов В. Р., Морозова О. І. Застосування технології цифрових двійників відносно безпілотних літальних апаратів. *Авіація, промисловість, суспільство* : матеріали наук.-практ. конф., м. Харків, 2024. Харків, 2024. С. 47–51.
40. Гаєвський В. В., Кунуп Т. В., Морозова О. І., Щеглов В. Р. Розділ 11. Інтелектуальні системи інтернету речей з використанням технологій цифрових двійників. *Methods and technologies of ensuring quality and safety of intelligent systems* : кол. монографія / за заг. ред. В. С. Харченка, О. І. Морозової. Харків : Міністерство освіти і науки України, Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ», 2023. С. 193–224.
41. Bisanti G., Mainetti L., Montanaro T., Patrono L., Sergi I. Digital twins for aircraft maintenance and operation: A systematic literature review and an IoT-enabled modular architecture. *Internet of Things*. 2023. Vol. 24, article no. 100991. DOI: 10.1016/j.iot.2023.100991.
42. ISO 23247-1:2021(en). *Automation systems and integration – Digital twin framework for manufacturing – Part 1: Overview and general principles*. URL:

<https://www.iso.org/obp/ui/en/#iso:std:iso:23247:-1:ed-1:v1:en> (дата звернення: 02.02.2026).

43. The DTC Glossary. *Digital Twin Consortium*. URL: <https://www.digitaltwinconsortium.org/glossary/glossary/#digital-twin> (дата звернення: 02.02.2026).

44. Kharchenko V., Shcheglov V., Ivasiuk O., Morozova O. Digital Twin-Based Lifecycle Methodology for Ensuring Safety of NPP/SMR I&C Systems. *Technologies*. 2026. Vol. 14, no. 1, article no. 46. P. 1–33. DOI: 10.3390/technologies14010046.

45. Grieves M. Digital Model, Digital Shadow, Digital Twin. 2023. URL: https://www.researchgate.net/publication/369830792_Digital_Model_Digital_Shadow_Digital_Twin (дата звернення: 02.02.2026).

46. Grieves M. Origins of the Digital Twin Concept. 2016. DOI: 10.13140/RG.2.2.26367.61609.

47. Singh M., Fuenmayor E., Hinchy E. P., Qiao Y., Murray N., Devine D. Digital Twin: Origin to Future. *Applied System Innovation*. 2021. Vol. 4, no. 2, article no. 36. DOI: 10.3390/asi4020036.

48. Schuh G., Anderl R., Dumitrescu R., Krüger A., ten Hompel M. Industrie 4.0 Maturity Index. Managing the Digital Transformation of Companies – UPDATE 2020. 2020. URL: <https://en.acatech.de/publication/industrie-4-0-maturity-index-update-2020/> (дата звернення: 02.02.2026).

49. Leon-Medina J. X., Tibaduiza D. A., Parés N., Pozo F. Digital twin technology in wind turbine components: A review. *Intelligent Systems with Applications*. 2025. Vol. 26, article no. 200535. DOI: 10.1016/j.iswa.2025.200535.

50. Coelho M., Houck K., Browning L., Sabharwall P., Folmar C., Cavaluzzi J., McClure P., Dunker J. Early Fault Detection in Nuclear Systems: A Digital Engineering Approach. *IEEE Open Journal of Systems Engineering*. 2025. P. 1–12. DOI: 10.1109/OJSE.2025.3562518.

51. Sadeghi A., Bellavista P., Song W., Yazdani-Asrami M. Digital Twins for Condition and Fleet Monitoring of Aircraft: Toward More-Intelligent Electrified

Aviation Systems. *IEEE Access*. 2024. Vol. 12. P. 99806–99832. DOI: 10.1109/ACCESS.2024.3371902.

52. Wang Q., Zhang X., Lu J., Xiao G., Ren Y., Li W. Digital Twin-Driven Physically Constrained Generative Adversarial Network for Industrial Boiler Fault Diagnosis. *IEEE Transactions on Instrumentation and Measurement*. 2025. Vol. 74, article no. 3530215. DOI: 10.1109/TIM.2025.3558806.

53. Shcheglov V., Kharchenko V., Volochiy B., Morozova O., Zaitseva E. Availability Assessing Intelligent Diagnostic System with Relearning after Complex Equipment Failures: Structural-Automata Model-based Methodology of Markov's Analysis. *Dependable Systems, Services and Technologies (DESSERT'2024)* : Proc. 14th IEEE Int. Conf., Athens, Greece, 2024. P. 1–8. DOI: 10.1109/DESSERT65323.2024.11122169.

54. Kabir S. An overview of fault tree analysis and its application in model based dependability analysis. *Expert Systems with Applications*. 2017. Vol. 77. P. 114–135. DOI: 10.1016/j.eswa.2017.01.058.

55. Keefe K., Sanders W. Reliability Analysis with Dynamic Reliability Block Diagrams in the Möbius Modeling Tool. *EAI Endorsed Transactions on Security and Safety*. 2016. Vol. 3. DOI: 10.4108/eai.14-12-2015.2262723.

56. Kharchenko V., Odarushchenko O., Butenko V., Popov P. Selecting mathematical software for dependability assessment of computer systems described by stiff Markov chains. *CEUR Workshop Proceedings*. 2013. Vol. 1000. P. 146–162. URL: https://www.researchgate.net/publication/289567556_Selecting_mathematical_software_for_dependability_assessment_of_computer_systems_described_by_stiff_Markov_chains (дата звернення: 02.02.2026).

57. Malhotra M., Muppala J. K., Trivedi K. S. Stiffness-tolerant methods for transient analysis of stiff Markov chains. *Microelectronics Reliability*. 1994. Vol. 34, no. 11. P. 1825–1841. DOI: 10.1016/0026-2714(94)90137-6.

58. Fonseca i Casas P., Ruiz-Martin C. Building Models in Pairs for Cross-Verification Using SDL and DEVS. *Advanced Theory and Simulations*. 2024. Vol. 7. DOI: 10.1002/adts.202300839.

РОЗДІЛ 2. РОЗРОБЛЕННЯ АРХІТЕКТУРИ ТА МАТЕМАТИЧНИХ МОДЕЛЕЙ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ДІАГНОСТУВАННЯ ПРОМИСЛОВОГО ОБЛАДНАННЯ З ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ І ЦИФРОВИХ ДВІЙНИКІВ ІЗ ДОНАВЧАННЯМ

2.1 Розроблення теоретико-множинної моделі та архітектури ІСД промислового обладнання з використанням індустріального інтернету речей із донавчанням на основі цифрового двійника

Як було показано у результатах аналізу першого розділу, забезпечення високого рівня готовності промислового обладнання з використанням ІоТ неможливе без своєчасного виявлення неспецифікованих відмов та роботою над зменшенням кількості прихованих відмов. Виявлене науково-прикладне протиріччя вимагає переходу від класичних статичних засобів контролю до інтелектуальних систем діагностування із донавчанням та залученням технології цифрових двійників.

Для системного вирішення цієї задачі у даному підрозділі пропонується розробити високорівневу архітектуру та метод функціонування ІСД. Запропоноване рішення має об'єднати на рівні граничних обчислень традиційні засоби оперативного і тестового діагностування з віртуальним середовищем ЦД, яке виступатиме безпечним полігоном для симуляції відмов, локального оновлення вагових коефіцієнтів та формування даних для глобального федеративного навчання. Основні концептуальні положення та принципи побудови запропонованої архітектури ІСД промислового обладнання в середовищі індустріального інтернету речей були розроблені автором і висвітлені у наукових працях [1-3], колективній монографії [4], а застосування запропонованих принципів розподілу обчислювальних ресурсів між Edge та Cloud рівнями для функціонування цифрового двійника і можливість їх адаптації для мобільних об'єктів (зокрема БПЛА) було апробовано у роботі [5]. Для математичного обґрунтування взаємодії цих компонентів побудову архітектури доцільно розпочати з формалізації теоретико-множинної моделі системи.

2.1.1 Теоретико-множинна модель ІСД промислового обладнання з використанням індустріального інтернету речей із донавчанням на основі цифрового двійника

Опис інтелектуальної системи діагностування (ІСД) промислового обладнання з використанням індустріального інтернету речей із донавчанням на основі цифрового двійника за допомогою апарату теорії множин забезпечує системний підхід до вивчення та аналізу складної кіберфізичної структури такого рішення. У науковій літературі існують приклади того, як формально-теоретико-множинні підходи застосовуються для моделювання складних систем [6, 7]. Теоретико-множинний опис дозволяє виявити та строго формалізувати взаємозв'язки між різними елементами, компонентами інфраструктури та процесами адаптації [6]. Теоретико-множинна модель, в якій використовується математичний апарат реляційної алгебри, дозволяє формалізувати інтелектуальну систему діагностування (ІСД) як сукупність скінченних множин компонентів, станів і функцій, та чітко задати правила їх взаємодії (відображення), що створює строгий математичний базис для подальшого синтезу архітектури та побудови марковських графів.

У наведеному теоретичному описі, відповідно до запропонованого способу функціонування, загальна модель ІСД промислового обладнання з використанням індустріального інтернету речей на основі цифрового двійника представлена у вигляді кортежу шести взаємопов'язаних базових множин:

$$IDS = \langle M, S, D, DT, L, E \rangle, \quad (2.1)$$

де M – множина режимів роботи системи; S – множина станів фізичного об'єкта діагностування; D – множина апаратних та алгоритмічних засобів, що забезпечують безпосереднє діагностування; DT – множина компонентів цифрового двійника (Digital Twin); L – множина алгоритмів і процесів донавчання

(Learning); E – множина засобів оцінювання та розрахунку функції готовності (Evaluation). Розглянемо детально кожен зі складових кортежу (2.1) та їхні взаємозв'язки.

1. Модель режимів роботи системи (M). Згідно з архітектурою розробленого комплексу, поведінка ІСД детермінується шістьма експлуатаційними режимами:

$$M = \{mset, mpdiag, mwork, mdiag, mmaint, mrep\}, \quad (2.2)$$

де *mset* – режим налаштування (запускається при старті системи або після відновлення, активує процедури донавчання *L*); *mpdiag* – режим профілактичного діагностування (подача профілактичних тестових послідовностей, що вимагає тимчасового переривання виконання об'єктом робочих функцій); *mwork* – робочий режим (штатна експлуатація з безперервним оперативним контролем); *mdiag* – режим діагностування (активується при виявленні відмови контролем в *mwork* для більш глибокого діагностування); *mmaint* – режим планового обслуговування (зупинка за графіком); *mrep* – режим відновлення (фізичний ремонт об'єкта після підтвердження відмови).

2. Модель станів об'єкта діагностування (S). Фізичний об'єкт характеризується множиною істинних технічних станів *S*, які прямо співвідносяться режимами роботи *M*:

$$S = Su \cup Sd \cup Sh \cup Sm \cup Sl, \quad (2.3)$$

де *Su* – множина працездатних станів (відповідає виключно режиму *mwork*, коли система виконує цільову функцію); *Sd* – множина станів явної відмови (ініціює перехід до *mrep*); *Sh* – множина станів прихованої відмови (система перебуває в *mwork*, але функціонує з порушеннями); *Sm* – стан тестування та обслуговування (відповідає режимам *mpdiag* та *mmaint*, оскільки в обох випадках об'єкт перериває роботу для перевірки або регламентних робіт, під час яких можуть бути виявлені

приховані відмови); Sl – стан донавчання (відповідає $mset$). Примітка: окремого стану ремонту у множині не виділяється, оскільки фізичне відновлення ($mrep$) є перехідним процесом (із заданою інтенсивністю μ) зі станів Sd або Sm до станів Su чи Sl .

3. Модель засобів діагностування граничного рівня (D). Множина засобів безпосереднього діагностування формалізує апаратну частину системи:

$$D = \langle Mx, Ctrlsys, Td, Oc \rangle, \quad (2.4)$$

де Mx – мультиплексор вхідних даних, що перемикає потоки інформації залежно від режиму; $Ctrlsys$ – загальний блок керування системою діагностування, що здійснює диспетчеризацію режимів із множини M ; Td – блок тестового діагностування (генерує тестові послідовності); Oc – блок оперативного контролю (працює паралельно з об'єктом у режимі $mwork$).

4. Модель цифрового двійника (DT). Оскільки реалізація ІСД вимагає як автономного прийняття рішень на місці, так і глобального обміну знаннями, множина ЦД декомпозується на граничну ($Edge / DTI$) та хмарну ($Cloud / DTA$) складові:

$$DT = DT_{edge} \cup DT_{cloud}. \quad (2.5)$$

Локальний екземпляр DT_{edge} функціонує в контурі ІСД на рівні обладнання:

$$DT_{edge} = \langle Rloc, Mvirt, An, PFloc \rangle, \quad (2.6)$$

де $Rloc$ (Repository) – репозиторій локальних історичних даних телеметрії; $Mvirt$ (Virtual Models) – множина віртуальних моделей об'єкта (фундаментальні фізичні, математичні, CAD-моделі та моделі на основі експертних правил). Вона формує базис для реалізації віртуальних сенсорів (Virtual Sensing) безпосередньо на

пристрої, надаючи фізичні закономірності для розрахунку непрямих параметрів; An (Analysis) – блок аналізу результатів оперативного та тестового контролю; PFloc (Parameter Formation) – локальний блок формування параметрів, що виступає адаптивним обчислювальним рушієм. Згідно з архітектурою системи, цей блок містить локальну нейромережеву модель. Використовуючи поточні дані з Rloc та фізичні обмеження з Mvirt (реалізуючи парадигму фізично-інформованого машинного навчання, PIML), нейромодель у PFloc здійснює ідентифікацію нових аномалій, швидке локальне донавчання та генерує оновлені діапазони для засобів контролю. Таким чином, на відміну від звичайної data-driven нейромоделі, подібна інтеграція перетворює цифровий двійник на засіб функціональної інтерпретації об'єкта, що дозволяє виявляти нові типи відмов навіть за відсутності їхніх прикладів у навчальній вибірці.

Хмарний агрегатор DTcloud (DTA) розгорнутий на серверному бекенді і сфокусований на завданнях глобальної агрегації та федеративного навчання:

$$Dcloud = \langle Rglob, PFglob \rangle, \quad (2.7)$$

де Rglob – глобальний репозиторій діагностичних повідомлень та агрегованих даних усього парку обладнання; PFglob – глобальний блок формування параметрів (засіб федеративного навчання), що відповідає за прийом різниць (дельт) вагових коефіцієнтів від усіх DTedge, агрегацію досвіду, формування єдиної оновленої загальної моделі та її подальше розповсюдження на локальні пристрої.

5. Модель донавчання (L). Множина алгоритмів адаптації визначає відображення ідентифікованих ознак прихованих відмов у нові знання під час режиму mset:

$$L(X(Sh), DT) \rightarrow \Delta W \cup \Delta P \cup \Delta Tseq, \quad (2.8)$$

де $X(Sh)$ – множина накопичених векторів даних (ознак та телеметрії), що характеризують прояви прихованої відмови (Sh), збережених у репозиторії ЦД; DT – поточний базис знань ЦД (його існуючі математичні та нейромережеві моделі), стан якого використовується алгоритмом спільно з новими даними $X(Sh)$ для розрахунку необхідних оновлень; ΔW – різниця (дельта) вагових коефіцієнтів локальної моделі (відправляється до $DTcloud$ для федеративного навчання); ΔP – оновлені параметри оперативного контролю (динамічна зміна меж допускового контролю діагностичної моделі для блоку Oc); $\Delta Tseq$ – скориговані тестові послідовності (передаються до блоку Td). Оскільки для різного промислового обладнання у різному середовищі допуски параметрів є індивідуальними, процес донавчання ЦД дозволяє адаптувати цю межу між нормою та відмовою під реальні умови експлуатації.

6. Модель оцінювання готовності та апостеріорних матриць (E). Ця множина формалізує математичний апарат (вбудований у блок розрахунку готовності ІСД) для оцінювання показників надійності в умовах динамічної зміни параметрів:

$$E = \langle Mbase, Mfrag, Sel, Calc \rangle, \quad (2.9)$$

де $Mbase$ – блок зберігання базових матриць марковських моделей; $Mfrag$ – блок зберігання фрагментних матриць; Sel – блок вибору фрагментної матриці; $Calc$ – блок обчислення функції готовності $A(t)$. Додатковою особливістю множини E є реалізація механізму апостеріорних мультифрагментних марковських моделей. Після успішного завершення режиму $mset$ та виконання алгоритмів L , приховані відмови ідентифікуються і переходять до розряду специфікованих ($Sh \rightarrow Sd$), а показник достовірності контролю D зростає. Блок Sel використовує апостеріорну логіку (знання «після дослідження»): він відсікає ті гілки та стани графа $Mbase$, повернення до яких стало неможливим завдяки донавчанню, і формує нову фрагментну матрицю меншої розмірності у $Mfrag$.

Запропонована теоретико-множинна модель формалізує взаємозв'язки між апаратними компонентами рівня Edge, віртуальним середовищем цифрового двійника, режимами функціонування та механізмом апостеріорного розрахунку надійності, а також зв'язків із рівнем Cloud. Сформована модель є математичним базисом для розроблення структурної архітектури ІСД у наступному підрозділі.

2.1.2 Архітектура ІСД промислового обладнання з використанням індустріального інтернету речей із донавчанням на основі цифрового двійника

Розроблена теоретико-множинна модель (2.1)–(2.9) визначає логіку взаємодії компонентів, проте для практичної реалізації системи необхідна її трансформація у структурну апаратно-програмну архітектуру. Враховуючи вимоги до систем індустріального інтернету речей (IIoT), зокрема обмеження обчислювальних ресурсів на місцях експлуатації та необхідність глобальної агрегації досвіду, архітектура ІСД будується за ієрархічним принципом та розподіляється на два базові рівні: рівень граничних обчислень (Edge Level) та хмарний рівень (Cloud Level).

Загальна структурна схема інтелектуальної системи діагностування, що реалізує заявлений спосіб наведена на рисунку 2.1.

Граничний рівень (Edge Level) розгортається безпосередньо на контролері або промисловому шлюзі в контурі об'єкта діагностування (O). Цей рівень містить апаратні засоби множини D та локальний екземпляр цифрового двійника DTedge, що містить наступні компоненти.

1. Мультиплексор вхідних даних (Mx), який приймає сирі дані від об'єкта та розподіляє їх між блоком оперативного контролю (Oc) та блоком тестового діагностування (Td) залежно від поточного режиму роботи (M).
2. Загальний блок керування системою (Ctrlsys), що здійснює диспетчеризацію режимів.
3. Підсистема базового контролю, що складається з блока оперативного контролю та блока тестового діагностування.

4. Блок цифрового двійника об'єкта діагностування (DTedge). Згідно з розробленою теоретичною моделлю, його внутрішня структура (рис. 2.2) складається з чотирьох компонентів: сховища історичних даних (Rloc), сховища віртуальних моделей (Mvirt), блока аналізу результату контролю (An) та блока формування параметрів (PFloc), який містить локальну нейромережеву модель і виступає головним адаптивним рушієм для підвищення достовірності діагностування.

5. Блок розрахунку готовності (E) (рис. 2.3) забезпечує реалізацію апарату апостеріорних мультифрагментних марковських моделей і складається з блоків зберігання базових (Mbase) та фрагментних (Mfrag) матриць, блока вибору фрагментної матриці (Sel) та блока обчислення функції готовності (Calc).

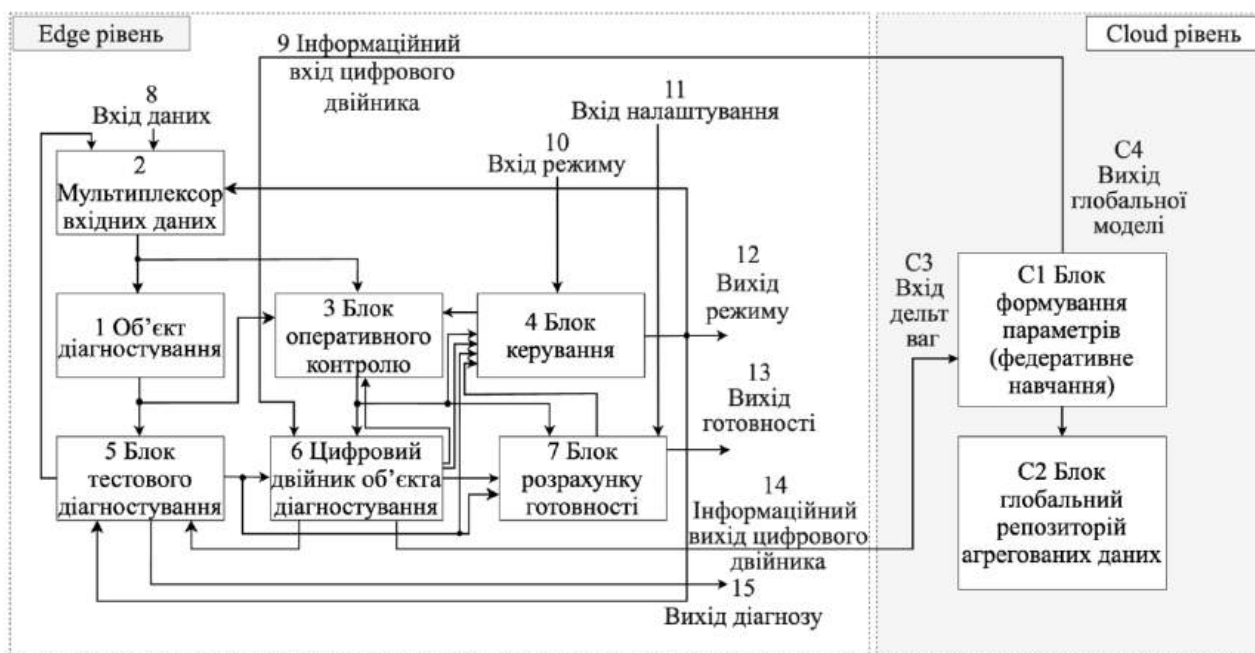


Рисунок 2.1 – Загальна структурна архітектура ІСД на основі цифрового двійника

Хмарний рівень (Cloud Level) відповідає множині DTcloud і розгортається на віддаленому сервері. Він містить глобальний репозиторій агрегованих даних (Rglob) та глобальний блок формування параметрів (PFglob), які забезпечують реалізацію процедур федеративного навчання для парку однотипного обладнання.



Рисунок 2.2 – Структура цифрового двійника об'єкта діагностування (DTedge)



Рисунок 2.3 – Внутрішня структура блока розрахунку готовності (E)

Структурні зв'язки та інтерфейси системи. Оскільки розроблена архітектура базується на поділі обчислень між Edge та Cloud рівнями, взаємодія між її компонентами визначається системою статичних інформаційних зв'язків. Усю сукупність інтерфейсів системи доцільно розділити на чотири логічні групи: зовнішні (міжрівневі), внутрішні рівня cloud, загальні внутрішні рівня edge та інтерфейси зворотного керування рівня edge.

Зв'язок між хмарним (Cloud) та граничним (Edge) рівнями реалізується через спеціалізований двонаправлений мережевий інтерфейс, що забезпечує підтримку процесів федеративного навчання. Цей інтерфейс утворено парою інформаційних каналів (рис. 2.1):

1. каналом передачі локального досвіду – від інформаційного виходу цифрового двійника (14) до входу дельт ваг (C3);
2. каналом оновлення моделі – від виходу глобальної моделі (C4) до інформаційного входу цифрового двійника (9). Такий поділ дозволяє мінімізувати навантаження на мережу, оскільки передаються лише зміни (дельти) вагових коефіцієнтів, а не сирі масиви телеметрії.

Внутрішні зв'язки рівня cloud забезпечують процеси агрегації та збереження глобального досвіду діагностування усього парку обладнання. Головним вузлом виступає глобальний блок формування параметрів (C1), який приймає дані через вхід дельт ваг (C3). Він має безпосередній структурний зв'язок із глобальним репозиторієм агрегованих даних (C2), куди передає для довгострокового архівування накопичені результати федеративного злиття моделей та діагностичні повідомлення. Після обробки масивів даних сформована оновлена загальна модель спрямовується від блоку (C1) до виходу глобальної моделі (C4) для її подальшого розповсюдження на локальні пристрої

Внутрішні зв'язки рівня edge забезпечують прямий потік даних від об'єкта діагностування (1) до засобів діагностування та ЦД. Базовим вузлом маршрутизації виступає мультиплексор вхідних даних (2), який під апаратним керуванням блоку (4) розподіляє потоки телеметрії між блоком оперативного контролю (3) та блоком тестового діагностування (5). Далі результати роботи цих

базових засобів контролю паралельно транслюються на входи цифрового двійника (6) та блоку розрахунку готовності (7).

Інтерфейси зворотного керування (адаптації) рівня edge є ключовою архітектурною особливістю запропонованої ІСД, що фізично реалізує концепцію донавчання. Вони представлені такими зв'язками:

1. Від цифрового двійника (6) до підсистеми базового контролю: наявність інтерфейсів переналаштування забезпечує можливість динамічного оновлення діапазонів допусків для блоку (3) та зміни тестових послідовностей для блоку (5) на основі рішень локальної нейромоделі.

2. Від підсистеми базового контролю та ЦД (6) до блоку розрахунку готовності (7): ці інформаційні зв'язки постачають дані про виявлені приховані відмови до блоку вибору фрагментної матриці (22), що є структурною умовою для виконання відсікання застарілих станів в умовах роботи апостеріорних марковських моделей.

Таким чином, запропонована дворівнева архітектура ІСД практично імплементує розроблену теоретико-множинну модель. Розподіл обчислювальних функцій між граничним (Edge) та хмарним (Cloud) рівнями забезпечує необхідний баланс: локальний цифровий двійник гарантує швидке реагування на приховані відмови в режимі реального часу, а хмарний агрегатор дозволяє масштабувати набуті знання на весь парк обладнання через механізм федеративного навчання. Сформована архітектура є базисом для реалізації динамічного методу технічного діагностування, який розглядається у наступному підрозділі.

2.1.3 Розроблення методу технічного діагностування промислового обладнання з використанням індустріального інтернету речей з оперативним та федеративним донавчанням на основі цифрового двійника

Для забезпечення автономного функціонування запропонованої у попередньому підрозділі апаратно-програмна архітектура в умовах виникнення неспецифікованих (прихованих) відмов необхідно формалізувати динаміку

взаємодії її компонентів у часі. З цією метою розроблено метод технічного діагностування, який визначає алгоритм поведінки системи як скінченного автомата, що здійснює переходи між експлуатаційними режимами.

В основу розробленого методу покладено адаптивний цикл послідовності переходів між шістьма базовими режимами (M): налаштування (mset), профілактичним діагностуванням (mpdiag), робочим режимом (mwork), діагностуванням (mdia), плановим обслуговуванням (mmaint) та відновленням (mrep).

Граф станів (State Machine), що візуалізує динаміку переходів системи розробленим алгоритмом, наведено на рисунку 2.4.

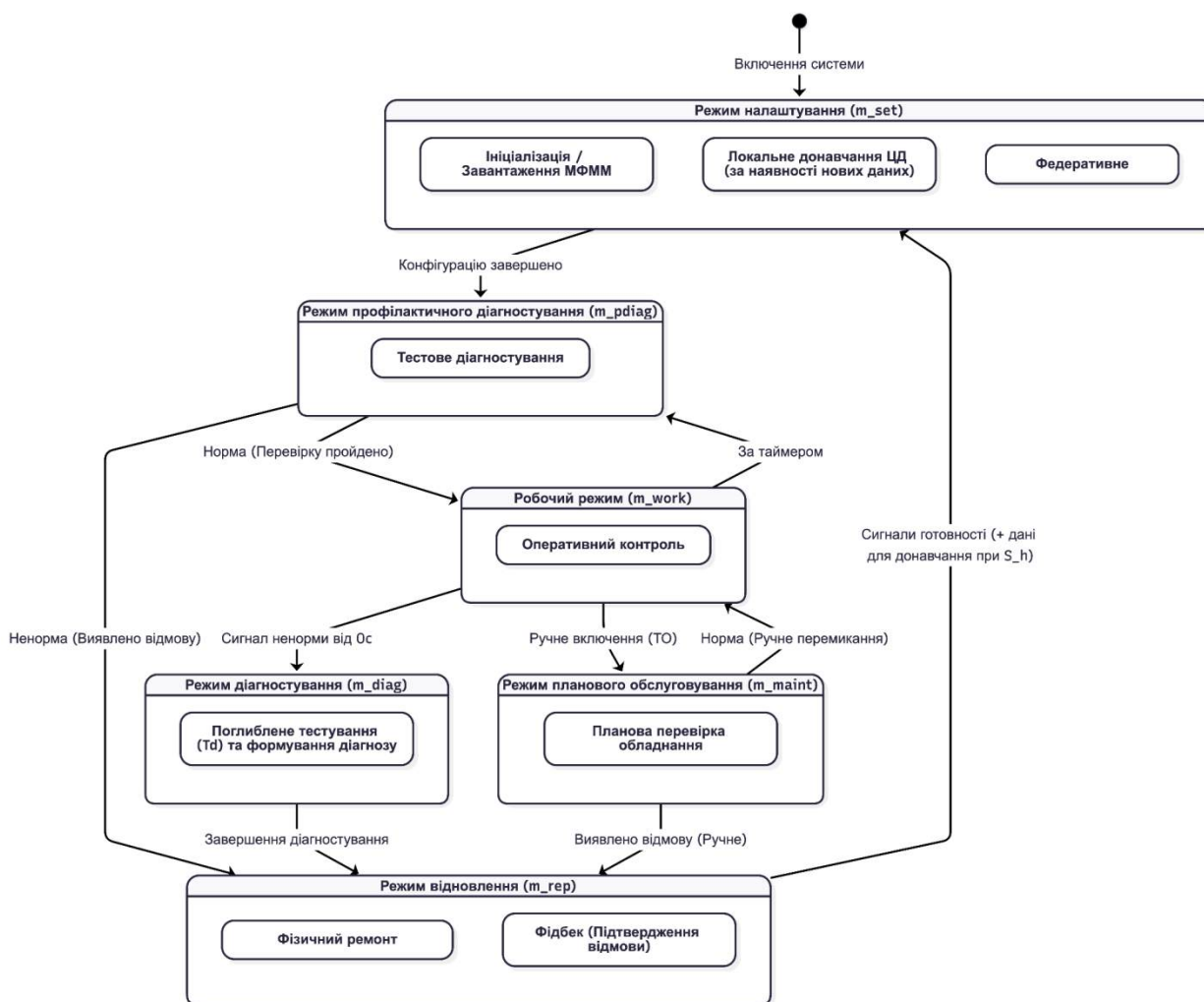


Рисунок 2.4 – Діаграма state machine динаміки переходів між режимами роботи

Реалізація методу здійснюється за наступними 6 етапами, що відповідають режимам роботи системи.

Етап 1. Ініціалізація, донавчання та конфігурація (Режим налаштування $mset$). Режим активується при включенні системи або після завершення ремонту. Блок розрахунку готовності (E) завантажує актуальну конфігурацію базових і фрагментних матриць та обчислює поточний показник готовності $A(t)$. Паралельно локальний екземпляр цифрового двійника (DTedge) виконує ініціалізацію. Якщо з попереднього циклу або з сервера надійшли нові дані про неспецифіковані відмови, нейромодель у блоці формування параметрів (PFloc) виконує локальне донавчання: коригує тестові послідовності, змінює діапазони допусків та відправляє дельти ваг (ΔW) на хмарний сервер для федеративного навчання. Після збереження параметрів формується сигнал завершення конфігурації.

Етап 2. Валідація та профілактика (Режим профілактичного діагностування $mpdiag$). Одразу після завершення налаштування (для гарантування коректності старту), а також періодично за таймером під час роботи, система переходить у цей режим. Блок тестового діагностування (Td) подає на об'єкт тестові послідовності. Алгоритм розгалужується:

1. у разі відповідності реакцій еталонним очікуванням (Норма) система переходить у робочий режим;
2. у разі виявлення відхилень (Ненорма) система генерує діагноз і негайно перемикається в режим відновлення ($mrep$).

Етап 3. Оперативний контроль (Робочий режим $twork$). Об'єкт виконує свої цільові функції, перебуваючи у множині працездатних станів S_u . Блок оперативного контролю (Oc) здійснює безперервний контроль обладнання. Якщо сигнали виходять за межі встановлених на першому етапі допусків, блок керування перериває роботу і переводить систему в режим діагностування ($mdiag$).

Етап 4. Регламентні роботи (Режим планового обслуговування $tmaint$). З робочого режиму система може бути переведена оператором (вручну) у режим

планового обслуговування. У цьому режимі обчислення та виконання цільових функцій зупиняються для проведення регламентних робіт службою обслуговування. Якщо під час цих робіт виявляється відмова, система переводиться у режим відновлення (mrep). Якщо дефектів не виявлено – повертається до робочого режиму.

Етап 5. Локалізація несправності (Режим діагностування mdiag). Цей режим є реакцією на спрацювання оперативного контролю на третьому етапі. Мультиплексор відкриває подачу діагностичних сигналів, і блок Td реалізує поглиблену тестову програму для локалізації місця та причини несправності. Після збору даних та видачі діагнозу на зовнішній інтерфейс, система безальтернативно переходить у режим відновлення (mrep).

Етап 6. Відновлення та зворотний зв'язок (Режим відновлення mrep). Виконання функцій об'єктом зупиняється, здійснюється фізичний ремонт. На цьому етапі відбувається ключова для інтелектуальної системи оцінка достовірності.

1. Якщо засоби діагностування відпрацювали коректно (відмова була специфікованою, Sd), оператор підтверджує це, формується сигнал готовності.

2. Якщо засоби контролю спрацювали некоректно (ідентифіковано приховану відмову, Sh, яку система не знала), оператор маркує ці дані та передає їх до сховища історичних даних (Rloc) цифрового двійника. Паралельно передається команда до блоку розрахунку готовності на редукцію (відсікання) фрагментної матриці для врахування нового досвіду. Після отримання сигналів готовності від ЦД та блоку розрахунку, система повертається до Етапу 1 (Режим mset), де відбувається остаточне донавчання на нових даних, замикаючи адаптивний контур.

Висновки за підрозділом 2.1. Отже, у підрозділі 2.1 отримано перший науковий результат: розроблено комплекс рішень, що формує інфраструктурний базис інтелектуальної системи діагностування (ІСД) із донавчанням. Цей результат об'єднує три взаємодоповнюючі рівні абстракції.

1. На математичному рівні розроблено теоретико-множинну модель, яка строго формалізувала взаємозв'язки між об'єктом, засобами контролю і цифровим двійником. Визначено механізм апостеріорного оцінювання надійності шляхом редукції марковського графа.

2. На структурному рівні запропоновано дворівневу (Edge-Cloud) архітектуру. Розподіл функцій забезпечує баланс: локальний двійник (Edge) гарантує реактивність донавчання на місці, а хмарний агрегатор (Cloud) забезпечує масштабування досвіду на весь парк обладнання через федеративне навчання.

3. На алгоритмічному рівні розроблено метод технічного діагностування у вигляді скінченного автомата. Метод регламентує переходи між шістьма режимами, утворюючи замкнений цикл ідентифікації та переведення прихованих відмов у розряд відомих.

Однак, впровадження такого адаптивного контуру робить параметри системи (зокрема, показник достовірності діагностування) динамічними в часі. Це унеможливорює використання класичних статичних моделей надійності для кількісної оцінки ефективності ІСД. Для вирішення цієї проблеми у наступному підрозділі пропонується розробити онтологічний класифікатор та синтезувати комплекс базових марковських моделей, які дозволять адекватно оцінити вплив стратегій донавчання на показники готовності обладнання.

2.2 Розроблення комплексу базових моделей надійності промислового обладнання з використанням індустріального інтернету речей та ІСД із донавчанням

Як було визначено у попередньому підрозділі, функціонування інтелектуальних систем діагностування (ІСД) із контуром донавчання має динамічний характер: показник достовірності діагностування еволюціонує в часі після виявлення та аналізу прихованих відмов. Така динаміка суттєво ускладнює

застосування класичних статичних моделей теорії надійності, оскільки вони не здатні відобразити процеси структурної та параметричної адаптації системи.

Для математичного опису систем зі змінними параметрами доцільно використовувати апарат мультифрагментних марковських моделей (МФММ) [2, 8]. Проте, враховуючи розмаїття архітектур сучасного промислового обладнання, засобів базового контролю та стратегій донавчання (зокрема, локальне чи федеративне, наявність простою для налаштування тощо), постає складна методологічна проблема побудови коректної моделі для кожного конкретного випадку.

У зв'язку з цим, ключовим завданням даного підрозділу є розроблення класифікатора та синтез на його основі множини базових моделей (БМ) для відновлювального промислового обладнання систем ІоТ з донавчанням. Розроблений класифікатор дозволить систематизувати властивості об'єкта та процесу донавчання, що надасть змогу обґрунтовано генерувати адекватні марковські граfi для різних сценаріїв експлуатації та більш точно оцінювати їхній вплив на функцію готовності системи.

2.2.1 Обґрунтування ознак класифікації

Для адекватного математичного опису поведінки і оцінки готовності інтелектуальних систем діагностування (ІСД) промислового обладнання систем ІоТ з функцією донавчання необхідно визначити та систематизувати низку специфічних ознак. Ці ознаки безпосередньо впливають на топологію марковського графа станів та динаміку зміни його параметрів у часі. Враховуючи різноманіття архітектур розгортання цифрових двійників та стратегій адаптації, доцільно розподілити ці ознаки на дві великі групи: внутрішньофрагментні (статичні) та мультифрагментні (динамічні) властивості.

До групи внутрішньофрагментних властивостей, що визначають топологію графа станів у межах одного стаціонарного періоду роботи (коли параметри надійності вважаються незмінними), віднесено такі ознаки.

1. *Види контролю.* Визначає наявність та конфігурацію засобів діагностування (тільки оперативний вбудований контроль, тільки тестове діагностування, або їх комбінація). Це безпосередньо впливає на базову кількість станів об'єкта та можливість миттєвої ідентифікації аномалій.

2. *Наявність технічного обслуговування (ТО).* Ознака визначає, чи передбачено регламентне переривання робочого процесу для профілактики. Введення станів ТО створює додаткові умови для виявлення прихованих дефектів, які не фіксуються засобами оперативного контролю. Врахування цієї ознаки є важливим для вузлів IoT, де доступ персоналу до обслуговування є ускладненим.

3. *Ескалація прихованої відмови.* Характеризує здатність системи переходити з безпечного стану прихованої відмови до стану аварійної (критичної) відмови, якщо дефект тривалий час залишається нерозпізнаним. Наявність цієї ознаки вимагає додавання у граф відповідних станів та інтенсивностей ескалації.

4. *Врахування кібератак.* Дозволяє реалізувати принцип безпеки, інформованої про захищеність (Security-Informed Safety), враховуючи вплив компрометації каналів зв'язку (зокрема між Edge та Cloud рівнями) чи кібератаки на окремі компоненти на готовність системи.

До групи мультифрагментних властивостей, які є ключовими для опису процесів донавчання цифрового двійника та визначають правила переходу між стаціонарними фрагментами моделі, віднесено наступні.

1. *Тип та вимірність фрагментності.* Визначає базовий підхід до структурного моделювання: однофрагментна модель (для систем зі статичними параметрами) або мультифрагментна (для адаптивних систем). Останні, залежно від кількості незалежних векторів донавчання, можуть мати лінійну, матричну або N-вимірну структуру. Ця ознака буде в класифікаторі винесена окремо.

2. *Множина змінних параметрів.* Визначає цільові показники, що еволюціонують під час адаптації ІСД. Основним змінним параметром є достовірність діагностування (D), проте при розвитку комплексу моделей можуть

додатково враховуватися зміни інтенсивності відмов (λd), інтенсивності відновлення (μ) та ескалації ($\lambda h c$).

3. *Спеціалізація донавчання (маршрутизація ΔD)*. Ця ознака визначає архітектурні правила маршрутизації набутого досвіду. Вона вказує, чи формує конкретний тригер власну незалежну гілку еволюції достовірності, чи його досвід спрямовується до загального пулу знань системи. Залежно від обраної конфігурації, виділено чотири напрямки локалізації.

a. *Спеціалізоване ($\Delta D1$)*: досвід виявлення прихованої відмови під час ТО.

b. *Спеціалізоване ($\Delta D2$)*: досвід виявлених робочих відмов.

c. *Спеціалізоване ($\Delta D3$)*: досвід виявлення прихованої відмови після ескалації до критичної.

d. *Узагальнене (ΔD)*: спільний недиференційований досвід. Комбінація цих блоків дозволяє формувати повністю узагальнене (всі тригери формують спільне ΔD), повністю спеціалізоване (кожен тригер має свій ізольований ΔDi) або частково спеціалізоване (гібридне) донавчання. Останнє дозволяє гнучко оптимізувати розмірність марковського графа (наприклад, шляхом злиття станів донавчання), враховуючи фізичну спорідненість даних для алгоритмів машинного навчання.

4. *Ознаки навчання*. Визначає обчислювальний механізм роботи та джерело даних для донавчання цифрового двійника. За цією ознакою формуються різні структури марковських графів.

a. Відсутнє: система функціонує зі статичними параметрами.

b. Індивідуальне донавчання: адаптація моделі базується виключно на власному досвіді конкретного екземпляра обладнання. Навчання може виконуватися локально або на сервері. Але топологія марковського графа для обох випадків є ідентичною (вводиться стан налаштування $S1$), вони можуть різнитися лише значенням інтенсивності переходу $\lambda 1$.

с. Федеративне навчання: передбачає агрегацію глобального досвіду парку однотипного IoT обладнання на рівні Cloud. Це структурно відображається новими станами та переходами.

5. *Події-ініціатори (тригери) донавчання.* Ознака визначає, коли (після якої події) система ініціює перехід до налаштування. Тригером може слугувати: виявлення відмови вбудованими засобами, виявлення прихованої відмови під час ТО, або факт ескалації прихованої відмови до аварійної.

6. *Вимоги до умов навчання.* Визначає вплив процесу донавчання на виконання цільових функцій об'єктом: чи відбувається навчання у фоновому режимі (без простою), чи потребує тимчасового переведення системи у стан простою. Часто IoT edge пристрої не здатні одночасно виконувати цільові функції керування процесом та ресурсоємне перенавчання нейромоделі.

2.2.2 Розроблення онтологічного класифікатора базових моделей

На основі визначених та обґрунтованих у попередньому підрозділі внутрішньофрагментних та мультифрагментних властивостей розроблено класифікатор базових моделей (БМ) готовності ІСД.

Процес розроблення класифікатора полягав у встановленні ієрархічних зв'язків між розрізненими властивостями системи та формуванні єдиного таксономічного простору (рис. 2.5). Класифікатор розділено на три зони: верхній рівень визначає фрагментарність графа, середній формулює загальні властивості об'єкта та засобів контролю (статика графа), тоді як нижній рівень розкриває динаміку донавчання (тригери, тип навчання, спеціалізацію донавчання, тощо).

З позицій системно-онтологічного аналізу, розроблений класифікатор являє собою прикладну онтологію предметної галузі. Таку онтологію можна формально представити як впорядковану трійку [9]:

$$O = \langle X, R, F \rangle, \quad (2.10)$$

де X (множина концептів) – розроблені базові математичні моделі та їхні таксономічні ознаки (виміри класифікатора), а словник концептів та їх допустимих значень наведено у таблиці 2.1; R (множина відношень) – семантичні обмеження, що визначають залежність між експлуатаційними параметрами та забороняють неможливі комбінації ознак при формуванні простору моделей; F (функції інтерпретації) – множина експертних правил та методів, що дозволяють особі, яка приймає рішення, обрати релевантний екземпляр моделі.

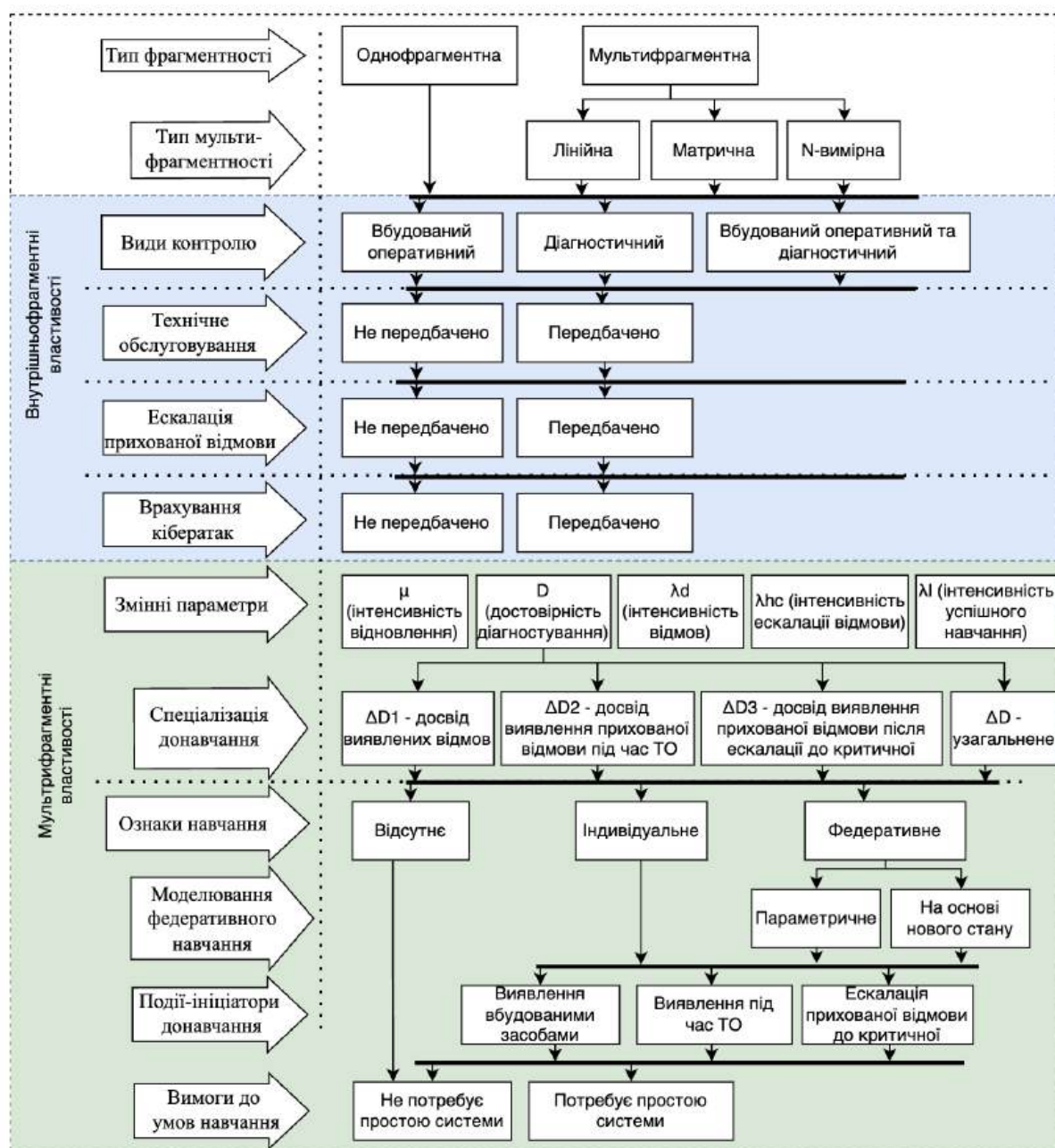


Рисунок 2.5 – класифікатор базових моделей (БМ) готовності ІСД

Таблиця 2.1 – Словник концептів онтології (множина X): таксономічні ознаки та їхні допустимі значення

№	Таксономічні ознаки	Допустимі значення
1	2	3
1	Тип фрагментності	1) однофрагментна; 2) мультифрагментна.
2	Тип мульти-фрагментності	1) лінійна; 2) матрична; 3) N-вимірна.
3	Види контролю	1) вбудований оперативний; 2) діагностичний; 3) вбудований оперативний та діагностичний.
4	Технічне обслуговування	1) передбачено; 2) не передбачено.
5	Ескалація прихованої відмови	1) передбачено; 2) не передбачено.
6	Врахування кібератак	1) Передбачено; 2) Не передбачено.
7	Змінні параметри	1) μ (інтенсивність відновлення); 2) D (достовірність діагностування); 3) λd (інтенсивність відмов); 4) $\lambda h c$ (інтенсивність ескалації відмови); 5) λl (інтенсивність успішного навчання).
8	Спеціалізація донавчання (можна обрати 1 або кілька варіантів)	1) $\Delta D1$ – досвід виявлених відмов; 2) $\Delta D2$ – досвід виявлення прихованої відмови під час ТО; 3) $\Delta D3$ – досвід виявлення прихованої відмови після ескалації до критичної; 4) ΔD – узагальнене.
9	Ознаки навчання	1) відсутнє; 2) індивідуальне; 3) федеративне.
10	Моделювання федеративного навчання	1) параметричне; 2) на основі нового стану.
11	Події-ініціатори донавчання (можна обрати 1 або кілька варіантів)	1) виявлення вбудованими засобами; 2) виявлення під час ТО; 3) ескалація прихованої відмови до критичної.

1	2	3
12	Вимоги до умов навчання	1) не потребує простою системи; 2) потребує простою системи.

Відповідно до цієї онтологічної структури, кожна конкретна базова модель (БМ) утворюється як унікальна комбінація обраних структурних та динамічних ознак. Формально, повна множина можливих моделей SPM формується як декартовий добуток підмножин варіантів реалізації за кожною k-ю ознакою класифікації:

$$SPM = F1 \times F2 \times \dots \times FK, \quad (2.11)$$

де F_k – підмножина можливих значень (варіантів реалізації) k-ї класифікаційної ознаки, а K – загальна кількість виділених у класифікаторі ознак.

Оскільки потужність множини SPM є значною (декартовий добуток генерує велику кількість комбінацій), для відсікання фізично неможливих чи недоцільних комбінацій застосовується множина відношень R. Множина R задається системою експертних семантичних правил (обмежень), що наведена у таблиці 2.2.

Таблиця 2.2 – Множина відношень та семантичних обмежень онтології (R)

№	Семантичне правило (відношення R)
1	2
1	ЯКЩО [Ознаки навчання] = «Відсутнє», ТО [Мультифрагментні властивості] = «Не застосовуються» ТА [Тип фрагментності] = «Однофрагментна»
2	ЯКЩО [Тип фрагментності] = «Однофрагментна», ТО ознаки [Мультифрагментні властивості] та [Тип мульти-фрагментності] блокуються

1	2
3	ЯКЩО [Тип фрагментності] = «Мультифрагментна», ТО [Мультифрагментні властивості] стають доступними
4	ЯКЩО [Мультифрагментні властивості] = «Доступні», ТО обов'язковим [Змінним параметром] є «Достовірність діагностування (D)»
5	ЯКЩО [Ознаки навчання] $\neq$ «Федеративне», ТО [Моделювання федеративного навчання] блокується
6	[Розмірність мульти-фрагментності] (N-вимірна) повинна дорівнювати кількості обраних варіантів у концепті [Спеціалізація донавчання]

Водночас, навіть після застосування правил R, простір валідних моделей залишається достатньо широким. Тому, з метою забезпечення можливості об'єктивного кількісного аналізу, з цього простору експертом виділяється репрезентативна досліджувана підмножина моделей SRM ($SRM \subset SPM$). Ця підмножина покриває найбільш актуальні сценарії впровадження систем з донавчанням: від простих IoT-компонентів до критичного обладнання.

У межах розробленої прикладної онтології, саме елементи виділеної підмножини SRM (цільові базові моделі BM1–BM5) формують множину альтернатив для задачі вибору [11]. При цьому об'єкти множини концептів (X) виступають у ролі критеріїв їх оцінювання. Екземпляри синтезованих базових моделей із підмножини SRM у вигляді зведеної факторно-критеріальної матриці альтернатив наведено у Таблиці 2.3.

У термінах розробленої прикладної онтології, кожній виділеній моделі $M_i \in SRM$ відповідає певна траєкторія в онтологічному графі-класифікаторі – неперервний шлях вибору концептів (ознак), який однозначно фіксує топологію марковського графа. Наприклад, вибір траєкторії, що проходить через блоки «матрична фрагментність», «ескалація відмови» та «частково спеціалізоване

донавчання», формує розгалужену математичну модель із паралельними гілками еволюції параметра достовірності.

Таблиця 2.3 – Факторно-критеріальна матриця альтернатив

Екземпляр (Модель)	BM1	BM2	BM3	BM4	BM5
1	2	3	4	5	6
Тип фрагментності	Мульти- фрагментна	Мульти- фрагментна	Мультифрагментна	Мульти- фрагментна	Мульти- фрагментна
Тип мульти- фрагментності	Лінійна	Лінійна	N-вимірна	Матрична	N-вимірна
Види контролю	Діагностичний	Вбудований оперативний та діагностичний	Діагностичний	Діагностичний	Діагностичний
Технічне обслуговування	Не передбачено	Передбачено	Передбачено	Передбачено	Передбачено
Ескалація прихованої відмови	Не передбачено	Не передбачено	Передбачено	Передбачено	Передбачено
Врахування кібератак	Не передбачено	Не передбачено	Не передбачено	Не передбачено	Не передбачено
Змінні параметри	D (достовірність діагностування)	D (достовірність діагностування)	D (достовірність діагностування)	D (достовірність діагностування)	D (достовірність діагностування)
Спеціалізація донавчання	ΔD – узагальнене	ΔD – узагальнене	1) $\Delta D1$ – досвід виявлених відмов. 2) $\Delta D2$ – досвід виявлення прихованої відмови під час ТО. 3) $\Delta D3$ – досвід виявлення прихованої відмови після ескалації до критичної.	1) $\Delta D1$ – досвід виявлених відмов. 4) ΔD – узагальнене.	1) $\Delta D1$ – досвід виявлених відмов. 2) $\Delta D2$ – досвід виявлення прихованої відмови під час ТО. 3) $\Delta D3$ – досвід виявлення прихованої відмови після ескалації до критичної.
Ознаки навчання	Індивідуальне	Індивідуальне	Індивідуальне	Індивідуальне	Федеративне
Моделювання федеративного навчання	-	-	-	-	На основі нового стану

1	2	3	4	5	6
Події-ініціатори донавчання	Виявлення вбудованими засобами	Виявлення вбудованими засобами	1) Виявлення вбудованими засобами. 2) Виявлення під час ТО. 3) Ескалація прихованої відмови до критичної.	1) Виявлення вбудованими засобами. 2) Виявлення під час ТО. 3) Ескалація прихованої відмови до критичної.	1) Виявлення вбудованими засобами. 2) Виявлення під час ТО. 3) Ескалація прихованої відмови до критичної.
Вимоги до умов навчання	Потребує простою системи	Потребує простою системи	Потребує простою системи	Потребує простою системи	Потребує простою системи

Таким чином, онтологічний класифікатор виконує роль конструктора, на базі якого у наступному підрозділі буде виконуватись розроблення комплексу репрезентативних базових моделей готовності промислового обладнання з використання ПоТ. Правила побудови такого класифікатора та принципи формалізації інформаційної технології на його базі висвітлено автором у роботі [3].

2.2.3 Розроблення комплексу базових моделей надійності

На основі розробленого класифікатора здійснюється структурний синтез досліджуваної множини моделей (SRM), які утворюють комплекс базових моделей (БМ). Для забезпечення можливості об'єктивного кількісного аналізу ефективності впровадження ІСД із донавчанням, кожна унікальна траєкторія з класифікатора генерує не одну ізольовану модель, а *порівняльну тріаду базових моделей*. Такий методологічний підхід дозволяє зіставляти показники адаптивної системи з її традиційними (статичними) та ідеалізованими аналогами в ідентичних умовах експлуатації.

Формально, для кожної досліджуваної базової моделі $BM_i \in SRM$ формується кортеж (тріада) моделей $Tr(BM_i)$:

$$Tr(BM_i) = \langle BM_i^{dyn}, BM_i^{stat}, BM_i^{id} \rangle, \quad (2.12)$$

де BM_i^{dyn} – динамічна мультифрагментна модель із донавчанням; BM_i^{stat} – статична однофрагментна модель без донавчання; BM_i^{id} – ідеалізована модель (з абсолютною достовірністю).

Правила формування та редукції графів станів для кожної складової тріади наведено нижче.

1. *Динамічна мультифрагментна модель із донавчанням (BM_i^{dyn})* – це основна мультифрагментна марковська модель (МФММ), траєкторія формування якої проходить через обидві зони класифікатора (статичних та динамічних властивостей). Простір станів цієї моделі є найбільш повним. Окрім базових станів працездатності (Su), явної відмови (Sd) та прихованої відмови (Sh), він обов'язково містить множину станів налаштування (навчання) цифрового двійника Sl. Перебування у стані Sl і успішне навчання ініціює міжфрагментний перехід графа до нового виміру, де параметр достовірності діагностування отримує приріст ΔD . Ця модель відображає еволюцію системи у часі.

2. *Статична модель без донавчання (BM_i^{stat})* – ця модель описує роботу традиційної системи технічного діагностування зі статичними параметрами контролю. Структурно вона утворюється шляхом математичної редукції (згортання) графа основної моделі BM_i^{dyn} . З простору станів вилучаються всі стани донавчання ($Sl = \emptyset$), а інтенсивності переходів, що вели до них, замикаються на стан працездатності поточного фрагмента (Su). Таким чином, мультифрагментна марковська модель (МФММ) перетворюється на класичну однофрагментну модель (ОФМ). Параметр достовірності діагностування у цій моделі залишається незмінним протягом усього часу експлуатації ($D = \text{const} = D_{\text{start}}$). Дана модель слугує базовою лінією (baseline) для оцінювання виграшу від впровадження донавчання.

3. *Ідеалізована модель (BM_i^{id})* – Модель описує гіпотетичну систему, засоби діагностування якої володіють абсолютною достовірністю ($D=1$). Відповідно, ймовірність виникнення невиявленої (прихованої) відмови дорівнює

нулю. Структурно ця модель генерується шляхом подальшої редукції графа BM_i^{stat} : з нього повністю вилучаються стани прихованих відмов ($Sh=\emptyset$) та стани їх ескалації ($Shc=\emptyset$). Усі потоки відмов у такій системі є розпізнаваними і переводять об'єкт у стан явної відмови Sd для подальшого відновлення. Ця модель формує теоретичну асимптотичну межу функції готовності, до якої повинна наближатися ІСД у процесі свого донавчання.

Далі будуть синтезуватися та візуалізуються лише найбільш розгорнуті динамічні графи (BM_i^{dyn}). Структури статичних та ідеальних моделей однозначно виводяться з них за вищеописаними правилами, що створює математичний базис для подальшої програмної реалізації алгоритмів оцінювання.

Відповідно до визначених у класифікаторі траєкторій, досліджувану множину динамічних моделей розділено на дві групи: лінійні моделі з узагальненим донавчанням та розгалужені моделі зі спеціалізацією досвіду.

Група лінійних моделей з узагальненим донавчанням. До цієї групи віднесено моделі, що покривають сценарії впровадження ІСД для типових компонентів систем індустріального інтернету речей (наприклад, IoT-сенсорів, базових приводів), які не потребують складної спеціалізації донавчання або засобів федеративного навчання. Обчислювальні ресурси таких систем дозволяють реалізовувати універсальне оновлення параметрів, за якого будь-який ідентифікований дефект формує спільний досвід.

Базова модель 1 (БМ1, ВМ1). Модель з узагальненим донавчанням. Траєкторія формування моделі у класифікаторі (рис. 2.6) описується наступним ланцюгом ознак: Лінійна мультифрагментність $\rightarrow$ Діагностичний контроль $\rightarrow$ Змінний параметр $D \rightarrow$ Оперативний контроль $\rightarrow$ Узагальнене донавчання (спільне ΔD) $\rightarrow$ Індивідуальне донавчання $\rightarrow$ Тригер навчання після виявлення відмови вбудованими засобами $\rightarrow$ Потребує простою. Граф станів моделі (рис. 2.7) містить базові стани: працездатний (Su), відмови (Sd), прихованої відмови (Sh) та стан донавчання (Sl). Після виявлення прихованої відмови та відновлення, система з імовірністю L переходить до стану налаштування Sl , що ініціює міжфрагментний перехід до нового простору станів Su' із підвищеним значенням

достовірності $D1=D0+\Delta D$. Ця модель слугує базисом для дослідження загального впливу донавчання на показник експлуатаційної готовності.

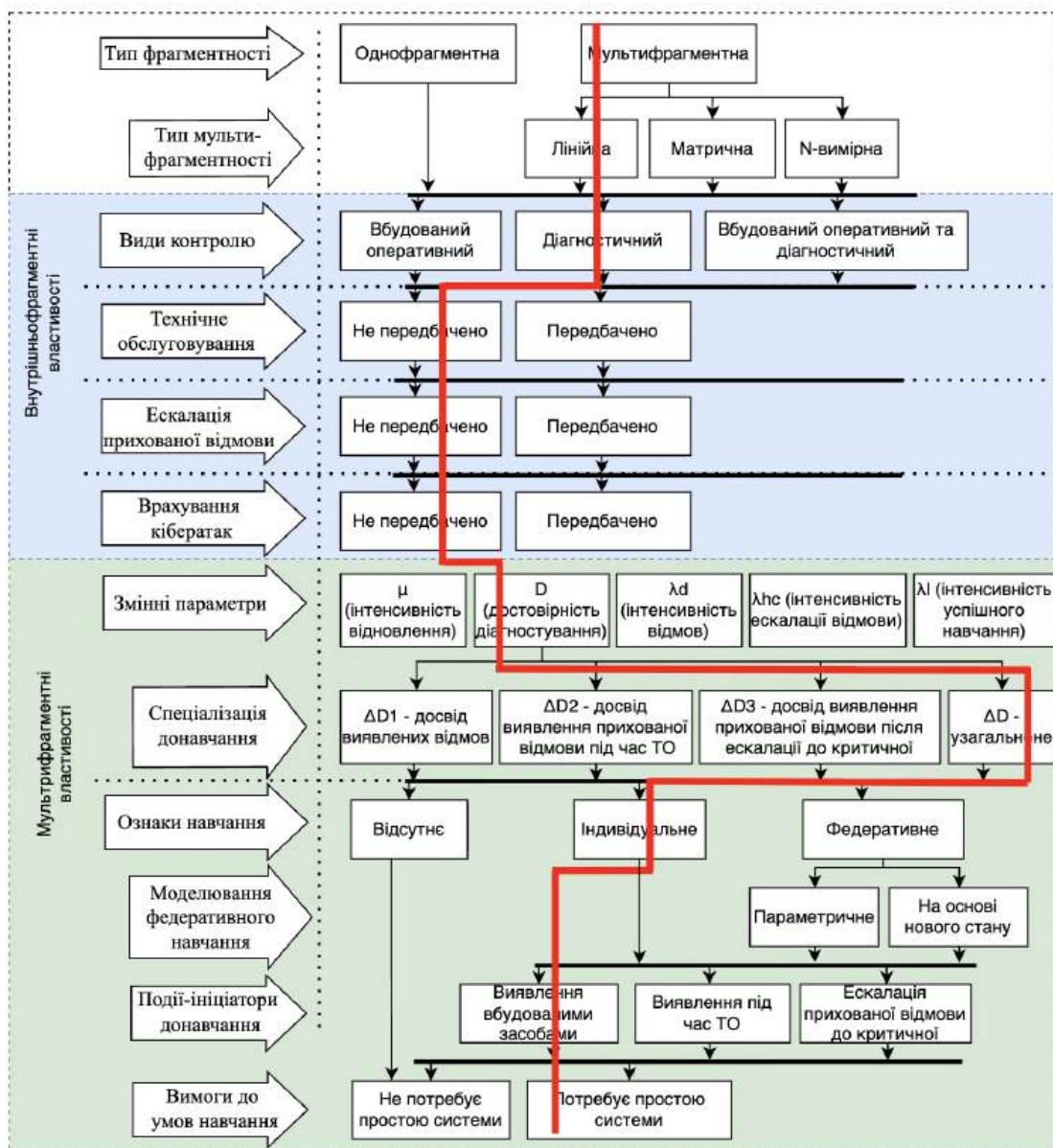


Рисунок 2.6 – траєкторія формування базової моделі BM1

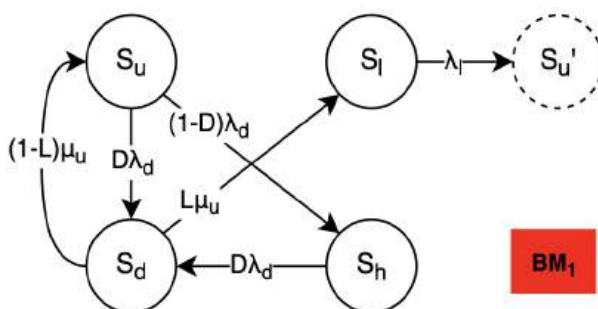


Рисунок 2.7 – Базова модель BM1

Базова модель 2 (БМ2, ВМ2). Модель з багаторівневим контролем. Траєкторія формування моделі (рис. 2.8) розширює ВМ2 за рахунок обрання ознаки «Оперативний та Діагностичний контроль» та додавання явного стану обслуговування (Sad). Відповідно, у графі станів (рис. 2.9) в межах фрагмента виділяються окремі стан для виявленої (але не локалізованої відмови) Sed та стан аварійної діагностики службою ТО Sad (стан аварійного maintenance), коли система діагностування не змогла відпрацювати, але контроль виявив відмову. Тобто при Sad служба ТО має менше інформації і відповідно відновлення буде більшим. Стан Sd тут – це успішно виконаний оперативний контроль та діагностування. Донавчання цифрового двійника ініціюється після проходження повного циклу діагностування та відновлення.

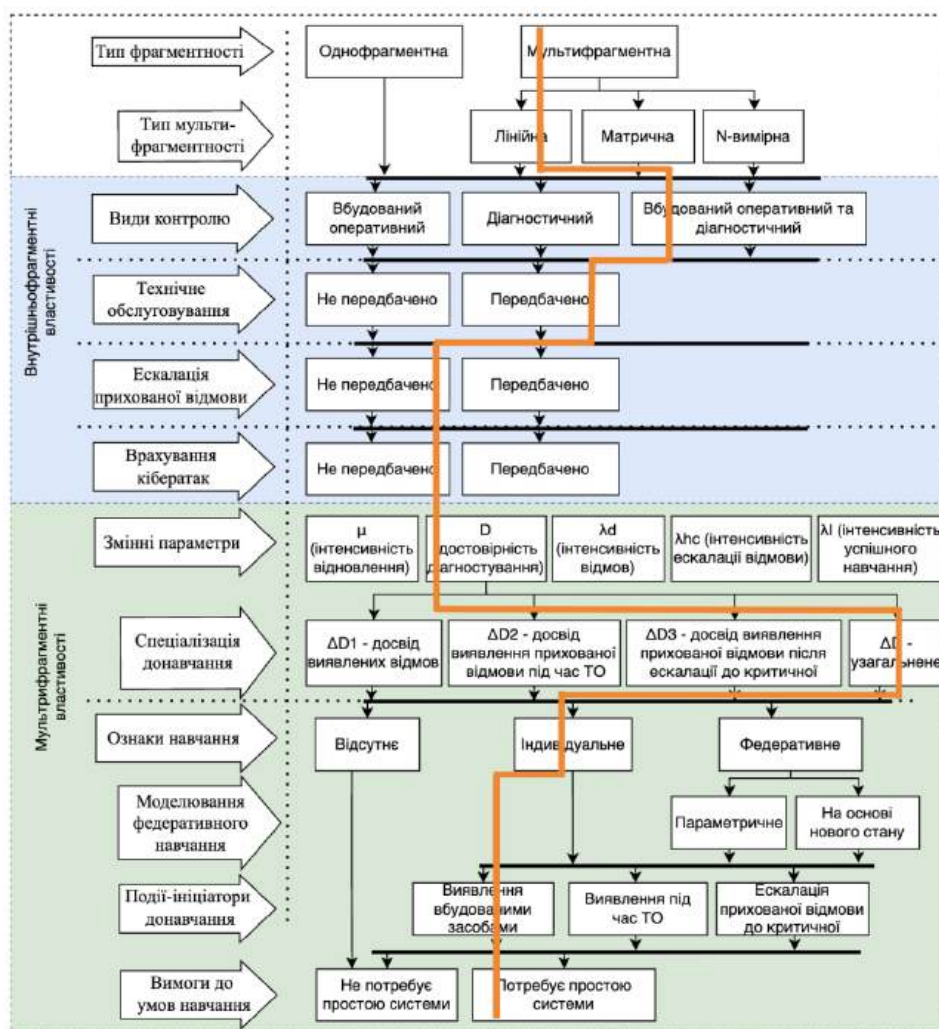


Рисунок 2.8 – траєкторія формування базової моделі ВМ2

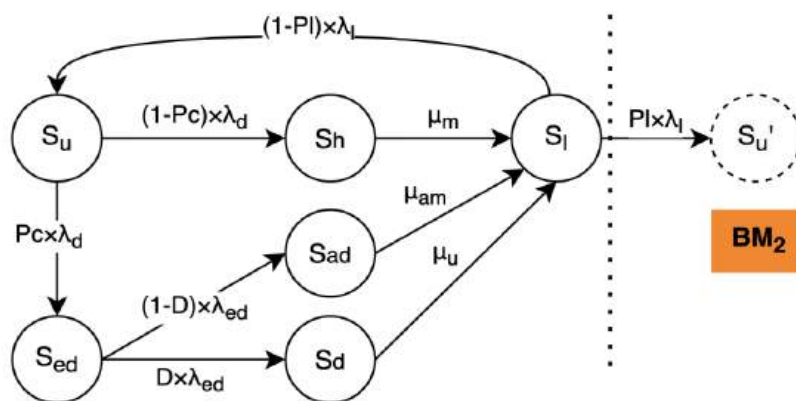


Рисунок 2.9 – Базова модель BM2

Група розгалужених моделей зі спеціалізацією досвіду. Ця група розроблена для математичного опису ІСД критичного промислового обладнання (наприклад, насосних агрегатів малих модульних реакторів, SMR). Її структурною відмінністю є обов'язкова наявність планового технічного обслуговування (ТО), можливість ескалації дефектів до аварій та необхідність спеціалізації донавчання.

Базова модель 3 (БМ3, BM3). Модель з повною спеціалізацією донавчання. Граф моделі (рис. 2.10) має розгалужену багатовимірну структуру. У базовому фрагменті присутні стани технічного обслуговування (S_m) та ескалації прихованої відмови до критичної (S_{hc}). Завдяки вибору повної спеціалізації граф містить три незалежні стани донавчання (S_{l1}, S_{l2}, S_{l3}), кожен з яких ініціює перехід у свій незалежний вимір фрагментності з індивідуальними кроками приросту знань.

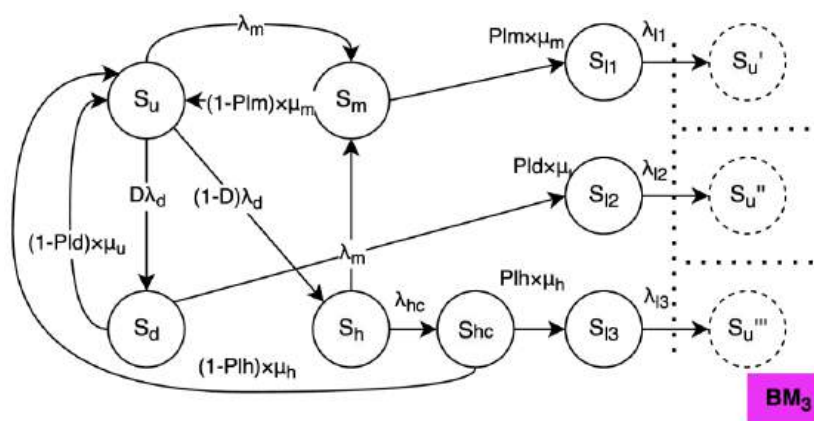


Рисунок 2.10 – Базова модель BM3

Траєкторія моделі у класифікаторі (рис. 2.11): N-вимірна мультифрагментність → Діагностичний контроль → Врахування ТО → Врахування ескалації → Відсутність врахування кібератак → Змінний параметр D → Повністю спеціалізоване донавчання (ізолювані $\Delta D1, \Delta D2, \Delta D3$) → Індивідуальне навчання → Усі події ініціатори навчання → Навчання потребує простою.

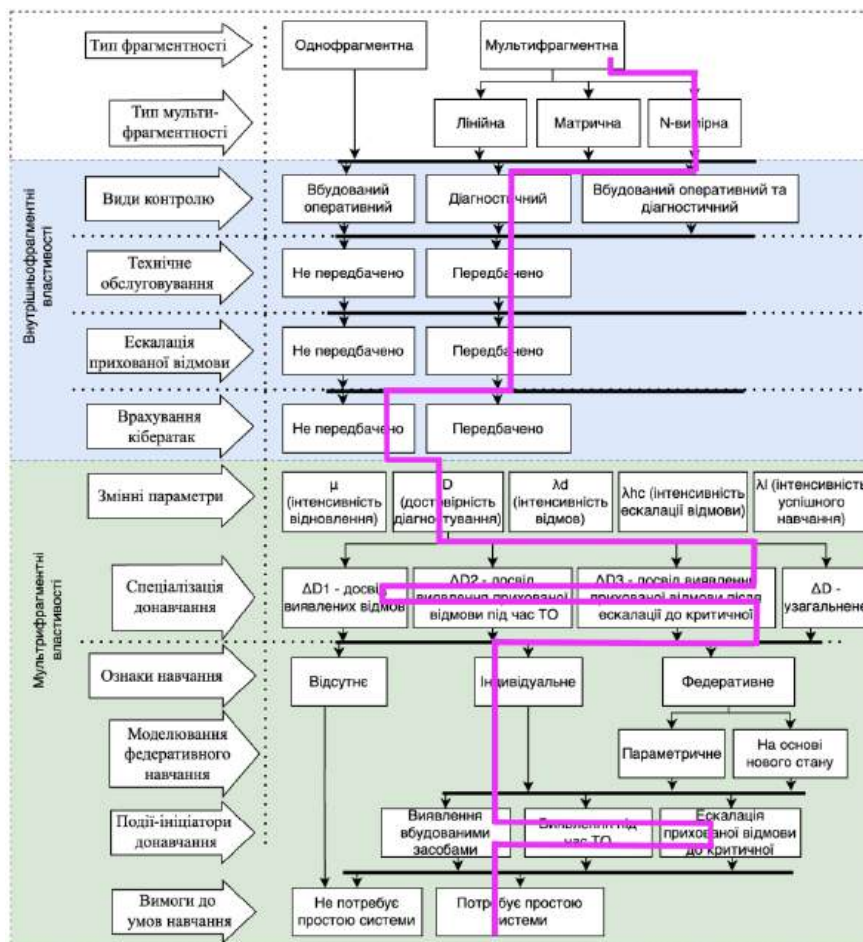


Рисунок 2.11 – траєкторія формування базової моделі BM3

Базова модель 4 (БМ4, BM1). Гібридна модель зі злиттям досвіду. Траєкторія формування (рис. 2.12) є оптимізацією моделі BM3 шляхом вибору спеціалізації тільки на одному типі навчання «досвід виявлених відмов», усі інші відмови опрацьовуються разом в рамках загальної спеціалізації. Таким чином маємо граф матричної форми, а не N-вимірний. Концептуальні засади та

математична структура цієї моделі були розроблені авторами у роботі [2]. У цій моделі (рис. 2.13) здійснюється маршрутизація набутого досвіду: досвід від виявлених робочих відмов ($\Delta D2$) формує власну ізольовану гілку еволюції, тоді як досвід від усіх прихованих відмов – виявлених під час ТО ($\Delta D1$) або після ескалації ($\Delta D3$) – об'єднується. Об'єднаний вузол навчання позначається як Slh (learning hidden), що акумулює в собі спільну спеціалізацію. Таке злиття застосовується, оскільки фізична природа цих даних для алгоритмів машинного навчання є спорідненою, що дозволяє суттєво зменшити розмірність матриці без суттєвої втрати адекватності. Або також змодельовати ситуацію коли дійсно такі відмови опрацьовуються разом.

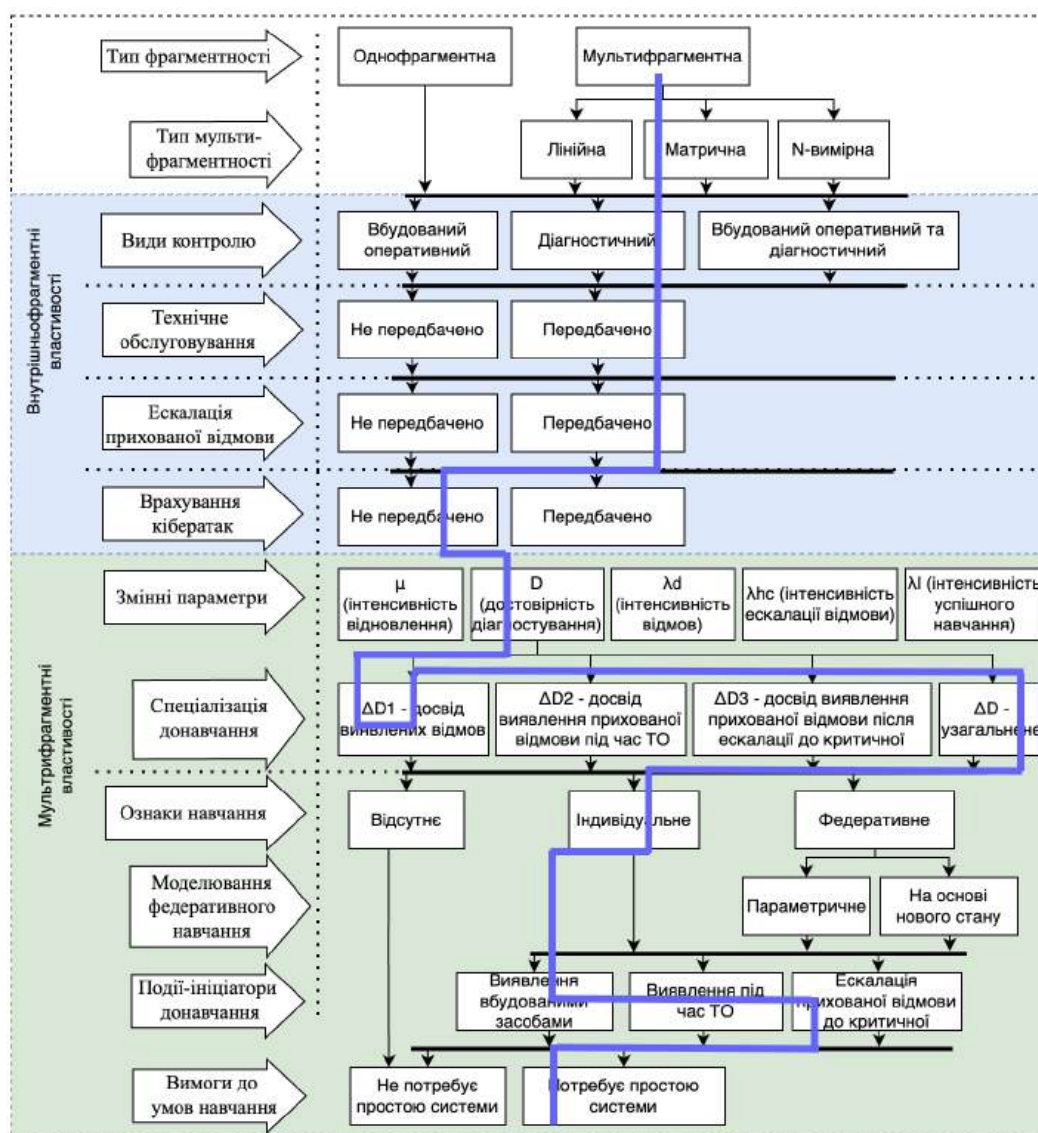


Рисунок 2.12 – траєкторія формування базової моделі BM4

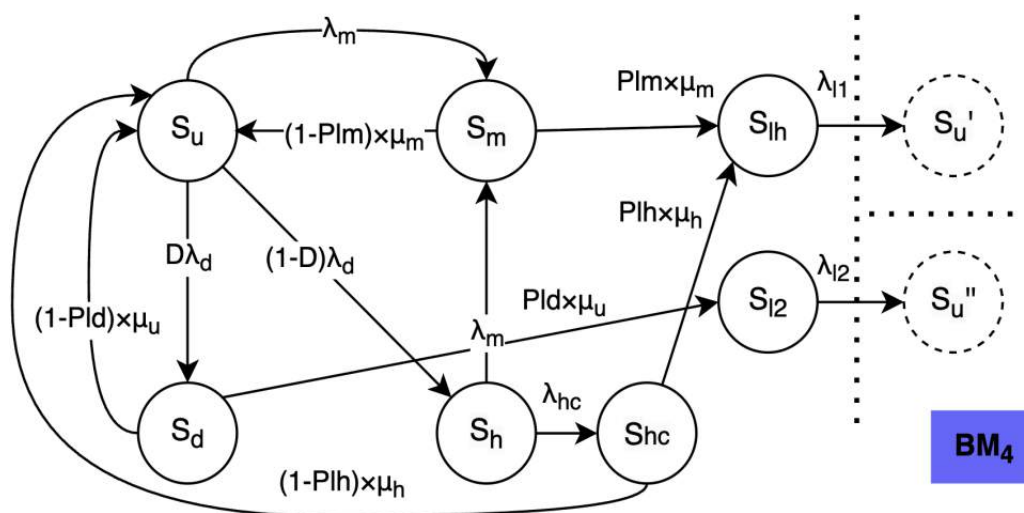


Рисунок 2.13 – Базова модель BM4

Базова модель 5 (BM5, BM5). Модель з федеративним навчанням. У граф моделі (рис. 2.14) вводиться незалежний перехід або стан (S_{lf}), який дозволяє системі переходити до фрагмента з вищим D , використовуючи агрегований глобальний досвід хмарного сховища (Cloud), навіть якщо локальна прихована відмова на даному конкретному екземплярі обладнання ще не сталася. Траєкторія формування (рис. 2.15) розширює BM3 за рахунок обрання гілки «Федеративне донавчання».

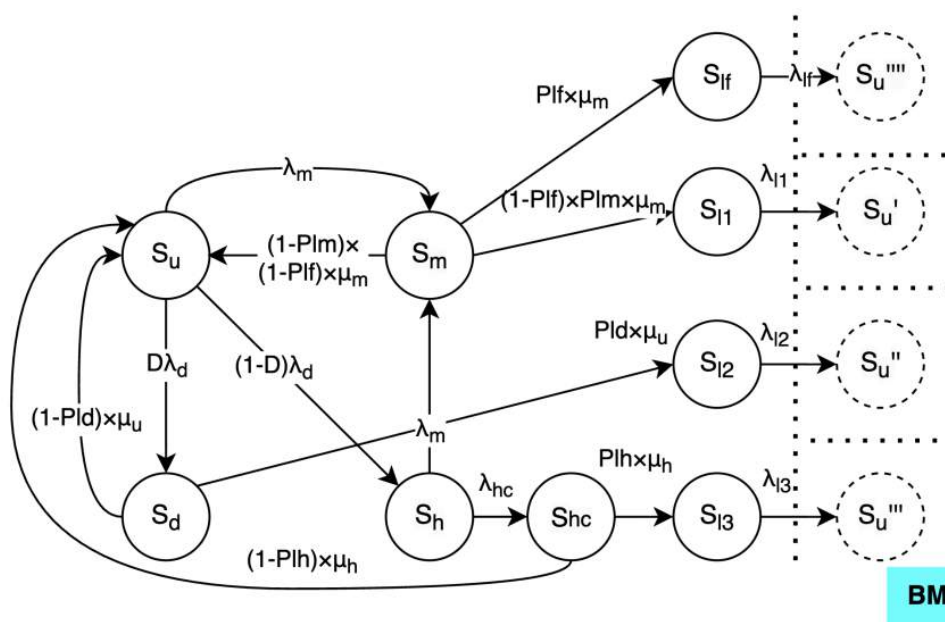


Рисунок 2.14 – Базова модель BM5

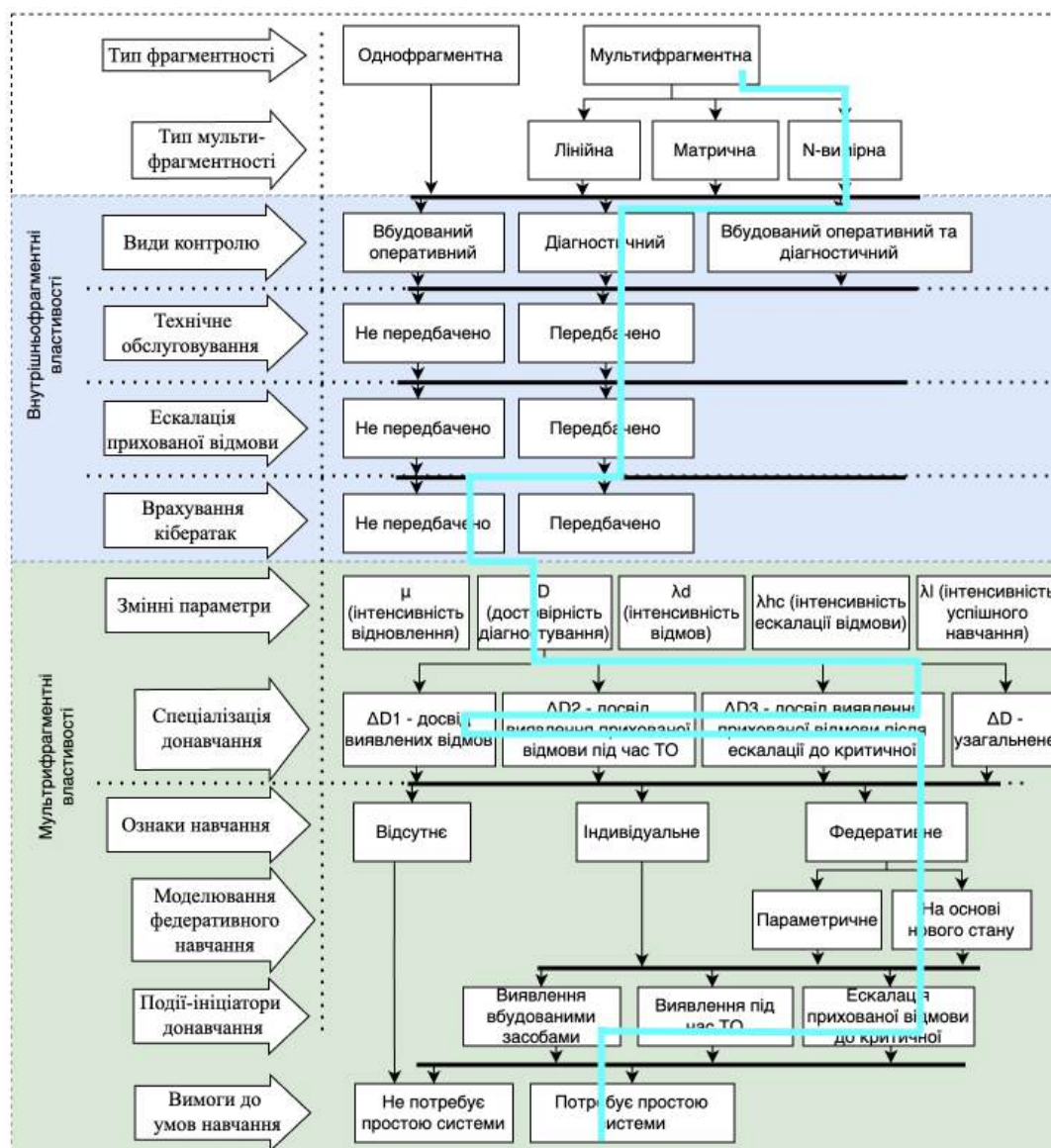


Рисунок 2.15 – траєкторія формування базової моделі BM5

Таким чином, сукупність усіх згенерованих концептуальних порівняльних тріад $Tr(BM_i)$ для кожної виділеної репрезентативної траєкторії класифікатора (BM1-BM5) утворює загальну множину базових моделей. Розроблений класифікатор разом із множиною базових моделей формують першу складову другого наукового результату (НР2) даної роботи. Матеріали досліджень щодо формалізації класифікатора та використання цих моделей успішно пройшли апробацію і висвітлені у працях [2, 3, 8, 10]. Остаточне математичне розгортання цього комплексу у мультифрагментні моделі та оцінювання впливу ІСД на показники надійності буде виконано у наступному розділі.

2.2.4 Багатокритеріальне ранжування та вибір цільової базової моделі

Для практичного застосування розробленої прикладної онтології, функцію логічного виведення (F) з формули (2.10) доцільно реалізувати з використанням методу експертного багатокритеріального ранжування альтернатив [12].

У запропонованому онтологічному підході є розмежування науково-аналітичної та прикладної (інженерної) складових. На першому етапі синтезується факторно-критеріальна матриця альтернатив (екземпляри SRM, наведені у табл. 2.3). На експлуатаційному етапі особі, яка приймає рішення і виступає в ролі експерта предметної галузі (інженеру з надійності), пропонується визначити профіль вимог до цільової моделі. При залишенні базової значущості всіх 12-ти критеріїв (таксономічних ознак) рівноцінною та встановленні напрямку оптимізації на максимум, експерт задає бальні оцінки для підмножини допустимих варіантів. Релевантним для поточного завдання варіантам присвоюється найвищий бал (10), а нерелевантним – найнижчий (0). Сформована множина бальних оцінок відіграє роль експертного правила, за яким обчислювальне ядро системи ранжує моделі (BM1–BM5).

Для верифікації розробленого методу та перевірки його адекватності було використано інструментарій інформаційно-аналітичної системи «ПОЛПЕДР-Альтернатива» [13]. В інструментальному середовищі було згенеровано дві множини вагових коефіцієнтів (експертних правил), що імітують запити для двох різних експлуатаційних сценаріїв (табл. 2.4).

Графічний розподіл результатів у системі подається у вигляді складених гістограм, де загальна висота стовпця (по осі ординат) відповідає підсумковому інтегральному рейтингу альтернативи (відповідної БМ), а різнокольорові сегменти відображають адитивний внесок кожного окремого критерію.

Сценарій 1. При введенні експертом бальних оцінок для простого необслуговуваного вузла (10 балів призначено варіантам «ТО: Не передбачено»,

«Навчання: Індивідуальне» тощо), система «ПОЛІЕДР-Альтернатива» виводить на перше місце найбільш релевантну лінійну модель ВМ1 (рис. 2.16).

Таблиця 2.4 – Множини експертних оцінок допустимих варіантів для різних сценаріїв вибору

№	Таксономічна ознака	(Критерій) Варіанти, що отримали найвищий бал (10) у Сценарії 1 (Простий вузол)	Варіанти, що отримали найвищий бал (10) у Сценарії 2 (Флот критичного обладнання)
1	Тип фрагментності	Мультифрагментна	Мультифрагментна
2	Тип мультифрагментності	Лінійна	N-вимірна
3	Види контролю	Діагностичний	Діагностичний
4	Технічне обслуговування	Не передбачено	Передбачено
5	Ескалація прихованої відмови	Не передбачено	Передбачено
6	Врахування кібератак	Не передбачено	Не передбачено
7	Змінні параметри	D (достовірність діагностування)	D (достовірність діагностування)
8	Спеціалізація донавчання	ΔD – узагальнене	$\Delta D1, \Delta D2, \Delta D3$ (всі 3 варіанти)
9	Ознаки навчання	Індивідуальне	Федеративне
10	Моделювання федер. навчання	не застосовується (0 балів)	На основі нового стану
11	Події-ініціатори донавчання	Виявлення вбудованими засобами	Виявлення вбудов. засобами, під час ТО, Ескалація (всі 3)
12	Вимоги до умов навчання	Потребує простою	Потребує простою

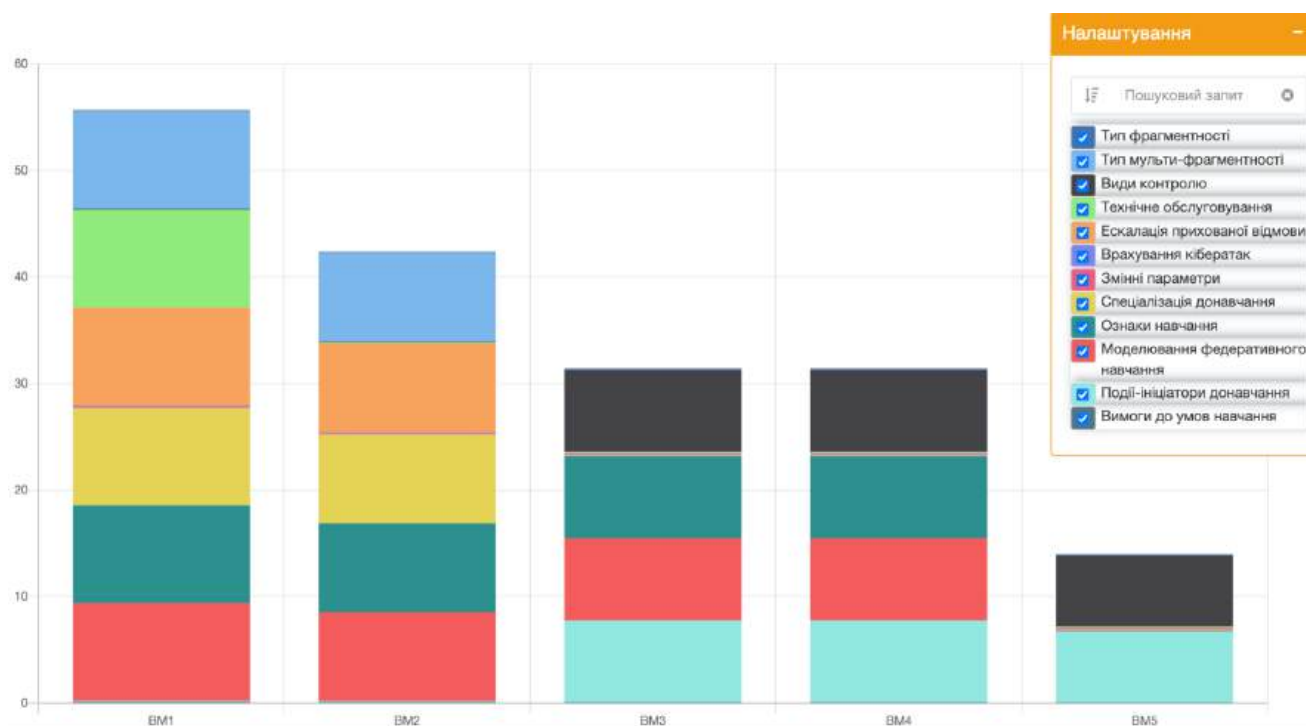


Рисунок 2.16 – Розподіл результатів ранжування альтернатив для Сценарію 1

Сценарій 2. Натомість, застосування експертних оцінок для флоту критичного обладнання (10 балів призначено варіантам «Навчання: Федеративне», «ТО: Передбачено», а також множинам подій-ініціаторів) змінює вибір онтологічний класифікатора на користь моделі BM5 (рис. 2.17).

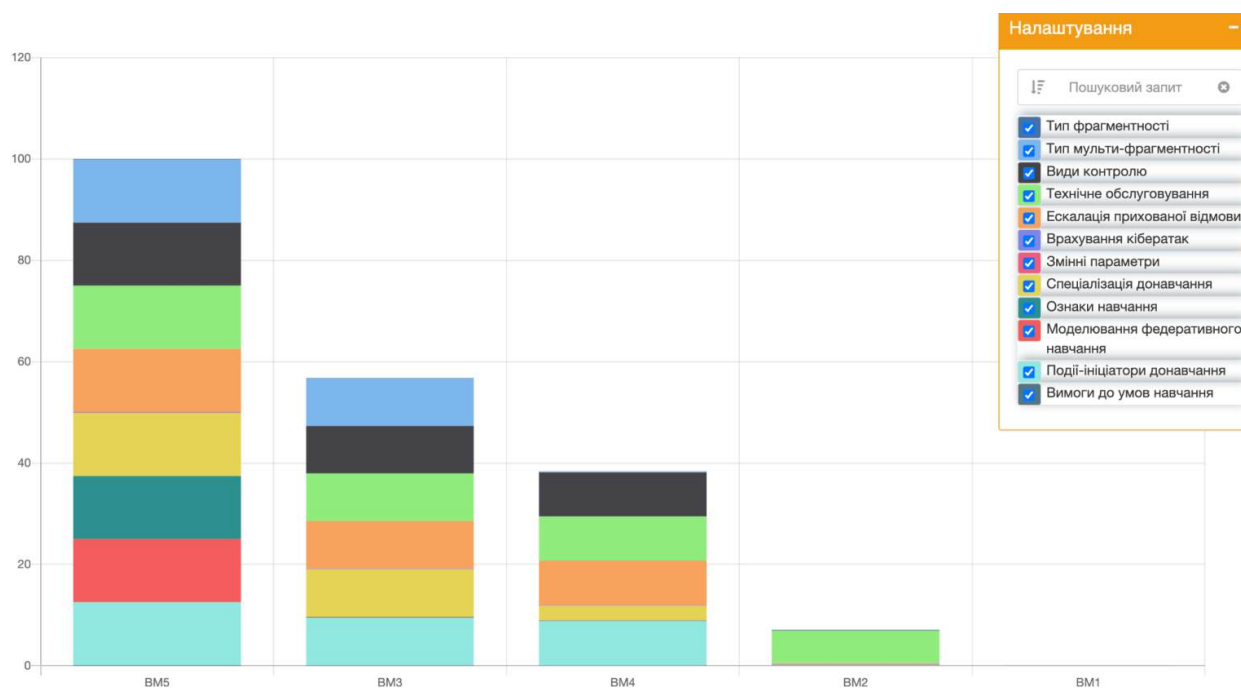


Рисунок 2.17 – Розподіл результатів ранжування альтернатив для Сценарію 2

Отже, застосування методу багатокритеріального ранжування в межах розробленої прикладної онтології дозволяє обґрунтовано обрати модель марковського графа, уникаючи суб'єктивних помилок при проектуванні ІСД.

2.3. Обґрунтування припущень

Для побудови та математичного дослідження синтезованого комплексу базових марковських моделей готовності прийнято низку припущень, які розділено на дві групи.

До загальних математичних припущень віднесено наступні.

1. Ординарність потоків подій. Імовірність одночасного настання двох і більше подій (наприклад, відмови та завершення ремонту) у нескінченно малий проміжок часу дорівнює нулю.

2. Експоненційний розподіл часу та відсутність післядії. У межах роботи для опису поведінки системи обрано апарат марковських, а не напівмарковських випадкових процесів, ґрунтуючись на припущенні про відсутність післядії. Хоча в реальних умовах експлуатації процеси старіння та відновлення обладнання часто підпорядковуються закону Вейбулла або логнормальному розподілу, приймається, що інтенсивності переходів (відмов та відновлень) є сталими і не змінюються, а час перебування у станах має експоненційний розподіл. Таке спрощення є цілком обґрунтованим: оскільки головним фокусом дослідження є оцінювання впливу донавчання на готовність, фізичним старінням компонентів у межах робочого циклу можна знехтувати, уникнувши надмірного ускладнення математичної моделі.

До специфічних припущень функціонування ІСД із донавчанням віднесено наступні.

1. Необмеженість кількості відновлень системи.

2. Рівномірне та монотонне зростання D . Параметр достовірності діагностування (D) змінюється дискретно, рівномірно (із заданим кроком ΔD) і

виключно в позитивну сторону після кожного успішного циклу донавчання. Ефект алгоритмічної деградації моделі не розглядається.

3. Ідеальність механізму оновлення. Апаратно-програмні засоби та канали зв'язку, що відповідають за передачу та розгортання результатів донавчання на пристрої, вважаються абсолютно надійними.

4. Відсутність автоматичного парирування відмов. У роботі не досліджувалися механізми апаратного чи програмного резервування. Проте зростання достовірності діагностування (D) завдяки донавчанню позитивно впливатиме на ефективність потенційних систем автоматичного відновлення, оскільки точна ідентифікація дефекту є необхідною умовою для їхнього успішного спрацювання.

5. Обмеженість джерел діагностичних даних. Зазначається як припущення, що система ІСД спирається виключно на штатні вбудовані засоби телеметрії та контролю. Використання додаткового резервного або зовнішнього тестування (наприклад, періодичні ручні перевірки тепловізорами, акустичними дефектоскопами тощо) для підвищення якості діагностики поза контуром донавчання цифрового двійника не розглядалося.

2.4 Висновки до другого розділу

У другому розділі за результатами досліджень отримано такі результати.

1. Розроблено комплекс рішень, що формує інфраструктурний базис ІСД із донавчанням на основі цифрового двійника, який включає три рівні.

а. На математичному рівні розроблено теоретико-множинну модель взаємозв'язків об'єкта, засобів контролю і цифрового двійника із апостеріорним оцінювання готовності шляхом редукції графа.

б. На структурному рівні запропоновано дворівневу (Edge-Cloud) архітектуру, що забезпечує реактивність локального донавчання та масштабування досвіду через федеративне навчання з описом зв'язків і компонентів.

с. На алгоритмічному рівні розроблено метод технічного діагностування у вигляді скінченного автомата, що регламентує переходи між шістьма режимами. **Зазначений комплекс становить перший науковий результат (НР1).** Основні концептуальні та архітектурні положення даного наукового результату висвітлено у фахових працях [1-3], колективній монографії [4], а також апробовано у матеріалах конференції [5].

2. Зазначений комплекс рішень дозволив сформувати та отримати перший науковий результат (НР1): Вперше запропоновано теоретико-множинну модель та архітектуру інтелектуальної системи діагностування промислового обладнання систем індустріального інтернету речей з використанням цифрового двійника, яка, на відміну від існуючих, забезпечує локальне та федеративне донавчання, коригування тестових послідовностей і параметрів оперативного контролю на базі виявлених прихованих відмов, а також можливість динамічного обчислення функції готовності, що надає змогу знизити ймовірність прихованих відмов та підвищити керованість процесами обслуговування

3. Розроблено онтологічний класифікатор базових моделей готовності ІСД промислового обладнання систем індустріального інтернету речей з донавчанням. Простір властивостей системи декомпозовано на три рівні (фрагментність, статика графа, динаміка донавчання). Введено ознаку спеціалізації (маршрутизації) досвіду.

4. На основі класифікатора та системи експертних семантичних обмежень (множини відношень R) синтезовано факторно-критеріальну множину альтернатив – комплекс базових моделей (BM1–BM5). Множина охоплює лінійні моделі з узагальненим донавчанням та розгалужені матричні і N -вимірні моделі з обслуговуванням та індивідуальним і федеративним донавчанням. Реалізація функції логічного виведення (F) на основі методу багатокритеріального ранжування (у системі «ПОЛІЕДР-Альтернатива») забезпечила можливість автоматизованого вибору цільової моделі. **Онтологічний класифікатор та комплекс базових моделей становлять першу складову другого наукового**

результату (НР2), що повністю буде сформований у наступному розділі. Матеріали досліджень, що лягли в основу цієї складової наукового результату, пройшли міжнародну апробацію та висвітлені у наукових працях [2, 3, 8, 10].

5. Сформульовано загальні математичні та специфічні припущення функціонування ІСД для подальшого моделювання на основі комплексу БМ при розробці МФММ.

Література до другого розділу

1. Щеглов В. Р., Морозова О. І. Методи та технології розроблення цифрових двійників для гарантоздатних систем індустриального інтернету речей. *Системи управління, навігації та зв'язку*. 2022. Вип. 4 (70). С. 127–137. DOI: 10.26906/SUNZ.2022.4.127.

2. Kharchenko V., Shcheglov V., Ivasiuk O., Morozova O. Digital Twin-Based Lifecycle Methodology for Ensuring Safety of NPP/SMR I&C Systems. *Technologies*. 2026. Vol. 14, no. 1, article no. 46. P. 1–33. DOI: 10.3390/technologies14010046.

3. Щеглов В. Р. Інформаційна технологія оцінювання готовності інтелектуальних систем діагностування з донавчанням на основі цифрових двійників. *Measuring and computing devices in technological processes*. 2026. № 1. С. 87–96. DOI: 10.31891/2219-9365-2026-85-11.

4. Гаєвський В. В., Кунуп Т. В., Морозова О. І., Щеглов В. Р. Розділ 11. Інтелектуальні системи інтернету речей з використанням технологій цифрових двійників. *Methods and technologies of ensuring quality and safety of intelligent systems* : кол. монографія / за заг. ред. В. С. Харченка, О. І. Морозової. Харків : Міністерство освіти і науки України, Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ», 2023. С. 193–224.

5. Щеглов В. Р., Морозова О. І. Застосування технології цифрових двійників відносно безпілотних літальних апаратів. *Авіація, промисловість,*

суспільство : матеріали наук.-практ. конф., м. Харків, 2024. Харків, 2024. С. 47–51.

6. Cohen B. Set Theory as a Semantic Framework for Object Oriented Modeling. *Object-Oriented Technologies*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1998. P. 161–168. DOI: 10.1007/3-540-69687-3_34.

7. Tazaki E., Amagasa M. Structural modeling in a class of systems using fuzzy sets theory. *IEEE Conference on Decision and Control including the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications* : Proc. Conf., New Orleans, LA, USA, 1977. P. 1408–1413. DOI: 10.1109/CDC.1977.271524.

8. Shcheglov V., Kharchenko V., Borges J., Morozova O. Availability Markov Models of Complex Equipment with Relearning Diagnostic Means after Detecting Latent Failures. *Dependable Systems, Services and Technologies (DESSERT'2023)* : Proc. 13th IEEE Int. Conf., Athens, Greece, 2023. P. 1–7. DOI: 10.1109/DESSERT61349.2023.10416539.

9. Горборуков В. В., Франчук О. В., Приходнюк В. В., Кирієнко О. В. Трансдисциплінарна інформаційно-аналітична платформа підтримки процесів оцінювання. *Наукові записки НаУКМА*. Комп'ютерні науки. 2022. Т. 5. С. 79–84. DOI: 10.18523/2617-3808.2022.5.79-84.

10. Shcheglov V., Kharchenko V., Volochiy B., Morozova O., Zaitseva E. Availability Assessing Intelligent Diagnostic System with Relearning after Complex Equipment Failures: Structural-Automata Model-based Methodology of Markov's Analysis. *Dependable Systems, Services and Technologies (DESSERT'2024)* : Proc. 14th IEEE Int. Conf., Athens, Greece, 2024. P. 1–8. DOI: 10.1109/DESSERT65323.2024.11122169.

11. Горборуков В. В., Приходнюк В. В., Франчук О. В. Алгоритм конкурентної нормалізації в системі рейтингового оцінювання інтелектуальних досягнень. *Наукові записки Малої академії наук України*. 2022. № 1 (23). С. 3–12. DOI: 10.51707/2618-0529-2022-23-01.

12. Stryzhak O., Horborukov V., Prychodniuk V., Franchuk O., Chepkov R. Decision-making System Based on The Ontology of The Choice Problem. *Journal of Physics: Conference Series*. 2021. Vol. 1828, no. 1. P. 012007. DOI: 10.1088/1742-6596/1828/1/012007.

13. Інформаційно-аналітична система підтримки прийняття рішень «ПОЛІЕДР-Альтернатива». URL: <https://e-devel.ulif.org.ua/sppr-int> (дата звернення: 08.04.2026).

РОЗДІЛ 3. РОЗРОБЛЕННЯ ТА ВЕРИФІКАЦІЯ МУЛЬТИФРАГМЕНТНИХ МАРКОВСЬКИХ МОДЕЛЕЙ ІСД ПРОМИСЛОВОГО ОБЛАДНАННЯ З ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ І ЦИФРОВИХ ДВІЙНИКІВ ІЗ ДОНАВЧАННЯМ

У попередньому розділі було розроблено класифікатор та синтезовано комплекс базових моделей (БМ) у вигляді структурних графів станів і переходів. Однак для кількісного оцінювання впливу процесів донавчання цифрового двійника на експлуатаційні властивості системи необхідно здійснити перехід від структурного до математичного моделювання. Головною метою даного розділу є розгортання БМ у повноцінні мультифрагментні марковські моделі (МФММ), формування відповідних систем диференціальних рівнянь Колмогорова-Чепмена та проведення симуляційного моделювання для підтвердження гіпотези щодо ефективності впровадження контуру донавчання. Базова гіпотеза дослідження полягає у тому, що впровадження контуру донавчання інтелектуальної системи діагностування формує експлуатаційний компроміс: стани донавчання, в які переходить система після виявлення відмов, потребують часу і призводять до тимчасового просідання функції готовності на початкових етапах, однак у кінцевому підсумку стаціонарне значення готовності зростає завдяки безперервному підвищенню достовірності діагностування і зменшенню часу перебування об'єкта в стані прихованої відмови.

3.1 Послідовність розроблення та дослідження МФММ

Перехід від структурних базових моделей (розроблених у Розділі 2) до кількісного оцінювання готовності потребує застосування адекватного математичного апарату. Особливістю інтелектуальних систем діагностування із донавчанням є наявність ефекту математичної «жорсткості» систем диференціальних рівнянь, що виникає через різницю на кілька порядків між інтенсивностями рідкісних відмов (повільні події) та швидкостями донавчання

(швидкі події). Крім того, при моделюванні складного обладнання марковські моделі характеризуються надмірним зростанням розмірності простору станів, що суттєво ускладнює проведення обчислень. Для вирішення цих проблем можуть допомогти різні методології стохастичного моделювання та відповідні інструменти.

Перед застосуванням методологій стохастичного моделювання необхідно формалізувати математичний апарат перетворення базових моделей у розрахунковий комплекс. Формування комплексу здійснюється шляхом математичного перетворення порівняльної тріади базових моделей $Tr(BM_i)$ (2.12) у розрахункові тріади МФММ за наступною послідовністю.

1. Статична модель (без донавчання) МФММ_{stat}: формується шляхом математичної редукції стану донавчання (Sl) з простору станів базової моделі BM_i. Усі переходи, що вели до Sl, перенаправляються до працездатного стану (Su).

$$\text{МФММ}_{stat} = BM_i \setminus \{Sl\}. \quad (3.1)$$

2. Ідеалізована модель (абсолютна достовірність, D=1) МФММ_{id}: формується шляхом редукції станів донавчання (Sl) та прихованої відмови (Sh). Переходи перенаправляються до Su та до стану виявленої відмови (Sd) відповідно.

$$\text{МФММ}_{id} = BM_i \setminus \{Sl, Sh\}. \quad (3.2)$$

3. Динамічна мультифрагментна модель (з донавчанням) МФММ_{dyn}: формується шляхом багатовимірною об'єднання сукупності фрагментів. Топологія графа (лінійна, матрична або N-вимірна) детермінується кількістю незалежних векторів донавчання V (кількість вимірів фрагментності). Положення кожного фрагмента у просторі моделі визначається вектором індексів $k = (k_1, k_2, \dots, k_V)$, що належить загальній множині індексів K. Загальний простір станів

моделі визначається виразом:

$$M\Phi MM_{id} = \bigcup_{k \in K} \Phi(k), \quad (3.3)$$

де $\Phi(k)$ – простір станів окремого фрагмента за координатами k . Кожен міжфрагментний перехід по вектору $v \in V$ супроводжується приростом відповідного параметра D достовірності діагностування на фіксований крок ΔD_v . Внутрішня структура кожного фрагмента $\Phi(k)$ не є сталою, а формується шляхом динамічної редукції базової моделі BM_i залежно від досягнення граничних умов:

$$\Phi(k) = BM_i \setminus \{Sl_v \mid D_v(k) = Dmax_v\}, \quad (3.4)$$

де Sl_v – стан донавчання за v -м вектором; $D_v(k)$ – поточне значення достовірності діагностування за v -м напрямком у фрагменті k ; $Dmax_v$ – ліміт донавчання за v -м напрямком. Згідно з правилом (3.4), структура фрагментів змінюється залежно від їх положення у графі.

- а. Транзитні фрагменти: якщо жоден параметр не досяг ліміту, редукція не відбувається ($\Phi(k) = BM_i$).
- б. Граничні фрагменти (крайові): при вичерпанні потенціалу донавчання за певним напрямком, з простору станів вилучається лише відповідний йому стан (Sl_v).
- с. Термінальний фрагмент: фрагмент, у якому всі V векторів досягли ліміту $Dmax$. Усі стани донавчання вилучаються. Якщо при цьому $Dmax=1$ для всіх векторів, з фрагменту додатково вилучаються стани прихованих відмов (Sh) і набуває форми ідеалізованої моделі ($M\Phi MM_{id}$). Якщо $Dmax < 1$, він перетворюється на статичну модель із залишковим рівнем прихованих відмов.

Отже, результатом застосування наведеного алгоритму (формули 3.1–3.4) до концептуальної тріади базових моделей $Tr(BM_i)$ (2.12) є формування

розрахункової порівняльної тріади мультифрагментних марковських моделей $Tr(MФММ_i)$.

$$Tr(MФММ_i) = \langle MФММ_i^{dyn}, MФММ_i^{stat}, MФММ_i^{id} \rangle. \quad (3.5)$$

Сукупність згенерованих розрахункових тріад для всієї досліджуваної множини об'єктів утворює множину мультифрагментних марковських моделей (МФММ).

Завдяки формалізації цього алгоритму остаточно формується друга (математична) складова наукового результату (НР2) – розрахунковий комплекс мультифрагментних марковських моделей готовності відновлювального промислового обладнання систем ІоТ, що дозволяє кількісно оцінити вплив динаміки донавчання цифрового двійника. Основні математичні положення розробленого алгоритму та комплексу МФММ, а також результати застосування методології на основі структурно-автоматної моделі, висвітлено у фахових наукових виданнях та пройшли апробацію на міжнародних конференціях [1-4].

З метою мінімізації обчислювальних похибок та підвищення достовірності результатів оцінювання, безпосередня реалізація і дослідження частини моделей з синтезованого комплексу МФММ здійснюється за двома методологіями стохастичного моделювання: на основі графів станів (state-based) [1] та на основі структурно-автоматних моделей (event-based) [2, 5].

Дослідження ефективності та точності обох підходів при розв'язанні моделей ІСД створює об'єктивні передумови для подальшого розроблення методу перехресної верифікації та валідації, що буде детально розглянуто у підрозділі 3.4. Незважаючи на відмінності у застосованому математичному апараті, загальна логіка дослідження підпорядковується єдиній послідовності, яка складається з шести етапів.

Етап 1. Обрання базової моделі та ідентифікація змінних параметрів.

На цьому етапі з комплексу, розробленого за допомогою класифікатора (підрозділ 2.2), обирається конкретна базова модель відповідно до типу об'єкта та

архітектури засобів контролю. Визначається ключовий параметр, що еволюціонує в процесі функціонування. У межах даного дослідження цільовим параметром є показник достовірності діагностування D (ймовірність успішного виявлення відмови), який змінюється внаслідок роботи алгоритмів машинного навчання цифрового двійника.

Етап 2. Структурно-математичний синтез моделі. На цьому етапі для обраної базової моделі генерується розрахункова тріада МФММ згідно з розробленим математичним алгоритмом (формули 3.1–3.5). Залежно від обраного апарату, подальша практична реалізація цього етапу має відмінності у послідовності дій.

1. **Методологія на основі станів.** Базова модель описує поведінку системи лише до моменту набуття нового досвіду (в межах одного стаціонарного стану параметрів). На цьому етапі здійснюється розгортання БМ у $\text{МФММ}_i^{\text{dyn}}$ за правилами описаними вище та з врахування виконанням таких кроків.

а. Визначення функції та кроку приросту параметра D . Згідно з припущень (розділ 2.3), для всіх моделювань буде використовуватись лінійний приріст із фіксованим кроком ΔD .

б. Визначення глибини фрагментності – кількості ітерацій донавчання (фрагментів), після яких параметр D досягає свого асимптотичного максимуму D_{max} .

с. Синтезу підсумкового розгорнутого графа, де стаціонарні фрагменти з'єднуються між собою орієнтованими переходами через стани налаштування/навчання (SI). Залежно від наявності маршрутизації досвіду утворюються моделі з лінійною, матричною або N -вимірною топологією;

2. **Методологія на основі структурно-автоматної моделі** має наступні етапи синтезу моделі:

а. формування детальної вербальної моделі об'єкта дослідження (опис усіх подій та ситуацій експлуатаційної поведінки, враховуючи дії оперативного контролю, діагностування та служби обслуговування), де

наявна базова модель з класифікатору буде слугувати джерелом для вербальної моделі і наступних пунктів за методологією;

- b. виділення базових подій (БП) та зведених базових подій (ЗвБП) на основі вербальної моделі;
- c. підготовка та кодування вектора станів;
- d. синтез структурно-автоматної моделі;
- e. автоматизована генерація підсумкового імовірнісного графа експлуатаційної поведінки;
- f. верифікація імовірнісного графа експлуатаційної поведінки.

Етап 3. Параметричний синтез (Параметризація). Для проведення кількісної симуляції визначаються фізично обґрунтовані діапазони значень інтенсивностей переходів. Специфікою МФММ ІСД є те, що вони належать до класу «жорстких» систем (stiff systems), оскільки події в них відбуваються на суттєво різних часових шкалах. Параметри розділяються на наступні групи.

1. Повільні події: інтенсивність виникнення відмов обладнання (λ_d), яка зазвичай становить $10^{-4} \dots 10^{-5}$ 1/год.
2. Швидкі події: інтенсивності відновлення (μ_i) та виконання процедур оперативного донавчання (λ_l), порядок яких становить $10^{-1} \dots 10^{-2}$ 1/год.
3. Ймовірнісні параметри контролю переходів: сюди відносяться параметри, які розподіляють ймовірність переходів у той чи інший стан при виникненні події. Наприклад P_{ld} – ймовірність, що буде необхідне донавчання після виявлення відмови вбудованими засобами.
4. Еволюційні параметри: початкова достовірність контролю (D_{start}), крок її приросту (ΔD) та ліміт навчання ($D_{max} \leq 1$).

Етап 4. Формування сценаріїв симуляційного моделювання. Моделювання виконується за набором заздалегідь підготовлених сценаріїв. Кожен сценарій – це певний унікальний набір параметрів та симуляція цих параметрів за принципом порівняльної тріади (сформульованим у підрозділі 2.2.3). Для початку береться базовий (baseline) набір параметрів (наприклад медіанне значення) й виконується дослідження поведінки моделі з цим набором. Далі,

шляхом почергової зміни певних параметрів (чи кількох параметрів), можна дослідити зміну поведінки моделі і оцінити вплив того чи іншого параметру на результуюче значення готовності (аналіз чутливості). Причому для кожного сценарію для кожної МФММ розраховуються три базові підсценарії (порівняльна тріада):

1. static (статичний): система без донавчання ($D = \text{const} = D_{\text{start}}$);
2. relearning (динамічний): система з функцією донавчання цифрового двійника ($D \rightarrow D_{\text{max}}$);
3. ideal (ідеалізований): теоретична межа системи з абсолютною достовірністю ($D = 1$).

Етап 5. Математичне розв'язання та симуляція. Розрахунок ймовірностей перебування системи у працездатних станах $P_i(t)$ у часі здійснюється із застосуванням інструментів, релевантних обраній методології.

1. *Методологія на основі станів.* На основі синтезованого розгорнутого графа МФММ формується система лінійних диференціальних рівнянь Колмогорова-Чепмена. Вирішення цієї системи диференціальних рівнянь в рамках даних моделей виконується за допомогою чисельних солверів та відповідних бібліотек для марковського моделювання і розв'язання диференціальних рівнянь у мовах R та Python. Підсумковим результатом обчислень є знаходження функції експлуатаційної готовності $A(t)$ як суми ймовірностей перебування об'єкта в усіх працездатних станах (сума всіх станів S_u по різних фрагментах).

2. *Методологія на основі структурно-автоматної моделі.* На основі структурно-автоматної моделі формується система диференціальних рівнянь Колмогорова-Чепмена, розв'язання якої здійснюється за допомогою спеціалізованого програмного комплексу ASNA.

Етап 6. Аналіз результатів та верифікація гіпотези. Отримані масиви даних та графіки функцій готовності $A(t)$ аналізуються за допомогою спеціалізованих метрик, розроблених для оцінювання систем із донавчанням.

1. A_{steady} (Стаціонарна готовність): кінцеве значення коефіцієнта готовності після завершення всіх циклів оновлення (вихід функції на плато). Слід зазначити, що конкретно цей параметр підраховується для кожного підсценарію порівняльної тріади для більш змістовної порівняльної оцінки.

2. A_{min} (Мінімальна готовність): глибина максимального «просідання» функції готовності на початкових етапах життєвого циклу через втрати часу на перебування системи у стані навчання (SI).

3. T_{loss} (Час «навчального боргу» / Time-to-breakeven): момент часу (точка перетину кривих динамічного та статичного сценаріїв), після якого готовність системи з донавчанням починає стабільно перевищувати готовність консервативної системи без донавчання.

На цьому етапі не лише підтверджується або спростовується базова гіпотеза щодо наявності довгострокового виграшу від донавчання, але й приймається комплексне рішення щодо загальної доцільності його впровадження. З одного боку, виграш у стаціонарній готовності (ΔA_{steady}) має бути достатньо вагомим, щоб виправдати організаційні та економічні витрати на розгортання інфраструктури цифрового двійника. З іншого боку, короткострокові втрати не повинні порушувати критичних обмежень експлуатації: просідання A_{min} не має опускатися нижче допустимого мінімуму, а час навчального боргу T_{loss} не повинен бути надто довгим або перевищувати розрахунковий термін окупності. Якщо розрахунки показують, що потенційний програш є занадто глибоким чи тривалим, а довгостроковий виграш мінімальним (наприклад, коли потенціал D_{max} близький до початкового D_{start}), впровадження такого роду оперативного донавчання визнається недоцільним.

3.2 Розроблення та дослідження МФММ1

Для початку кількісних досліджень та перевірки гіпотези щодо впливу донавчання на показники готовності обрано Базову модель 1 (БМ1). Ця модель є фундаментальною, має відносно невелику кількість станів та ілюструє базову

поведінку адаптивної системи діагностування. Також було прийнято рішення розпочати апробацію з першої методології – методології на основі станів. Аналогічне дослідження було апробовано в рамках конференції і статті DESSERT 2023 [1].

3.2.1 Припущення

Оскільки математична модель передбачає спрощене представлення об'єкта реального світу, перед її розробленням визначено низку додаткових до загальних припущень з розділу 2.3:

1. розглядається узагальнений IoT-компонент або базовий вузол складного обладнання, що працює у неперервному часі;
2. процес донавчання засобів діагностування цифрового двійника вимагає певних витрат часу, протягом якого основне обладнання перебуває у стані простою та не виконує цільові функції;
3. вплив засобів контролю на систему вважається ідеалізованим і розглядається лише через ймовірність успішного і неуспішного (прихованого) діагностування відмови (параметр D);
4. вихід об'єкта зі стану прихованої відмови до стану виявленої відмови відбувається за тією ж інтенсивністю, що і базовий потік відмов і балансується тим самим параметром достовірності діагностування D (тобто інтенсивність переходу $Sh \rightarrow Sd$ дорівнює $D\lambda d$).

3.2.2 Розроблення

На етапі структурного синтезу БМ1 розгортається у мультифрагментну марковську модель МФММ1 (згідно формул 3.3-3.4). Кожен стаціонарний фрагмент МФММ1 описує поведінку системи за незмінного рівня навчання і містить чотири базові стани:

1. S_u (U_p state) – працездатний стан;

2. S_d (Detected failure) – стан простою внаслідок виявленої відмови та проведення відновлення;
3. S_h (Hidden failure) – стан прихованої відмови, в якому обладнання не виконує цільові функції, але відмова ще не ідентифікована;
4. S_l (Learning state) – стан донавчання діагностичних засобів цифрового двійника.

У МФММ1(рис. 3.1) єдиним змінним параметром виступає достовірність діагностування (D). Оскільки змінюється лише один параметр, формується лінійна топологія (ланцюгове з'єднання фрагментів). Зростання параметра моделюється лінійною функцією із заданим кроком ΔD : $D_i = D_{i-1} + \Delta D$, за умови $D_i \leq D_{\max}$. D_{start} – початковий рівень достовірності діагностування (у першому фрагменті); $D_{\max}$ – кінцевий рівень достовірності діагностування (у останньому фрагменті); ΔD – фіксований рівномірний крок приросту параметра після кожного циклу донавчання; i – порядковий номер фрагмента ($i \in \{1, 2, \dots, N\}$). Перехід системи з i -го до $(i+1)$ -го фрагмента відбувається через донавчання ($S_l \rightarrow S_u, i+1$).

Для наочної генералізації та дослідження загальної динаміки процесу розглядається МФММ1, що складається з трьох фрагментів ($N=3$). Ця кількість є мінімально необхідною і достатньою для візуалізації та математичного аналізу повного життєвого циклу адаптивної системи. Граф станів та переходів такої лінійної МФММ1 наведено на рисунку 3.1. Назви параметрів та станів описані у таблиці 3.1.

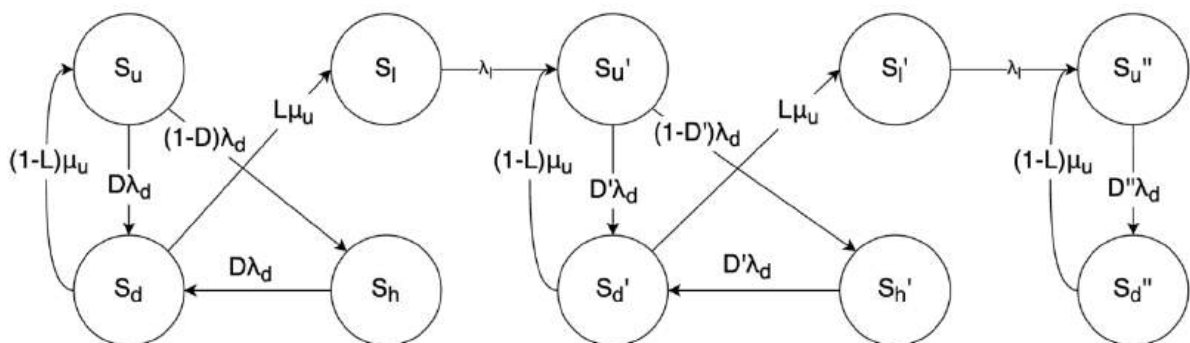


Рисунок 3.1 – Граф станів та переходів лінійної МФММ1 з 3 фрагментами

Важливою структурною особливістю розробленої моделі є те, що третій (останній) фрагмент демонструє випадок досягнення системою ідеальної достовірності діагностування ($D''=D_{\max}=1$). За таких умов ймовірність пропуску відмови дорівнює нулю. Як наслідок, стани прихованої відмови S_h та подальшого донавчання S_l стають надлишковими, і структура останнього фрагмента редукується до двох базових станів (S_u та S_d).

Таблиця 3.1 – Опис параметрів та станів МФММ1

№	Стани та параметри	Опис
1	S_u	Працездатний стан
2	S_d	Стан відмови
3	S_h	Стан прихованої відмови
4	S_l	Стан донавчання
5	λ_d	Інтенсивність відмов
6	λ_l	Інтенсивність донавчання
7	μ_u	Інтенсивність відновлення
8	D	Достовірність діагностування
9	L	Ймовірність необхідності донавчання

Показник готовності A показує відсоток ситуацій (часу), коли система знаходиться в робочому стані. Він розраховується як сума ймовірностей перебування в кожному з робочих станів. У цьому випадку, маючи приклад графіка з 2 можливими успішними ітераціями перенавчання, ми маємо наступну формулу.

$$A = S_u + S'_u + S''_u. \quad (3.6)$$

В свою чергу математично функціонування МФММ1 у неперервному часі описується системою диференціальних рівнянь Колмогорова-Чепмена. Диференціальні рівняння Колмогорова-Чапмана для дискретно-неперервних марковських ланцюгів використовуються для опису та аналізу динамічної поведінки системи в часі. Це дозволяє визначити ймовірність перебування системи в певному стані в будь-який момент часу. Загальний вигляд системи диференціальних рівнянь:

$$\frac{dP(t)}{dt} = P(t)Q, \quad (3.7)$$

де $P(t)$ – матриця ймовірності переходів, елемент $P_{ij}(t)$ представляє ймовірність того, що система перейде зі стану i до стану j за час t . Q (3.8) – матриця інтенсивностей переходів, де елемент Q_{ij} представляє інтенсивність переходу зі стану i до стану j . Результати розв'язання диференціальних рівнянь (3.7), а саме ймовірність перебування в кожному з можливих станів, використовуються в кінцевій формулі (3.6) для обчислення показника готовності (A).

Розглянемо детальніше формування матриці інтенсивностей переходів. Враховуючи редукцію останнього фрагмента, загальна кількість станів для МФММ1 з трьох фрагментів ($N=3$) становить 10 (по 4 стани у перших двох фрагментах та 2 стани в останньому). Відповідно, вектор ймовірностей станів $P(t)$ має наступний вигляд.

$$P(t) = [Pu(t), Pd(t), Ph(t), Pl(t), Pu'(t), Pd'(t), Ph'(t), Pl'(t), Pu''(t), Pd''(t)]. \quad (3.8)$$

Матриця інтенсивностей переходів Q розмірністю 10×10 має наступний вигляд:

$$Q = \begin{bmatrix} -\lambda_d & D_1\lambda_d & (1-D_1)\lambda_d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ (1-L)\mu_u & -\mu_u & 0 & L\mu_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D_1\lambda_d & -D_1\lambda_d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\lambda_l & \lambda_l & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_d & D_2\lambda_d & (1-D_2)\lambda_d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-L)\mu_u & -\mu_u & 0 & L\mu_u & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & D_2\lambda_d & -D_2\lambda_d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_l & \lambda_l & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_d & \lambda_d & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_u & -\mu_u \end{bmatrix}, \quad (3.9)$$

де діагональні елементи матриці q_{ii} дорівнюють взятій з протилежним знаком сумі інтенсивностей всіх вихідних потоків з відповідного стану ($q_{ii} = \sum_{j \neq i} q_{ij}$); D_1 та D_2 – показники достовірності діагностування у першому та другому фрагментах відповідно ($D_1 = D_{start}$, $D_2 = D_{start} + \Delta D$).

Переходи між фрагментами локалізовані в елементах $q_{4,5}$ та $q_{8,9}$ (інтенсивність успішного донавчання λ_l), які пов'язують стани $Sl \rightarrow Su'$ та $Sl' \rightarrow Su''$. Нижня права підматриця 2×2 відображає редукований третій фрагмент із ідеальною діагностикою.

Для розв'язання системи (3.7) приймається, що в початковий момент часу ($t=0$) система є повністю працездатною ($P_u(0)=1$), а ймовірності перебування в усіх інших станах дорівнюють нулю.

3.2.3 Дослідження

Метою етапу дослідження є аналіз впливу різних параметрів процесу донавчання на загальну готовність системи (sensitivity analysis, аналіз чутливості) та емпірична перевірка базової гіпотези, сформульованої на початку 3-го розділу.

Параметризація моделі. Для проведення симуляційного моделювання МФММ1 було визначено фізично обґрунтовані діапазони значень інтенсивностей переходів, що відповідають типовим експлуатаційним характеристикам базових IoT-компонентів промислових систем. Базові значення параметрів для формування еталонного (baseline) сценарію наведено у таблиці 3.2.

Таблиця 3.2 – Обґрунтування параметрів для моделювання МФММ1

Параметри	Опис
λ_d	10^{-4} – це значення інтенсивності відмов електричних розподільних шаф, 10^{-6} – для більш критичних компонентів, як панелей управління для АЕС. Базовим значенням є 10^{-4}
λ_l	Від 0.1 до 1: час донавчання і застосування його змін (з переходом далі в працездатний стан) займає від 1 години до 10 годин в залежності від складності змін. Базовим значенням є 0.25
μ_u	Інтенсивність відновлення зі значенням 2 описує ситуацію, коли компонент, наприклад вузол в розподільному шафі, може бути замінений за 30 хвилин (швидкий ремонт шляхом вузлової заміни). Коефіцієнт 10^{-2} означає, що система буде непрацездатною протягом 4 днів через критичну помилку, наприклад ремонт трансформатора після його загоряння. Базовим значенням є 0.25
D	Звичайна достовірність діагностування у промисловому чи критичному обладнанні коливається від 0.7 до 0.95. Базовим значенням є 0.85
L	В цій моделі ми виконуємо донавчання лише зі стану виявленої відмови. В наступних моделях це буде розширено, однак маємо обрати середній діапазон ймовірності донавчання в незалежності від причини переходу в Sd. Значення має бути за порядком співставно до D. Від 0.1 до 0.95. Базовим значенням є 0.1

Інструментарій та програмна реалізація. Програмна реалізація моделі та обчислення ймовірностей станів виконувалися у середовищі статистичного програмування R. Вибір цього інструментарію зумовлений його аналітичними можливостями та наявністю спеціалізованих бібліотек для стохастичного моделювання. Основним обчислювальним ядром виступив пакет *markovchain*. Розроблений скрипт для МФММ1 генерував матрицю інтенсивностей переходів Q на основі заданої кількості фрагментів N, вказаних значень для параметрів з таблиці 3.2 та кроку приросту ΔD . За допомогою вбудованих функцій пакета *markovchain* та чисельних методів розв'язання диференціальних рівнянь обчислювалися ймовірності перебування системи у кожному зі станів $P_k(t)$ для неперервного часу t. Для автоматизованої побудови та візуалізації графів

переходів згенерованих мультифрагментних моделей використовувався пакет *diagram*.

Метрики оцінювання та послідовність експериментів. Для оцінювання результатів моделювання та порівняльного аналізу сценаріїв використовуються ключові показники експлуатаційної надійності, частково обґрунтовані у підрозділі 1.3.3. Усі вони розглядаються в контексті адаптивної моделі з донавчанням і порівнюються з аналогічною моделлю без донавчання. До зведеного переліку метрик належать:

1. A_{steady} (в оригінальному дослідженні StdAl [1]) – стаціонарна готовність, на яку виходить система після завершення всіх циклів донавчання;
2. A_{min} (в оригінальному дослідженні MinAl [1]) – мінімальне (найгірше) значення функції готовності, що виникає на початковому етапі життєвого циклу адаптивної системи внаслідок її простоїв у стані донавчання SI;
3. LoA_{max} (Loss of Availability) – максимальна втрата готовності. Розраховується як найбільша різниця між кривою статичної моделі ($A_{\text{stat}}(t)$) без навчання та моделі з донавчанням ($A(t)$) у момент максимального просідання;
4. T_{loss} – час, протягом якого система з донавчанням функціонує зі зниженим рівнем готовності до моменту перетину її графіка з базовою статичною кривою;
5. T_{std} – час, необхідний системі для подолання 95 % діапазону свого відновлення після початкового просідання – $A(T_{\text{std}}) \geq A_{\text{min}} + 0.995(A_{\text{steady}} - A_{\text{min}})$.

Дослідження. Для перевірки того, як зміна кожного окремого параметра впливає на загальну динаміку та ключові метрики системи, було проведено комплекс експериментів. Під час кожного тесту ізольовано змінювався один із параметрів (в межах діапазонів, визначених у табл. 3.1), тоді як інші залишалися на базовому рівні. Ключові результати розрахунків зведено до єдиної порівняльної таблиці 3.3.

Відправною точкою (baseline) для всіх експериментів є базовий сценарій (№1), що базується на початкових налаштуваннях параметрів, наведених у

таблиці 3.2. На рисунку 3.2 наведено лінійні графіки динаміки функції готовності $A(t)$ для цього випадку.

Таблиця 3.3 – Результати симуляцій з почерговою зміною параметрів відносно базового сценарію (baseline)

Змінений параметр	Зміни відсутні	λd	λl	μu	L	Dstart
Значення зміненого параметру відносно базового	-	$10^{-4} \rightarrow 10^{-3}$	$0.25 \rightarrow 10^{-2}$	$0.25 \rightarrow 10^{-2}$	$0.1 \rightarrow 0.7$	$0.85 \rightarrow 0.95$
№ сценарію	1	2	3	4	5	6
Asteady	0.9996	0.99601	0.9996	0.9999	0.9996	0.9996
Amin	0.87006	0.52165	0.86929	0.86267	0.91381	0.95691
Tloss	[3-1057]	[3-334]	[3-5570]	[3-1242]	[3-1058]	[3-1850]
LoAmax	0.02035	0.16047	0.01958	0.01983	0.06410	0.00727
Tstd	$\sim 8e+5$	$\sim 2.4e+5$	$\sim 8e+5$	$\sim 8e+5$	$\sim 1.3e+5$	$\sim 7.8e+5$

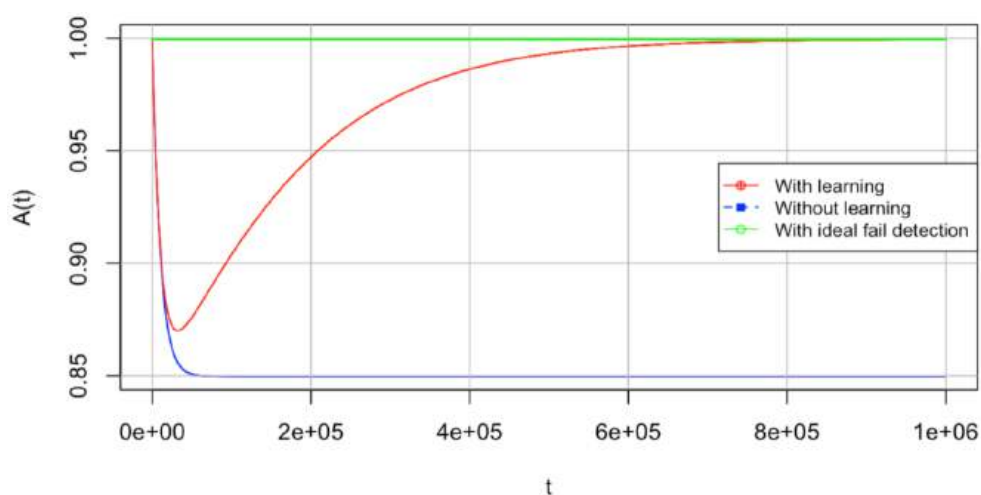


Рисунок 3.2 – Динаміка функцій готовності МФММ1 для базових параметрів (базовий сценарій)

Збільшений масштаб на початку для візуалізації перших годин динаміки функцій готовності зображено на рисунку 3.3, де чітко зафіксовано зону програшу в готовності для системи з навчанням.

Як видно з графіків базового сценарію, базова гіпотеза дослідження знаходить математичне підтвердження. Адаптивна система (з донавчанням) на

початку експлуатації демонструє падіння готовності порівняно зі статичною системою, але виграє в стаціонарному значенні. Рисунок 3.3 ілюструє першу складову цього компромісу – так звану «зону навчального боргу». На першому етапі експлуатації готовність системи з донавчанням падає швидше, ніж у статичної системи. Це пояснюється тим, що ідентифікація нових класів відмов та особливо застосування результатів локального донавчання потребують додаткового часу на налаштування алгоритмів ЦД, що генерує тимчасові простой обладнання.

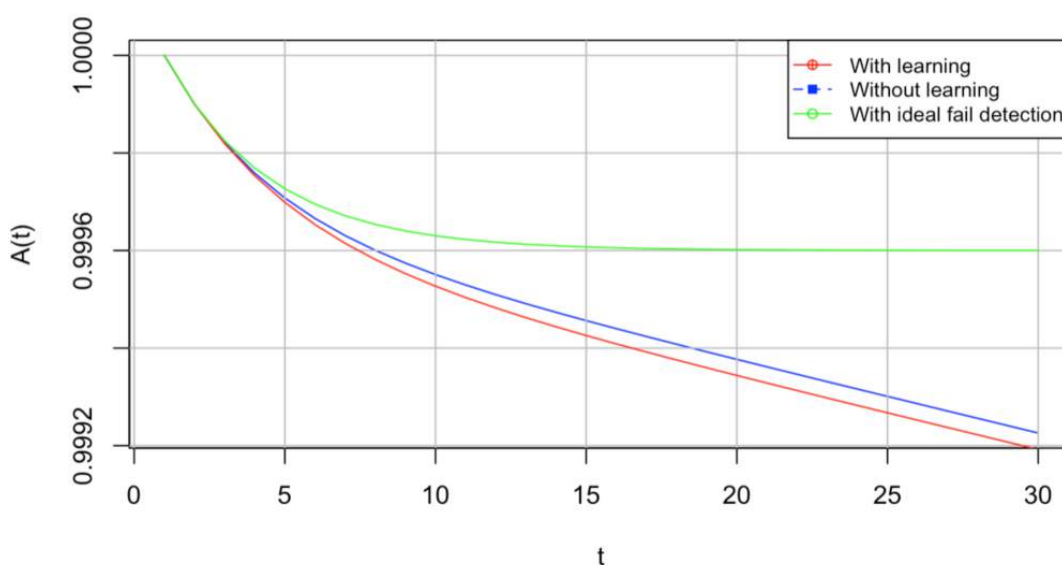


Рисунок 3.3 – Наближення у масштабі на початок функцій готовності МФММ1 для базових параметрів (базовий сценарій)

Такий початковий програш є припустимим для тих систем IoT, де короткочасне зниження готовності на етапі введення в експлуатацію не призводить до катастрофічних наслідків, або якщо процес донавчання може бути суміщений із плановим технічним обслуговуванням. Якщо ж вимоги до безперервності технологічного процесу є жорсткими (наприклад, регламент не допускає сумарного простою більше однієї години на місяць), використання режимів онлайн-донавчання із зупинкою обладнання буде неприпустимим. Однак, за умови припустимості цих початкових витрат, у точці перетину кривих (Tloss) система з донавчанням компенсує втрати і наздоганяє статичну модель.

Після цієї точки розпочинається етап виграшу (друга складова компромісу): адаптована ІСД функціонує з вищою достовірністю (D), має менше прихованих відмов, що у довгостроковій перспективі дає вище стаціонарне значення готовності (A_{steady}).

Подальший аналіз чутливості МФММ1 до зміни параметрів (згідно з даними зведеної таблиці 3.3) дозволив виявити низку закономірностей експлуатаційної поведінки системи за різних умов функціонування.

Сценарій №2: Вплив інтенсивності відмов (λ_d). Збільшення інтенсивності відмов λ_d призводить до того, що об'єкт частіше виходить з ладу і, відповідно, частіше потрапляє до стану донавчання. З одного боку, це значно прискорює процес еволюції цифрового двійника – час перехідного процесу T_{std} скорочується. З іншого боку, часті зупинки на навчання і відмови в цілому спричиняють суттєве погіршення початкової готовності (глибоке просідання A_{min}) та призводять до незначного зниження кінцевої стаціонарної готовності A_{steady} .

Сценарій №3: Вплив швидкості донавчання (λ_l). Зменшення швидкості виконання процедури донавчання λ_l практично не впливає на стаціонарну готовність A_{steady} та максимальне просідання A_{min} . Проте, повільне донавчання критично збільшує час програшу у готовності T_{loss} (зафіксовано зростання майже у 5 разів порівняно з базовим сценарієм), що суттєво відтерміновує момент, коли адаптивна система стає вигіднішою за статичну.

Сценарій №4: Вплив інтенсивності відновлення (μ). Зменшення інтенсивності відновлення μ в діапазоні $[1 - 0.01]$ (що фізично означає збільшення тривалості ремонтних робіт) знижує функцію готовності $A(t)$ приблизно на 1% у всіх точках графіка. Крім того, триваліші ремонти призводять до збільшення інтервалу втрат T_{loss} на 20%.

Сценарій №5: Вплив ймовірності ініціалізації навчання (L). Збільшення ймовірності переходу до навчання після відновлення L з базового значення 0.1 до 0.7 демонструє виражений позитивний ефект. Застосування такої інтенсивної стратегії збору даних покращує найгірший показник готовності A_{min} на 4,5% та

прискорює вихід системи на стаціонарний режим (T_{std}) у 6 разів. Параметр L суттєво впливає на швидкість стабілізації моделі, що підтверджується динамікою кривих на рисунку 3.4.

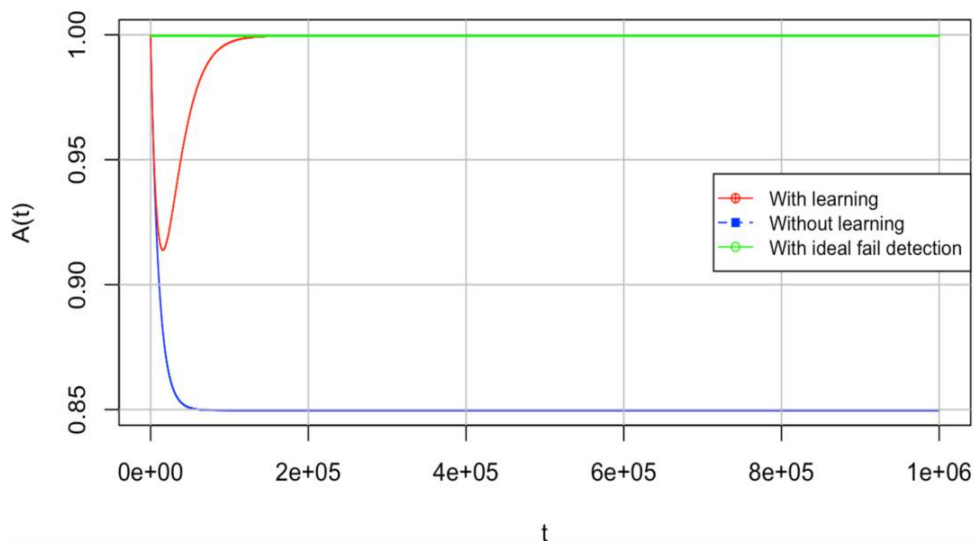


Рисунок 3.4 – Динаміка функцій готовності МФММ1 для частого донавчання (сценарій №5)

Сценарій №6: (D_{start}). Оскільки достовірність діагностування D є єдиним еволюційним параметром моделі, у сценарії №6 досліджувався вплив її початкового значення D_{start} в діапазоні $[0.85 - 0.95]$. Результати показують пряму залежність: чим вищим є D_{start} , тим ближче всі криві функції готовності наближаються до ідеалізованого показника A_{steady}^{id} . Збільшення початкової достовірності на 0.1 підвищує всі ключові метрики готовності майже пропорційно та суттєво зменшує глибину початкових втрат.

Додаткові структурні експерименти. Окремо досліджено параметри, що безпосередньо змінюють топологію та розмірність МФММ1 (тому вони не увійшли до зведеної таблиці).

1. Темп зростання D : Моделювання еволюції через більшу кількість ітерацій (10 фрагментів замість 3) довело, що чим меншим є крок приросту ΔD на кожному етапі, тим меншими стають час перехідного процесу T_{std} та глибина просідання A_{min} .

2. Ліміт донавчання ($D_{\max}$): Чим меншим є максимально досяжне значення достовірності, тим більшими є втрати від навчання (суттєво зростає T_{loss} , падають $A_{\min}$ та A_{steady}). За умови $D_{\max} < 1.0$ останній фрагмент моделі не підлягає редукції, зберігаючи переходи до станів прихованої відмови.

3.2.4 Попередні результати

У результаті симуляційного моделювання МФММ1 базову гіпотезу дослідження підтверджено. Аналіз еталонного сценарію показав, що адаптивна система на початкових етапах експлуатації демонструє тимчасове просідання функції готовності (досягаючи локального мінімуму $A_{\min}$) через простой обладнання під час донавчання. Однак, у міру зростання достовірності контролю, після проходження часу просідання T_{loss} , крива готовності перетинає статичний показник і виходить на вищий стаціонарний рівень A_{steady} , ефективно компенсуючи початкові витрати часу.

Комплексний аналіз чутливості виявив закономірності впливу експлуатаційних параметрів. Збільшення інтенсивності відмов (λ_d) або ймовірності ініціалізації навчання (L) прискорює еволюцію цифрового двійника (скорочується час перехідного процесу T_{std}), але поглиблює початкові втрати $A_{\min}$. Натомість низька швидкість донавчання (λ_l) або тривалі ремонти (μ) сильно розтягують інтервал втрат T_{loss} . Водночас покращення початкової достовірності (D_{start}) пропорційно підвищує всі ключові метрики системи.

Параметри, що безпосередньо змінюють топологію МФММ1, визначають доцільність самого підходу. Доведено, що менший крок приросту достовірності D пом'якшує просідання $A_{\min}$. Визначальним фактором є ліміт донавчання $D_{\max}$: за умови $D_{\max} < 1.0$ останній фрагмент моделі не редукується, що унеможливорює досягнення ідеальної готовності. Якщо алгоритми мають низький потенціал ($D_{\max}$ близьке до D_{start}), невеликий приріст достовірності діагностування не перекриває збитки від простоїв, тому від оперативного донавчання доцільно відмовитися взагалі.

3.3 Розроблення та дослідження МФММ2

Незважаючи на те, що досліджена МФММ1 успішно підтвердила базову гіпотезу щодо довгострокової ефективності донавчання, вона є концептуально спрощеною і не повною мірою відображає експлуатаційні особливості складного промислового обладнання. Зокрема, у цій моделі не враховано наявність служб технічного обслуговування та ремонту (СТОіР) для проведення планових і аварійних робіт, а процес виявлення прихованих відмов не має окремого показника (відповідна інтенсивність жорстко прив'язана до базового потоку відмов λ_d , що унеможлиблює її незалежне дослідження). Для усунення цих недоліків пропонується перехід до більш комплексної моделі МФММ2. Її розроблення та дослідження здійснюється із застосуванням другої із запропонованих методологій – на основі структурно-автоматних моделей (САМ) [2, 5], що дозволяє формалізувати генерацію простору станів. Аналогічне дослідження було апробовано в рамках конференції і статті DESSERT 2024 [2].

3.3.1 Припущення

Для розроблення структурно-автоматної моделі (САМ) експлуатаційної поведінки складного промислового обладнання із залученням служб технічного обслуговування і ремонту (СТОіР) прийнято такі базові припущення:

1. немає обмежень на кількість ремонтів системи;
2. ручне діагностування службою СТОіР виконується періодично, але тільки після прихованої відмови, тобто в даній моделі це механізм виходу із стану прихованої відмови;
3. прихована відмова виявляється тільки СТОіР і у 2х випадках:
 - a. під час наступного запланованого ручного діагностування;
 - b. у випадку виникнення іншої неприхованої відмови і запиту на ремонт від СТОіР;

4. під час прихованої відмови система знаходиться у непрацездатному стані (простої), але спостереження не завершуються;
5. система може перенавчатися тільки після виявлення відмови;
6. система під час навчання тільки покращує достовірність діагностування;
7. під час навчання і конфігурації системи діагностування, промислове обладнання знаходиться в простої;
8. зміна конфігурації після навчання завжди успішна;
9. сигнал завершення конфігурації завжди успішний.

3.3.2 Розроблення вербальної моделі та формування базових подій

Методологія стохастичного моделювання віртуальної експлуатаційної надійнісної поведінки інформаційних систем на основі структурно-автоматної моделі [2, 5] дозволяє уникнути прямого ручного синтезу та моделювання громіздких марковських графів шляхом задання початкових умов та логічних правил переходів, об'єднання спільних базових подій у спільні вирази структурно-автоматної моделі, яка вже в спеціалізованому програмному забезпеченні ASNA розгортається та використовується для створення і вирішення диференціальних рівнянь Колмогорова-Чепмена. модель генерується автоматично. Процес розроблення МФММ2 згідно з цією методологією розділено має наступні послідовні етапи: формування вербальної моделі, визначення параметрів і базових подій, формування вектор станів ймовірнісного графа, розроблення опорного ймовірнісного графа, синтез структурно-автоматної моделі, верифікація. Почнемо з вербальної моделі.

Вербальна модель ІСД з донавчанням за ВМ2. Вербальна модель показує всі події та ситуації, що відбуваються під час роботи діагностичної системи з функцією повторного навчання складного обладнання. Моделюється ситуація діагностичної системи з повторним донавчанням, коли система може виконати лише 2 ітерації донавчання для підвищення достовірності діагностування.

Початок «спостережень» за роботою системи: промислове обладнання почало свою роботу, система діагностування налаштована і проводить періодичні тести, а служба СТОіР готова як виїжджати на виклики ремонту, так і на періодичне ручне діагностування.

Перша очікувана пара подій: подія 1.1 «початок роботи складного обладнання» – подія 1.2 «втрата працездатності».

В результаті відмови актуальними далі є друга очікувана пара подій: подія 2.1 «початок процедури контролю» – подія 2.2 «завершення процедури контролю». Також через відмову вважаємо, що система переходить в стан простою.

Друга очікувана пара подій: подія 2.1 «початок процедури контролю» – подія 2.2 «завершення процедури контролю». Контроль дозволяє розпізнати наявність певної відмови в цілому, без аналізу конкретної причини. Навіть такого сигналу може бути достатньо щоб виконати відновлення. СТОіР отримує сигнал для проведення додаткової діагностики на місці.

Варіант для очікуваної події 2.2.1: завершення процедури контролю є успішним. Наявність відмови визначено. СТОіР вже отримує сигнал, але можна спробувати покращити ситуацію та локалізувати причину помилки за допомогою діагностики. Тому тут є актуальною третя очікувана пара подій – подія 3.1 «початок процедури вбудованої діагностики» та подія 3.2 «завершення процедури вбудованої діагностики».

Варіант для очікуваної події 2.2.2: завершення процедури контролю не є успішним. Маємо приховану відмову і система продовжує знаходитись в стані простою. Єдиний вихід – виявлення відмови СТОіР в результаті запланованого ТО (технічне обслуговування). Таким чином, для варіанту 2.2.2 далі актуальною є четверта пара подій – подія 4.1 «початок запланованого ТО службою СТОіР» та подія 4.2 «завершення запланованого ТО службою СТОіР».

Третя очікувана пара подій: подія 3.1 «початок процедури вбудованої діагностики» та подія 3.2 «завершення процедури вбудованої діагностики»

Варіант для очікуваної події 3.2.1: завершення процедури вбудованої діагностики є успішним. Це означає, що причину відмови визначено. Це дає

більше інформації СТОіР, тому вони вже можуть оптимізувати відновлення – наприклад, узяти необхідний модуль для заміни, необхідні інструменти, чи дізнатись про конкретний алгоритм дій для такої ситуації. Це все пришвидшує відновлення. Далі актуальною шоста пара подій – подія 6.1 «початок ремонту службою СТОіР» та подія 6.2 «завершення ремонту службою СТОіР»

Варіант для очікуваної події 3.2.2: завершення процедури вбудованої діагностики не є успішним. Це означає, що ми маємо приховану причину відмови, але все ж наявність відмови була виявлена контролем. Хоча СТОіР не буде знати причини, але вони знають, що щось не так і можуть провести ТО на місці. Таким чином, далі актуальною є п'ята пара подій – подія 5.1 «початок аварійного ТО службою СТОіР» та подія 5.2 «завершення аварійного ТО службою СТОіР».

Четверта очікувана пара подій: подія 4.1 «початок запланованого ТО службою СТОіР» та подія 4.2 «завершення запланованого ТО службою СТОіР».

Зазначимо, що під запланованим ТО мається на увазі комбінація як проведення діагностики так і виконання ремонту, після успішного завершення цих подій система переходить у працездатний стан. А запланованим воно називається, оскільки проводиться періодично за будь-яких обставин. В моделі вважаємо, що заплановане ТО службою СТОіР завжди успішне ($P_{sd} = 1$), тому після події 4.2 «завершення запланованого ТО службою СТОіР» наступною парою очікуваних подій є сьома пара – подія 7.1 «початок навчання процедури вбудованої діагностики» та подія 7.2 «завершення навчання процедури вбудованої діагностики».

П'ята очікувана пара подій: подія 5.1 «початок аварійного ТО службою СТОіР» та подія 5.2 «завершення аварійного ТО службою СТОіР».

Зазначимо, що під аварійним ТО мається на увазі комбінація як проведення діагностики так і виконання ремонту, після успішного завершення цих подій система переходить у працездатний стан. В моделі вважаємо, що аварійне ТО службою СТОіР завжди успішне ($P_{ad} = 1$), тому після події 5.2 «завершення аварійного ТО службою СТОіР» наступною парою очікуваних подій є сьома пара

– подія 7.1 «початок навчання процедури вбудованої діагностики» та подія 7.2 «завершення навчання процедури вбудованої діагностики».

Шоста очікувана пара подій: подія 6.1 «початок ремонту службою СТОiP» та подія 6.2 «завершення ремонту службою СТОiP».

В моделі вважаємо, що ремонт СТОiP завжди успішний ($Pr = 1$), тому після події 6.2 «завершення ремонту службою СТОiP» наступною парою очікуваних подій є сьома пара – подія 7.1 «початок навчання процедури вбудованої діагностики» та подія 7.2 «завершення навчання процедури вбудованої діагностики».

Зауваження: вважаємо наступну очікувану пару подій навчання блокуючою працездатний стан. Це означає, що хоча система вже відновлена, але навчання та зміна конфігурації потребує деякий час систему потримати в стані простою.

Сьома очікувана пара подій: подія 7.1 «початок навчання процедури вбудованої діагностики» та подія 7.2 «завершення навчання процедури вбудованої діагностики».

Варіант для очікуваної події 7.2.1: завершення навчання процедури вбудованої діагностики є успішним, тому після цього варіанту маємо покращену достовірність діагностування. Виходимо з того, що кожне навчання лише покращує систему. Далі система працює із іншими параметрами, але за тим самим алгоритмом. Для опису наступної ітерації використовується вже нова пара подій аналогічна до 1.1 та 1.2 – це восьма очікувана пара подій із подіями 8.1 «початок роботи складного обладнання після першого успішного навчання» – подія 8.2 «втрата працездатності після першого успішного навчання».

Варіант для очікуваної події 7.2.2: завершення навчання процедури вбудованої діагностики не є успішним. В такому випадку змін у показниках і конфігурації вбудованої діагностики не відбулось, хоча й система вже відновлена. Відповідно далі повертаємося на початок ітерації до першої пари подій – подія 1.1 «початок роботи складного обладнання» та подія 1.2 «втрата працездатності».

Зауваження 1: Після завершення навчання процедури вбудованої діагностики як і з варіантом 7.2.1, так і з 7.2.2, вважаємо що система переходить в працездатний стан.

Зауваження 2: Тривалість навчання системи вбудованої діагностики коротка і за час після чергового відновлення системи і одразу її навчання неактуальні такі події (такі події малоімовірні): БП1 (відмова системи), БП4 (заплановане ТО службою СТОіР).

Початок другої ітерації навчання. Нова ітерація після навчання характеризується тією ж структурою моделі з подібними подіями. Єдина відмінність полягає у зміні (покращенні) параметру достовірності вбудованої діагностичної процедури після першого успішного навчання. Тому ми пропустимо детальний опис.

Початок третьої (останньої) ітерації без навчання. Зазвичай нова ітерація після навчання характеризується тією ж структурою моделі з подібними подіями. Єдина відмінність полягає у зміні (покращенні) параметру достовірності вбудованої діагностичної процедури після першого успішного навчання, як це було описано під час другої ітерації. Але може бути ситуація коли більше навчання не можливі, або ймовірність успішності роботи процедури вбудованої діагностики = 1. В такому випадку вся структура вербальної моделі схожа до попередніх ітерацій (відповідно до описаного правила у другій ітерації). Але тут будуть відсутні кроки навчання (аналогічно до пари подій 7.1 та 7.2). Таким чином, після відновлення, яким завершуються пари базових подій 18.1 «початок запланованого ТО службою СТОіР після другого навчання» та подія 18.2 «завершення запланованого ТО службою СТОіР після другого навчання», 19.1 «початок аварійного ТО службою СТОіР після другого навчання» та подія 19.2 «завершення аварійного ТО службою СТОіР після другого навчання», 20.1 «початок ремонту службою СТОіР після другого навчання» та подія 20.2 «завершення ремонту службою СТОіР після другого навчання» система буде далі зациклюватись і повертатись до початкової пари цієї ітерації – 15.1 «початок

роботи складного обладнання після другого успішного навчання» – подія 15.2 «втрата працездатності після другого успішного навчання».

Формування базових подій. Наступний крок – це формування базових подій (БП) на основі розробленої вербальної моделі. Саме вербальна модель є джерелом правди та вербальною документацією для усіх подальших висновків, тому вона має бути завжди в актуальному стані. Серед пар подій, що були виписані у вербальній моделі, кожна друга подія пари вважається базовою. У випадку, коли часовим проміжком між двома парами подій можна знехтувати, то ми вводимо поняття зведеної базової події (ЗвБП). У результаті маємо наступні БП та ЗвБП:

1. «початок роботи складного обладнання» та «втрата працездатності» (БП1);
2. «початок процедури вбудованого контролю» та «завершення процедури вбудованого контролю» (ЗвБП2);
3. «початок процедури вбудованої діагностики» та «завершення процедури вбудованої діагностики» (БП3);
4. «початок запланованого ТО службою СТОіР» та «завершення запланованого ТО службою СТОіР» (БП4);
5. «початок аварійного ТО службою СТОіР» та «завершення аварійне ТО службою СТОіР» (БП5);
6. «початок ремонту службою СТОіР» та «завершення ремонту службою СТОіР» (БП6);
7. «початок навчання процедури вбудованої діагностики» та «завершення навчання процедури вбудованої діагностики» (БП7);
8. «початок роботи складного обладнання після першого успішного навчання» та «втрата працездатності після першого успішного навчання» (БП8);
9. «початок процедури контролю після першого успішного навчання» та «завершення процедури контролю після першого успішного навчання» (ЗвБП9);

10. «початок процедури вбудованої діагностики після першого успішного навчання» та «завершення процедури вбудованої діагностики після першого успішного навчання» (БП10);

11. «початок запланованого ТО службою СТОіР після першого успішного навчання» та «завершення запланованого ТО службою СТОіР після першого успішного навчання» (БП11);

12. «початок аварійного ТО службою СТОіР після першого успішного навчання» та «завершення аварійного ТО службою СТОіР після першого успішного навчання» (БП12);

13. «початок ремонту службою СТОіР після першого успішного навчання» та «завершення ремонту службою СТОіР після першого успішного навчання» (БП13);

14. «початок навчання процедури вбудованої діагностики після першого успішного навчання» та «завершення навчання процедури вбудованої діагностики після першого успішного навчання» (БП14);

15. «початок роботи складного обладнання після другого успішного навчання» та «втрата працездатності після другого успішного навчання» (БП15);

16. «початок процедури контролю після другого успішного навчання» та «завершення процедури контролю після другого успішного навчання» (ЗвБП16);

17. «початок процедури вбудованої діагностики після другого успішного навчання» та «завершення процедури вбудованої діагностики після другого успішного навчання» (БП17);

18. «початок запланованого ТО службою СТОіР після другого успішного навчання» та «завершення запланованого ТО службою СТОіР після другого успішного навчання» (БП18);

19. «початок аварійного ТО службою СТОіР після другого успішного навчання» та «завершення аварійного ТО службою СТОіР після другого успішного навчання» (БП19);

20. «початок ремонту службою СТОіР після другого успішного навчання» та «завершення ремонту службою СТОіР після другого успішного навчання» (БП20).

3.3.3 Розроблення ймовірнісного графу

Для розроблення ймовірнісного графу (формалізоване представлення вербальної моделі) спочатку необхідно визначити показники та параметри, які будуть представлені у дискретно-неперервній стохастичній моделі та сформувати вектор станів ймовірнісного графа експлуатаційної поведінки системи. Вектор станів – це сукупність компонент, які дають змогу представити всі можливі стани об'єкта дослідження (моделі).

Почнемо з формулювання показників та параметрів експлуатаційної поведінки системи:

1. $P_c < 1$ – ймовірність успішного завершення процедури контролю;
2. $t_c = 0$ – середній час процедури контролю вважається миттєвим;
3. $D < 1$ – (достовірність діагностування) ймовірність успішного завершення процедури вбудованої діагностики, причому цей параметр є еволюційним і покращується на кожній ітерації донавчання ($D \rightarrow D' \rightarrow D''$);
4. t_{ed} – середній час виконання вбудованої діагностики, що формує відповідну інтенсивність $\lambda_{ed} = 1/t_{ed}$;
5. $P_{sd} = 1$ – ймовірність успішного виконання процедури діагностики та ремонту під час запланованого ТО вважається безпомилковою;
6. t_{sd} – середній час запланованого ТО, інтенсивність $\mu_m = 1/t_{sd}$;
7. $P_{ad} = 1$ – ймовірність успішного виконання процедури діагностики та ремонту під час аварійного ТО;
8. t_{ad} – середній час аварійного ТО, інтенсивність $\mu_{am} = 1/t_{ad}$;
9. $P_r = 1$ – ймовірність успішного відновлення системи після успішної діагностики;
10. t_r – середній час ремонту, інтенсивність $\mu_u = 1/t_r$;

11. $P_l < 1$ – ймовірність успішного навчання засобу діагностики;
12. t_l – середній час процедури донавчання цифрового двійника, що формує інтенсивність $\lambda_l = 1/t_l$;
13. t_w – середній час безвідмовної роботи складної системи, що формує базову інтенсивність відмов $\lambda_d = 1/t_w$;
14. N – максимально можлива кількість успішних завершених навчань засобу діагностування (використовується як лічильник для коригування показників функціональності на кожній ітерації та згортання графа).

Сформуємо вектор станів ймовірнісного графа експлуатаційної поведінки системи.

1. V_1 – Стан системи ($V_1 \in 0,1$).
 - a. $V_1=0$ – початкове значення, система працездатна (відмови немає).
 - b. $V_1=1$ – система непрацездатна (є відмова).
2. V_2 – Стан вбудованої діагностики ($V_2 \in 0..1$).
 - a. $V_2=0$ – початкове значення, вбудована діагностика очікує команди від засобу контролю.
 - b. $V_2=1$ – вбудована діагностика виконується.
3. V_3 – Стан СТОіР ($V_3 \in 0..3$).
 - a. $V_3=0$ – початкове значення, СТОіР очікує.
 - b. $V_3=1$ – СТОіР задіяна до діагностики та ремонту під час запланованого ТО.
 - c. $V_3=2$ – СТОіР задіяна до діагностики та ремонту під час аварійного ТО (після неуспішного результату роботи вбудованої діагностики).
 - d. $V_3=3$ – СТОіР задіяна до ремонту.
4. V_4 – Стан навчання ($V_4 \in 0,1$).
 - a. $V_4=0$ – початкове значення, навчання процедури вбудованої діагностики не відбувається.

№	Стан, що розглядається і актуальна для цього стану БП	Вектори станів графа					№ стану графа	Перехід зі стану в стан	Формула розрахунку інтенсивності переходів
		V 1	V 2	V 3	V 4	V 5			
1	2	3	4	5	6	7	8	9	10
1	—	0	0	0	0	2	1	--	--

Продовження таблиці 3.4

1	2	3	4	5	6	7	8	9	10
2	Стан 1 БП1 втрата працездатності (ЗвБП2 завершення процедури вбудованого контролю)	1	1	0	0	2	2	1->2	$P_c \times tw$
		1	0	1	0	2	3	1->3	$(1-P_c) \times tw$
3	Стан 2 БП3 завершення процедури вбудованої діагностики	1	0	3	0	2	4	2->4	$D \times led$
		1	0	2	0	2	5	2->5	$(1-D) \times led$
4	Стан 3 БП4 запланованого (періодичного) ТО службою СТОіР	0	0	0	1	2	6	3->6	μm
5	Стан 4 БП6 завершення ремонту службою СТОіР	0	0	0	1	2	6	4->6	μu
6	Стан 5 БП5 завершення аварійного ТО службою СТОіР	0	0	0	1	2	6	5->6	μam
7	Стан 6 БП7 завершення навчання системи вбудованої діагностики	0	0	0	0	1	7	6->7	$Pl \times tl$
		0	0	0	0	2	1	6->1	$(1-Pl) \times tl$
8	Стан 7 БП1 (БП8?) втрата працездатності першого успішного навчання (ЗвБП9 завершення процедури вбудованого контролю після першого успішного навчання)	1	1	0	0	1	8	7->8	$P_c \times tw$
		1	0	1	0	1	9	7->9	$(1-P_c) \times tw$
9	Стан 8 БП10 завершення процедури вбудованої діагностики після першого успішного навчання	1	0	3	0	1	10	8->10	$D' \times led$
		1	0	2	0	1	11	8->11	$(1-D') \times led$
10	Стан 9 БП11 запланованого (періодичного) ТО службою СТОіР після першого успішного навчання	0	0	0	1	1	12	9->12	μm
11	Стан 10 БП13 завершення ремонту службою СТОіР після першого успішного навчання	0	0	0	1	1	12	10->12	μu

1	2	3	4	5	6	7	8	9	10
12	Стан 11 БП12 завершення аварійного ТО службою СТОіР після першого успішного навчання	0	0	0	1	1	12	11->12	μ_{am}
13	Стан 12 БП14 завершення навчання системи вбудованої діагностики після першого успішного навчання	0	0	0	0	0	13	12->13	$P_l \times t_l$
		0	0	0	0	1	7	12->7	$(1-P_l) \times t_l$
14	Стан 13 БП15 втрата працездатності після другого успішного навчання (ЗвБП16 завершення процедури вбудованого контролю після другого успішного навчання)	1	1	0	0	0	8	13->14	$P_c \times t_w$
		1	0	1	0	0	9	13->15	$(1-P_c) \times t_w$
15	Стан 14 БП17 завершення процедури вбудованої діагностики після другого успішного навчання	1	0	3	0	0	16	14->16	$D'' \times \lambda_{ed}$
		1	0	2	0	0	17	14->17	$(1-D'') \times \lambda_{ed}$
16	Стан 15 БП18 запланованого (періодичного) ТО службою СТОіР після другого успішного навчання	0	0	0	0	0	13	15->13	μ_m
17	Стан 16 БП20 завершення ремонту службою СТОіР після другого успішного навчання	0	0	0	0	0	13	16->13	μ_u
18	Стан 17 БП19 завершення аварійного ТО службою СТОіР після другого успішного навчання	0	0	0	0	0	13	17->13	μ_{am}

3.3.4 Розроблення структурно-автоматної моделі

Синтез структурно-автоматної моделі (САМ) та генерація опорного графа. На основі визначеного фазового простору станів та формалізованих параметрів здійснюється безпосередній синтез структурно-автоматної моделі. Функціонування системи описується через множину базових подій, для кожної з яких формулюються логічні умови виникнення, функції розподілу імовірностей переходів (ФРП) та правила модифікації компонент вектора стану (ПМКВС).

Завдяки введенню лічильника ітерацій V5, логіку однотипних базових подій на різних етапах еволюції системи було згорнуто в універсальні правила, що оптимізувало математичний опис. Узагальнену САМ для досліджуваної МФММ2 наведено у таблиці 3.5. Застосування наведених правил ПМКВС до початкового вектора Vinit ініціює автоматичну генерацію опорного марковського графа експлуатаційної поведінки, який для заданих обмежень ($N = 2$) налічує 17 унікальних станів. Слід зазначити, що V6 використовується як контейнер в ASNA для збереження змінювального значення достовірності діагностування.

Таблиця 3.5 – Структурно-автоматна модель експлуатаційної поведінки системи

Базові події	Опис ситуацій, в яких відбуваються базові події	ФРІП	ПМКВС
1	2	3	4
БП1 (БП8, БП15) втрата працездатності (ЗвБП2 (ЗвБП9, ЗвБП16) завершення процедури вбудованого контролю)	$(V1=0) \text{ AND } (V2=0) \text{ AND } (V3=0) \text{ AND } (V4=0) \text{ AND } (V5 \leq N)$	$P_c \times tw$	$V1:=1; V2:=1;$
		$(1-P_c) \times tw$	$V1:=1; V3:=1;$
БП3 (БП10, БП17) завершення процедури вбудованої діагностики	$(V1=1) \text{ AND } (V2=1) \text{ AND } (V3=0) \text{ AND } (V4=0) \text{ AND } (V5 \leq N)$	$V6 \times \lambda_{ed}$	$V2:=0; V3:=3;$
		$(1-V6) \times \lambda_{ed}$	$V2:=0; V3:=2;$
БП4 (БП11) запланованого (періодичного) ТО службою СТОіР	$(V1=1) \text{ AND } (V2=0) \text{ AND } (V3=1) \text{ AND } (V4=0) \text{ AND } (0 < V5 < (N+1))$	μ_m	$V1:=0; V3:=0; V4:=1;$
БП5 (БП12) завершення аварійного ТО СТОіР	$(V1=1) \text{ AND } (V2=0) \text{ AND } (V3=2) \text{ AND } (V4=0) \text{ AND } (0 < V5 < (N+1))$	μ_{am}	$V1:=0; V3:=0; V4:=1;$
БП6 (БП13) завершення ремонту службою СТОіР	$(V1=1) \text{ AND } (V2=0) \text{ AND } (V3=3) \text{ AND } (V4=0) \text{ AND } (0 < V5 < (N+1))$	μ_u	$V1:=0; V3:=0; V4:=1;$

1	2	3	4
БП7 (БП14) завершення навчання системи вбудованої діагностики	$(V1=0) \text{ AND } (V2=0) \text{ AND } (V3=0) \text{ AND } (V4=1) \text{ AND } (0 < V5 < (N+1))$	$P1 \times t1$	$V4:=0;$ $V5:=V5-1;$ $V6:=V6+Ls;$
		$(1-P1) \times t1$	$V4:=0;$
БП18 запланованого (періодичного) ТО службою СТОіР	$(V1=1) \text{ AND } (V2=0) \text{ AND } (V3=1) \text{ AND } (V4=0) \text{ AND } (V5=0)$	μm	$V1:=0; V3:=0;$
БП19 завершення аварійного ТО службою СТОіР	$(V1=1) \text{ AND } (V2=0) \text{ AND } (V3=2) \text{ AND } (V4=0) \text{ AND } (V5=0)$	μam	$V1:=0; V3:=0;$
БП20 завершення ремонту службою СТОіР	$(V1=1) \text{ AND } (V2=0) \text{ AND } (V3=3) \text{ AND } (V4=0) \text{ AND } (V5=0)$	μu	$V1:=0; V3:=0;$

Відповідно до методології, верифікація розробленої САМ здійснюється шляхом порівняння двох ймовірнісних графів експлуатаційної поведінки. Перший граф – це тестувальний (опорний) ймовірнісний граф, побудований на попередніх етапах аналітично (на основі базових подій та вектора станів). Другий граф генерується автоматизовано спеціалізованим програмним засобом ASNA на основі введених компонентів розробленої САМ. Головною метою такого порівняння є виявлення будь-яких невідповідностей між графами: наявності зайвих або пропущених станів, надмірності чи відсутності переходів, а також перевірка коректності значень інтенсивностей цих переходів.

У межах даного дисертаційного дослідження верифікацію розробленої САМ було успішно виконано саме за цим принципом. Шляхом перехресного порівняння власноруч побудованого тестувального ймовірнісного графа із графом, автоматично згенерованим програмною моделлю

3.3.5 Дослідження

Індикатор готовності для даної моделі розраховується як сума ймовірностей перебування в кожному з робочих станів. У цьому випадку, маючи приклад графіка з 2 можливими успішними ітераціями перенавчання, формула буде аналогічною до МФММ1, а саме 3.6.

На основі САМ аналогічно до формул 3.7-3.9 з дослідження МФММ1 формується система диференціальних рівнянь Колмогорова-Чепмена. Проте вона повністю формується і вирішується за допомогою метода у спеціалізованому програмному забезпеченні ASNA. Для розв'язання системи диференціальних рівнянь у ASNA використовувався метод чисельного інтегрування Рунге-Кутти-Мерсона. Для забезпечення необхідної точності розрахунків було обрано кінцеву точку інтервалу моделювання $T_k = 3000$ годин та крок інтегрування $Re = 10^{-18}$. Для проведення подальшого порівняльного аналізу було сформовано базовий (еталонний) сценарій експлуатації, який забезпечує стаціонарний коефіцієнт готовності $Ab = 0,9215$ при наступних значеннях вхідних параметрів та індикаторів, що за своєю логікою діапазонів співвідносяться з обґрунтуванням у підрозділі 3.2 у таблиці 3.2.

1. Параметри часу: $\lambda_d = 1/200$; $\lambda_{ed} = 120$; $\mu_{am} = 1/20$; $\mu_m = 1/188$; $\mu_u = 1/8$; $\lambda_l = 2$; лічильник ітерацій $N = 2$.

2. Ймовірнісні індикатори: $P_c = 0.95$; $P_l = 0.7$; еволюція достовірності діагностування: $P_{ed} = 0.7$ (базова), $P_{ed}' = 0.85$ (1-ша ітерація), $P_{ed}'' = 1.0$ (2-га ітерація).

Далі, аналогічно до дослідження МФММ1, виконується ряд симуляційних сценаріїв для валідації створеної моделі (sensitivity analysis). Результати експериментів з описом представлені у таблиці 3.6.

Таблиця 3.6 – Валідаційні експерименти для МФММ2 за методологією CAM

№	Параметр чи індикатор	Базове значення	Ab	Змінені значення	A	Коментар
1	2	3	4	5	6	7
1	λd	1/200	0,9215	1/50	0,7461	Чим вище середнє значення часу безвідмовної роботи обладнання, тим вище коефіцієнт готовності. Як і очіувалося, зміни цього параметра мають сильний вплив.
				1/100	0,8546	
				1/1000	0,982	
2	λed	120	0,9215	1200	0,9215	Зниження інтенсивності тривалості процедури вбудованої діагностики знижує готовність. При збільшенні існує певна точка насичення (діапазон 120-1,2), при якій збільшення інтенсивності не має сенсу.
				1.2	0,9182	
				0.12	0,8891	
3	μam	1/20	0,9215	1	0,9215	Зниження інтенсивності діагностичної та ремонтної процедури під час аварійного обслуговування практично не впливає на готовність, що є несподіваним. Тільки при дуже низьких значеннях. Однак у процесі було виявлено, що протягом тривалого періоду спостереження система все менше і менше залежить від значень, які впливають тільки на перехідні стани. З огляду на те, що при $D'' = 1$ значення μam не має значення для останньої ітерації. Тому, повторивши цей експеримент з $D'' = 0,95$, маємо досить значний ефект від змін μam .
				1/200	0,9211	
				1/2000	0,7641	
	$\mu am (D'')$	1/20 (0.95)	0,9192	1	0,9229	
				1/200	0,8841	
4	μm	1/188	0,9215	1	0,9630	Очікується, що інтенсивність процедури діагностики та ремонту під час планового технічного обслуговування матиме значний вплив на готовність. Збільшення цього значення допомагає у випадках прихованої несправності, коли контроль не спрацював.
				1/50	0,9517	
				1/500	0,8597	
				1/2000	0,6842	
5	μu	1/8	0,9215	1.25	0,9516	Збільшення інтенсивності відновлення значно підвищує готовність. І відповідно навпаки. Це один з найбільш впливових параметрів в моделі.
				1/25	0,8577	
				1/80	0,7006	

1	2	3	4	5	6	7
6	λ	2	0,9215	20	0,9215	Зміна інтенсивності вбудованої діагностичної процедури навчання за інформацією від СТОіР відповідно впливає на готовність. Коли інтенсивність збільшується, готовність зростає, і навпаки. Слід зазначити, що дуже малі значення не є актуальними в реальній системі.
				1/200	0,9109	
				1/500	0,7847	
7	Pc	0.95	0,9215	0.99	0,9531	Ймовірність успішного завершення процедури контролю Pc також є одним із впливових показників. Її зниження призводить до частіших прихованих відмов і зниження експлуатаційної готовності. Водночас це частково компенсується збільшенням параметра μm.
				0.85	0,8508	
				0.7	0,7632	
8	Pl	0.7	0,9215	0.05	0,9100	Ймовірність успішного навчання вбудованої діагностичної процедури Pl має незначний вплив для останньої ітерації і впливає переважно на перехідні стани. Таким чином, чим довше ми спостерігаємо, тим менше Pl впливає на готовність. З заданими параметрами Tk і Re маємо невеликий вплив. Але коли зменшуємо Tk до 500 і знижуємо інтенсивність аварійного відновлення μam до 1/100 (що є актуальним для менш навчених діагностичних процедур), бачимо, що вплив збільшився. Тому в довгостроковій перспективі, коли швидко донавчання досягає ліміту, початковий час навчання не є критичним, але при плануванні готовності на початку, і особливо коли час аварійного відновлення є значним, його також слід враховувати.
				0.5	0,9214	
				0.95	0,9216	
	Pl (Tk) (μam)	0.7 (500) (1/100)	0.8728	0.05	0.8264	
				0.5	0.8601	
				0.95	0.8862	

1	2	3	4	5	6	7
9	D	0.7	0,9215	0.05	0,8821	Ймовірність успішного завершення вбудованої діагностичної процедури не сильно змінюється, коли вартість діагностичної помилки відносно невеликою ($\mu_{am} = 1/20$ проти $\mu_i = 1/8$).
	D'	0.85		0.1		
	D''	1		0.15		
10	D	0.7	0.9214	0.05	0.6865	З іншого боку, як тільки вартість вбудованої діагностичної помилки збільшується, зміна ймовірності успішного завершення вбудованої діагностичної процедури починає мати сильніший вплив. Особливо при низьких значеннях. Слід також зазначити, що при тривалому спостереженні перехідні стани практично перестають впливати на кінцеву доступність. Тому при поточних значеннях $T_k=3000$, D'' має більший вплив. Це можна побачити в експерименті 11, де середнє значення D, D' і D'' є більшим за базові значення, але D'' є меншим за базове D'', і готовність стає меншою, ніж готовність з базовими параметрами $A < A_b$. Таким чином, залежно від часу спостереження, значення Pl і параметра μ_{am} , D (включаючи його зміни з часом) може зазнавати різних впливів.
	D'	0.85		0.1		
	D''	1		0.15		
	μ_{am}	1/100		1/100		
11	D	0.7	0.9214	0.90	0.9141	
	D'	0.85		0.95		
	D''	1		0.98		
	μ_{am}	1/100		1/100		

3.3.6 Попередні результати

Проведені експерименти з МФММ2 довели адекватність розробленої моделі: загальна динаміка коефіцієнта готовності підпорядковується очікуваному тренду, підтверджуючи базову гіпотезу, висунуту на етапі дослідження МФММ1.

Зокрема, доведено, що донавчання інтелектуальної системи діагностування демонструє найвищу ефективність у випадках, коли «вартість» помилки діагностування є високою (наприклад, коли час аварійного ручного пошуку дефекту t_{ad} значно перевищує час ремонту t_r). Водночас, в умовах довгострокової експлуатації (при великих значеннях T_k) вплив перехідних станів (таких як сам процес навчання) поступово нівелюється, і вирішальне значення має остаточне (стаціонарне) значення достовірності D .

Загалом можна встановити наступну логіку донавчання у поєднанні із СТОіР. У разі виявлення СТОіР невідповідності між фактичним технічним станом обладнання та результатами діагностування, до ЦД вносяться відповідні коригувальні відомості. Це ініціює процедуру донавчання, яка адаптує моделі до нових умов експлуатації та підвищує загальну достовірність діагностування (D). Завдяки цьому, при виникненні аналогічних відмов у майбутньому, забезпечується точна локалізація дефекту вбудованими засобами, що дозволяє персоналу СТОіР прибувати на об'єкт із заздалегідь підготовленими інструментами, комплектуючими та алгоритмами дій. У кінцевому підсумку це призводить до суттєвого зменшення середньої тривалості процедури відновлення (t_r) та підвищення стаціонарної готовності системи.

Окремим результатом є апробація самої методології структурно-автоматного моделювання. До її переваг належить здатність вирішувати проблему моделювання графі із великою кількістю, в тому числі схожих, що є релевантним для МФММ зі змінними параметрами. Вербалізація умов та компактний табличний опис базових подій дозволяє генерувати графи з сотнями станів, уникаючи помилок ручного проєктування та фокусуючись на логіці поведінки системи.

Разом з тим, застосування даної методології виявило низку обмежень. По-перше, наявність розриву в 5–6 порядків між інтенсивностями відмов та швидкостями відновлення/навчання робить матрицю переходів математично «жорсткою» (stiff system). По-друге, оскільки інструментарій (зокрема ASNA) базується на чисельних методах розв'язання (Рунге-Кутти-Мерсон), він дає

залежне від часу рішення, яке лише апроксимує стаціонарний розподіл. Це призводить до обмеженої точності обчислень (похибка може виникати після 4-го знаку після коми порівняно з аналітичним розв'язанням лінійних рівнянь). Ці обмеження вимагають ретельного підбору кроку інтегрування та застосування додаткових методів верифікації результатів, що буде детально розглянуто у наступному підрозділі. І останнім недоліком слід зазначити обмежені програмні можливості щодо експорту та візуалізації результатів.

3.4 Метод валідації та верифікації МФММ ІСД

Постановка проблеми достовірності стохастичного моделювання Проектування та оцінювання готовності критичного промислового обладнання (зокрема, систем управління малих модульних реакторів SMR) [3] вимагає гарантування надвисоких показників надійності, де вимоги до коефіцієнта готовності регламентуються жорсткими стандартами безпеки і можуть становити 0,99999 і вище. Забезпечення такого рівня довіри до результатів вимагає високої точності на етапі стохастичного моделювання, що супроводжується двома фундаментальними проблемами: структурною та обчислювальною.

Перша проблема – структурна складність та верифікація синтезу. Використання марковських моделей для ІСД із донавчанням призводить до зростання кількості станів через наявність змінних параметрів (еволюція достовірності діагностування D на кожній ітерації навчання). При розширенні базової моделі кількість станів та переходів у графі лінійно зростає пропорційно кількості фрагментів. Ручний структурно-параметричний синтез таких графів є вразливим до логічних помилок дослідника (наприклад, пропущені стани, некоректні інтенсивності переходів або порушення марковської властивості). Відсутність формалізованих процедур верифікації самої структури графа на етапі, що передує математичним розрахункам, може призводити до отримання хибних результатів моделювання.

Друга проблема – обмеження інструментарію та математична жорсткість. Навіть за умови коректної структури графа, розрахунок ймовірностей стикається з явищем математичної «жорсткості» систем диференціальних рівнянь Колмогорова-Чепмена [6]. Як зазначається у фахових дослідженнях марковського моделювання систем критичної інфраструктури [6], ця жорсткість виникає через значний розрив у порядках величин: інтенсивності виникнення відмов вимірюються роками ($\lambda \approx 10^{-5}$ год⁻¹), тоді як процедури відновлення, контролю або донавчання – годинами ($\mu \approx 10^0$ год⁻¹). Формально марковська модель класифікується як система з високою жорсткістю, якщо її індекс жорсткості $s(x)$ (відношення максимального та мінімального модулів дійсних частин власних значень матриці Якобі) задовольняє умову $s(x) \geq 10^3$ [6]. Оскільки матрицею Якобі для таких систем фактично виступає матриця інтенсивностей переходів Q , на практиці цей індекс відображає співвідношення між найбільшим та найменшим значеннями інтенсивностей у системі.

Використання класичних чисельних методів (наприклад, явних методів Рунге-Кутти), які лежать в основі багатьох загальних інструментів марковського моделювання, для жорстких систем вимагає малого кроку інтегрування. На тривалих інтервалах моделювання експлуатації це призводить до накопичення локальних похибок машинного округлення [6, 7]. У результаті розрахункові значення коефіцієнта готовності можуть втрачати точність вже на 3 та 4 знаку після коми. Подолання проблеми жорсткості спирається на дві обчислювальні стратегії, кожна з яких має суттєві обмеження для моделювання ІСД [6].

1. Стратегія уникнення жорсткості (Stiffness-Avoidance Approach, SAA) передбачає математичну агрегацію станів зі швидкими інтенсивностями переходів [8]. В такому випадку система перестає бути жорсткою. Однак процедура дезагрегації базується на припущенні про миттєве досягнення стаціонарного значення в середині агрегованих станів, що вносить похибку апроксимації у перехідні процеси [8]. Проте SAA більше адаптивна до великих моделей розміром у десятки чи сотні тисяч станів [6].

2. Стратегія толерантності до жорсткості (Stiffness-Tolerance Approach, STA) зберігає оригінальну матрицю без агрегації, гарантуючи високу точність завдяки неявним L-стабільним методам розв'язання [9]. Точність до 8 знаків після коми із методом TR-BDF2, та неявні методи Рунге-Кутти для ще більшої точності. Проте застосування STA обмежується проблемою розмірності: Алгоритмічна складність такого метода складає $O(N^3)$ [9].

Метою підрозділу є розроблення комплексного методу багатоступеневої перехресної верифікації та валідації (V&V) марковських моделей промислового обладнання з використанням ПоТ та донавчання, інтегрованого з алгоритмом обґрунтування обчислювальної стратегії та програмного інструментарію (Tool Justification). Пропонований метод спрямований на вирішення структурних та обчислювальних проблем, описаних вище.

Внутрішня верифікація в межах методологій. Почнемо з визначень. Верифікація – це використання емпіричних даних, спостережень, випробувань або експериментів для підтвердження істинності або раціонального обґрунтування гіпотези. Або підтвердження шляхом перевірки та надання об'єктивних доказів того, що визначені вимоги виконано. У свою чергу, валідація – це встановлення на основі об'єктивних доказів того, що технічні характеристики відповідають призначенню. Тобто верифікація відповідає за відповідність вимогам, а валідація – за відповідність вимог щодо передбачуваного використання. Процес підтвердження адекватності марковських моделей ІСД розпочинається з процедур внутрішньої верифікації, які притаманні кожній з обраних методологій. По-перше, здійснюється перевірка марковської властивості моделі: підтверджується відсутність післядії та відповідність часу перебування у станах експоненційному закону розподілу. По-друге, виконується обов'язкова верифікація згенерованого інструментарієм графа шляхом його структурного та логічного порівняння з початковим опорним математичним графом. Це дозволяє виключити помилки програмної імплементації логічних правил (зокрема, ПМКВС у структурно-автоматній моделі).

Перехресна верифікація (Cross-Verification) та її переваги. Внутрішньої верифікації може бути недостатньо, тому пропонується використання ідеї перехресної верифікації, яка полягає у паралельному застосуванні двох незалежних парадигм моделювання до одного об'єкта. По-перше, це перевірка (тестування) альтернативної методології з опорою на результати перевіреної еталонної. По-друге, паралельна побудова моделей за двома незалежними парадигмами концептуально реалізує принцип N-версійного проектування, що мінімізує ймовірність структурно-логічних помилок дослідника (людського фактора). По-третє, це взаємне порівняння та контроль коректності різних програмних інструментів. По-четверте, це самоперевірка, навіть якщо одна людина моделює ту саму систему двома різними шляхами. Перехресна верифікація виконується поетапно наступним чином.

1. Крос-верифікація вихідних математичних графів. На цьому етапі структурно порівнюються математичні графи (наявність станів, переходів та їх інтенсивностей), побудовані за двома різними методологіями на основі єдиного вербального опису. У разі виявлення розбіжностей проводиться аналіз причин та вносяться відповідні коригування доки не буде досягнуто повної структурної ідентичності математичних графів.

2. Крос-верифікація інструментальних графів. Виконується порівняння графів, згенерованих безпосередньо у різних програмних засобах (наприклад, пакет `markovchain` у середовищі R/Python та ASNA), з уже верифікованим еталонним математичним графом із попереднього кроку. Це гарантує коректність програмної імплементації логіки моделі в конкретному інструменті.

3. Крос-верифікація результатів симуляції. На фінальному етапі проводиться звірка результатів симуляції, отриманих із різних програмних засобів, які використовують верифіковані графи та абсолютно ідентичні вхідні параметри експерименту. За умови отримання повного збігу результатів констатується успішна та повна перехресна верифікація. У разі виникнення розбіжностей здійснюється глибокий аналіз їхньої природи. Проблема може полягати або у конфігурації симуляції (наприклад, задані різні інтервали чи кроки

моделювання), або у самому інструменті (його обмеженій точності чи специфіці внутрішніх математичних методів розв'язання диференціальних рівнянь). Розуміння цих обмежень є критично важливим при моделюванні критичних систем із високими вимогами до точності функції готовності.

Валідація марковських моделей ІСД та причини неочікуваних результатів. Після успішної верифікації структури проводиться етап валідації, мета якого – підтвердити, що поведінка моделі відповідає реальному об'єкту. Для цього проводиться серія валідаційних імітаційних експериментів (із варіюванням параметрів моделі) з подальшим аналізом функції готовності $A(t)$. На цьому етапі дослідник може зіткнутися з неочікуваними (недостовірними) результатами, які суперечать фізичному змісту роботи ІСД. Основними причинами неочікуваних результатів є такі.

1. Помилки параметризації (використання некоректних вхідних значень інтенсивностей переходів). Можливе рішення: ретроспективний перегляд вихідних експлуатаційних даних, уточнення статистичних вибірок та повторна експертна оцінка параметрів надійності.

2. Порушення марковської властивості (реальний час виконання процедур діагностування, відновлення чи донавчання суттєво відрізняється від показникового/експоненційного закону розподілу). Можливе рішення: застосування методу фаз Ерланга [10, 11]. Цей підхід дозволяє формалізовано звести довільні (наприклад, детерміновані або нормально розподілені) часові затримки до марковського процесу. Це досягається шляхом розщеплення одного неекспоненційного переходу на послідовність k фіктивних станів (фаз), час перебування в кожному з яких підпорядковується експоненційному розподілу.

3. Невідповідність вербальної моделі реальному процесу (наявність неврахованих станів, хибних або пропущених переходів між ними). Можливе рішення: перегляд вербальної моделі, структурне коригування опорного графа та повторна верифікація.

4. Обмежена точність обчислювальних інструментів. Можливе рішення: застосування спеціалізованого алгоритму обґрунтування вибору інструментарію (Tool Justification), що буде розглянуто далі.

Алгоритм вибору інструментарію (Tool Justification). Деталізуючи четверту причину, слід зазначити, що вона є особливо важливою для критичних систем з високими вимогами до точності оцінки готовності (5 і вище знаків після коми). Як було зазначено на початку підрозділу, жорсткі системи впливають на точність рішення, що особливо помітно при використанні класичних методів розв'язання диференціальних рівнянь (наприклад, явних методів Рунге-Кутти), які призводять до накопичення локальних похибок машинного округлення. Для розв'язання проблеми пропонується алгоритм вибору інструментарію (рис. 3.6).

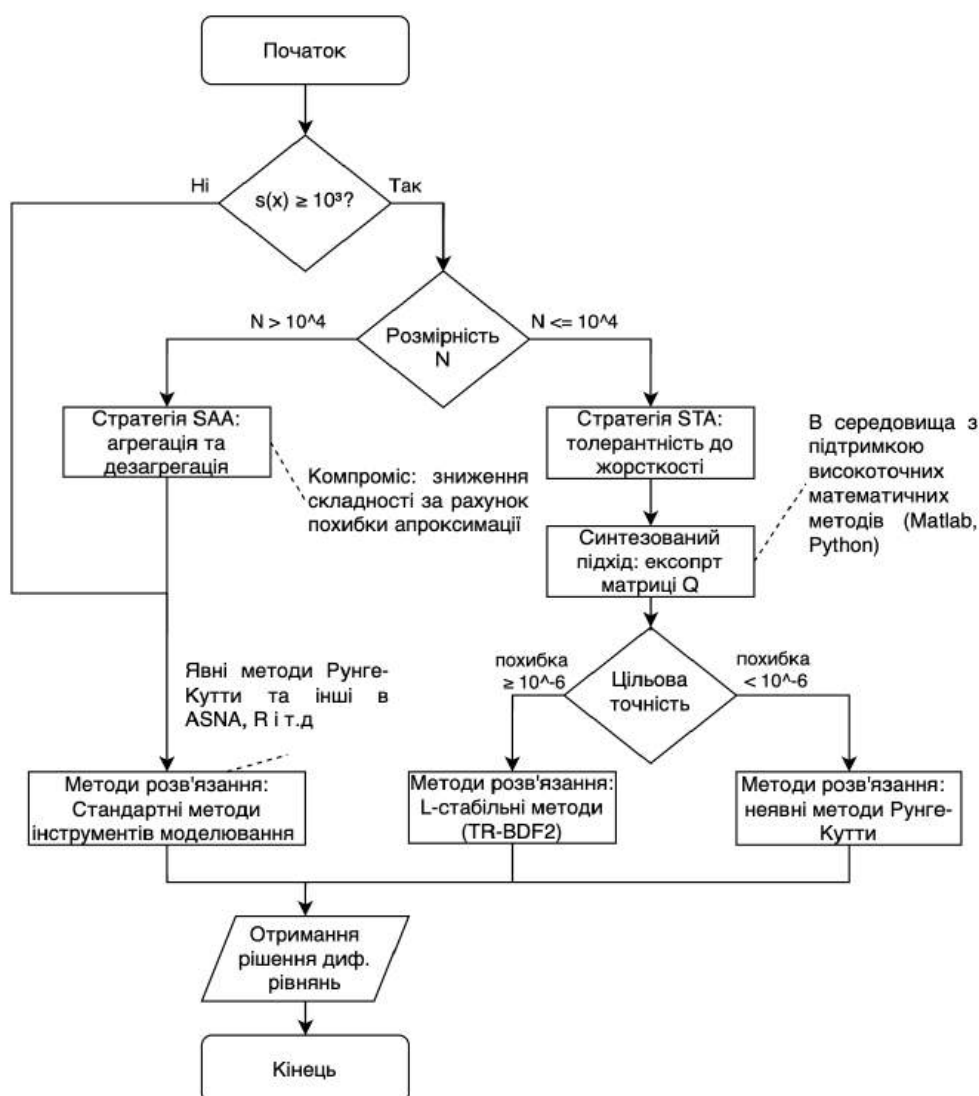


Рисунок 3.6 – Алгоритм вибору інструментарію

Алгоритм передбачає послідовну багатокритеріальну оцінку: якщо система класифікується як жорстка ($s(x) \geq 10^3$), маршрутизація обчислень залежить від розмірності простору станів. Для надвеликих графів ($N > 10^4$) застосовується стратегія SAA (агрегація) [6, 8], що знижує обчислювальну складність за рахунок внесення похибки апроксимації. Однак для компактних систем ($N \leq 10^4$) алгоритм регламентує перехід до стратегії STA [6, 9]. Вона реалізується через синтезований підхід: вихідна матриця Q експортується до фундаментальних математичних середовищ (MATLAB, Python), де залежно від цільових вимог розв'язується або базовими L-стабільними методами (TR-BDF2 для допустимої похибки $\geq 10^{-6}$), або неявними методами Рунге-Кутти для гарантування надвисокої точності (похибка $< 10^{-6}$).

Комплексний метод валідації та верифікації (V&V). З метою об'єднання описаних етапів у єдиний технологічний процес, розроблено комплексний метод V&V марковських моделей ІСД. Метод формалізується множиною базових елементів, наведених у таблиці 3.7.

Таблиця 3.7 – Визначення елементів запропонованого методу V&V

№	Елемент діаграми	Опис
1	O	Об'єкт дослідження (ІСД)
2	M	Методологія. $M = \{ M_s, M_e \}$
3	G	Граф. $G = \{ G_s, G_e \}$
4	T	Інструмент (Tool). $T = \{ T_s, T_e \}$
5	Gt	Граф, побудований інструментом. $Gt = \{ Gts, Gte \}$
6	R	Результати модельних експериментів. $R = \{ Rs, Re \}$

Примітка: індекс s біля елементів позначає методологію на основі графів станів (state-based), а індекс e – методологію на основі структурно-автоматних моделей, орієнтовану на події (event-based). Загальну структурну схему

розробленого методу наведено на рисунку 3.7. Схема розділена на два сценарії застосування залежно від рівня критичності об'єкта та доступних ресурсів.

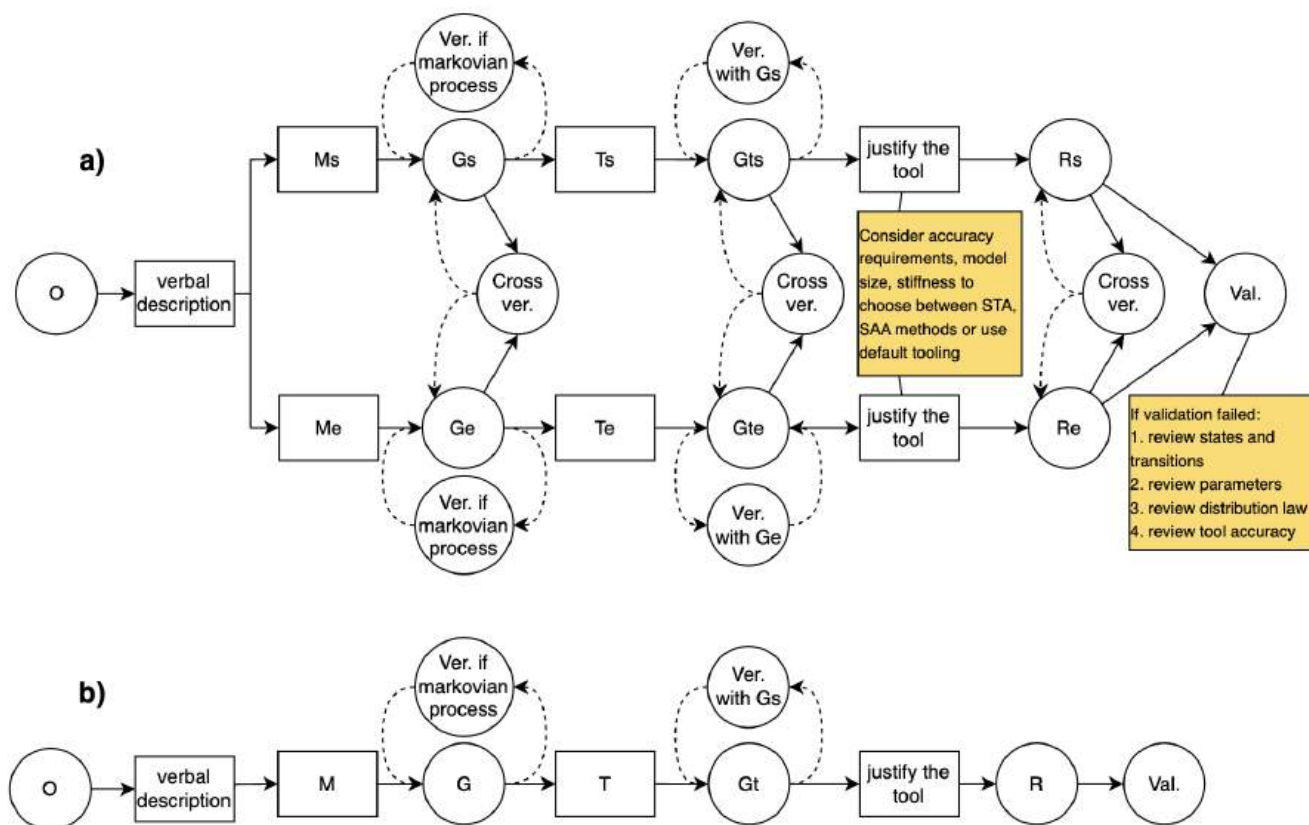


Рисунок 3.7 – Структурна схема методу V&V марковських моделей ІСД: а) повний цикл перехресної верифікації та валідації; б) спрощений варіант для однієї методології

На рисунку 3.7, а проілюстровано повний цикл перехресної верифікації та валідації (Cross V&V). У цьому сценарії дві гілки моделювання (Ms та Me) постійно контролюють одна одну на кожному етапі: від порівняння вихідних математичних графів (Gs і Ge) до зв'язки фінальних результатів симуляції (Rs і Re), відповідно до сформульованих трьох етапів крос-верифікації. Центральним елементом схеми є блок обґрунтування інструменту (Justify the tool), який ініціює описаний раніше алгоритм для забезпечення математичної точності розрахунків.

На рисунку 3.7, б наведено спрощений варіант методу. Він застосовується у випадках, коли використання другої методології є недоцільним або неможливим

(наприклад, при обмеженнях ресурсів розробки або для некритичних систем ПоТ нижнього рівня). У цьому випадку цикл перехресної верифікації виключається, однак зберігаються вимоги до внутрішньої верифікації єдиної обраної методології, обов'язкове виконання алгоритму вибору інструментарію та етап валідації результатів.

Таким чином, у даному підрозділі отримано третій науковий результат (НР3), який полягає у розробленні комплексного методу багатоступеневої валідації та верифікації марковських моделей інтелектуальних систем діагностування. Детальне формулювання наукової новизни та переваг отриманого результату наведено у загальних висновках до розділу. Алгоритмічні основи запропонованого методу перехресної верифікації, а також його місце в загальній інформаційній технології автоматизованого оцінювання готовності ІСД були висвітлені автором у науковій праці [4].

3.5 Висновки до третього розділу

У розділі виконано комплексне розроблення, симуляційне моделювання та верифікацію математичних моделей експлуатаційної поведінки інтелектуальних систем діагностування (ІСД) із донавчанням, що дало змогу сформулювати такі висновки.

1. Сформовано другу (математичну) складову наукового результату 2 – розроблено алгоритм структурно-математичного синтезу розрахункових МФММ, – та **отримано в повному обсязі другий науковий результат (НР2):** удосконалено базові та мультифрагментні моделі готовності відновлювального промислового обладнання систем індустріального інтернету речей, які, на відміну від існуючих, утворюють множину, сформовану на основі розробленого онтологічного класифікатора, та враховують зміну достовірності діагностування, зумовлену донавчанням цифрового двійника після прояву різних типів відмов, що дозволяє обирати структуру моделі залежно від особливостей об'єкта і процесів діагностування та підвищити достовірність оцінювання готовності системи.

Матеріали досліджень, що формують цей науковий результат пройшли міжнародну апробацію та висвітлені у наукових працях [1-4].

2. Розроблено та досліджено МФММ1 та МФММ2. Емпірично підтверджено базову гіпотезу дослідження: впровадження процедур донавчання ІСД призводить до тимчасового зниження функції готовності на початкових етапах експлуатації (формування «навчального боргу» на протязі T_{loss}), однак забезпечує виграв у стаціонарному коефіцієнті готовності (A_{steady}) у довгостроковій перспективі. В результаті аналізу чутливості виявлено низку закономірностей: донавчання демонструє найвищу ефективність для систем, де час аварійного відновлення після прихованої відмови значно перевищує час штатного ремонту, а підвищення початкової достовірності (D_{start}) пропорційно зменшує глибину початкових втрат готовності. Водночас визначено умови недоцільності впровадження донавчання: якщо алгоритми мають низький потенціал (асимптотичний максимум D_{max} є малим або близьким до D_{start}), то невеликий приріст достовірності діагностування не перекриває збитків від простоїв обладнання, тому від оперативного донавчання доцільно відмовитися. Виявлено структурні недоліки спрощеного лінійного моделювання (МФММ1), яке не здатне явно відобразити багаторівневі процеси технічного обслуговування (СТОіР) та відокремити вбудований контроль від діагностування. З огляду на це, розроблена матрична модель МФММ2 є більш адекватною для відображення реальних експлуатаційних процесів.

3. Апробовано методологію структурно-автоматного моделювання (САМ). Доведено, що використання орієнтованої на події парадигми (event-based) із застосуванням правил модифікації компонент вектора стану дозволяє ефективно вирішити проблему моделюванням великих моделей та автоматизувати генерацію простору станів системи з урахуванням еволюції її параметрів.

4. Виявлено та формалізовано проблему обчислювальної складності (жорсткості). Встановлено, що розрив у 5–6 порядків між інтенсивностями виникнення відмов обладнання (λ_d) та швидкостями процедур відновлення/навчання (μ , λ_l) робить матриці переходів критичних систем

математично «жорсткими» ($s(x) \geq 10^3$). Це обмежує точність класичних методів чисельного інтегрування (наприклад, явних методів Рунге-Кутти) через накопичення похибок округлення.

5. У розділі отримано третій науковий результат (НРЗ): удосконалено метод верифікації та валідації мультифрагментних марковських моделей інтелектуальних систем діагностування промислового обладнання з використанням індустриального інтернету речей, який відрізняється використанням багатоступеневої перехресної перевірки результатів за допомогою двох незалежних методологій (на основі графів станів та структурно-автоматних моделей), а також врахуванням математичної «жорсткості» матриці інтенсивностей для вибору чисельного методу розв'язання, що дозволяє уникнути обчислювальних похибок та підвищити достовірність оцінювання готовності. Основні теоретичні та практичні положення третього наукового результату опубліковано у фахових наукових виданнях [3, 4] та матеріалах міжнародної конференції [2].

Література до третього розділу

1. Shcheglov V., Kharchenko V., Borges J., Morozova O. Availability Markov Models of Complex Equipment with Relearning Diagnostic Means after Detecting Latent Failures. *Dependable Systems, Services and Technologies (DESSERT'2023)* : Proc. 13th IEEE Int. Conf., Athens, Greece, 2023. P. 1–7. DOI: 10.1109/DESSERT61349.2023.10416539.
2. Shcheglov V., Kharchenko V., Volochiy B., Morozova O., Zaitseva E. Availability Assessing Intelligent Diagnostic System with Relearning after Complex Equipment Failures: Structural-Automata Model-based Methodology of Markov's Analysis. *Dependable Systems, Services and Technologies (DESSERT'2024)* : Proc. 14th IEEE Int. Conf., Athens, Greece, 2024. P. 1–8. DOI: 10.1109/DESSERT65323.2024.11122169.

3. Kharchenko V., Shcheglov V., Ivasiuk O., Morozova O. Digital Twin-Based Lifecycle Methodology for Ensuring Safety of NPP/SMR I&C Systems. *Technologies*. 2026. Vol. 14, no. 1, article no. 46. P. 1–33. DOI: 10.3390/technologies14010046.
4. Щеглов В. Р. Інформаційна технологія оцінювання готовності інтелектуальних систем діагностування з донавчанням на основі цифрових двійників. *Measuring and computing devices in technological processes*. 2026. № 1. С. 87–96. DOI: 10.31891/2219-9365-2026-85-11.
5. Volochiy B., Mandziy B., Ozirkovsky L. Extending the features of software for reliability analysis of fault-tolerant systems. *Computational Problems of Electrical Engineering*. 2012. Vol. 2, no. 2. P. 113–121.
6. Kharchenko V., Odarushchenko O., Butenko V., Popov P. Selecting mathematical software for dependability assessment of computer systems described by stiff Markov chains. *CEUR Workshop Proceedings*. 2013. Vol. 1000. P. 146–162. URL: https://www.researchgate.net/publication/289567556_Selecting_mathematical_software_for_dependability_assessment_of_computer_systems_described_by_stiff_Markov_chains (дата звернення: 02.02.2026).
7. Hairer E., Wanner G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Vol. 14. Springer Science & Business Media, 1996. URL: <https://link.springer.com/book/10.1007/978-3-642-05221-7> chains (дата звернення: 03.03.2026).
8. Bobbio A., Trivedi K. S. An aggregation technique for the transient analysis of stiff Markov chains. *IEEE Transactions on Computers*. 1986. Vol. C-35, no. 9. P. 803–814. URL: <https://ieeexplore.ieee.org/document/1676840> (дата звернення: 03.03.2026).
9. Malhotra M., Muppala J. K., Trivedi K. S. Stiffness-tolerant methods for transient analysis of stiff Markov chains. *Microelectronics Reliability*. 1994. Vol. 34, no. 11. P. 1825–1841. URL: <https://www.sciencedirect.com/science/article/abs/pii/0026271494901376> (дата звернення: 03.03.2026).

10. Volochiy S., Fedasyuk D., Chohey R. Formalized development of the state transition graphs using the Erlang phase method. *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications* : Proc. 9th IEEE Int. Conf., Bucharest, Romania, Sept. 21–23, 2017. P. 1098–1101. 10.1109/IDAACS.2017.8095255.
11. Strielkina A., Volochiy S., Kharchenko V. Discrete-Continuous Stochastic Model of Insulin Pump Functioning for Health IoT System Using Erlang Phase Method. *ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer* : Proc. 15th Int. Conf., Volume II: Workshops, Kherson, Ukraine, Jun. 12–15, 2019. P. 793–808. URL: http://ceur-ws.org/Vol-2393/paper_402.pdf (дата звернення: 03.03.2026).

РОЗДІЛ 4 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ТА ПРОГРАМНИЙ ЗАСІБ ДЛЯ СИСТЕМ ДІАГНОСТУВАННЯ ПРОМИСЛОВОГО ОБЛАДНАННЯ З ВИКОРИСТАННЯМ ІНДУСТРІАЛЬНОГО ІНТЕРНЕТУ РЕЧЕЙ ІЗ ДОНАВЧАННЯМ НА ОСНОВІ ЦИФРОВИХ ДВІЙНИКІВ. ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ

4.1 Послідовність використання та особливості практичної реалізації інформаційної технології і розроблених моделей

На підставі розроблених у другому і третьому розділах моделей та методів сформулюємо загальну послідовність використання розроблених засобів для проектування, оцінювання та забезпечення готовності промислового обладнання в середовищі індустріального інтернету речей за допомогою інтелектуальних систем діагностування із цифровими двійниками. Зазначена послідовність представляє собою єдиний технологічний процес і складається з наступних кроків.

1. Аналіз об'єкта та формування вимог. Визначення вимог до показників готовності промислового обладнання, аналіз його критичності, особливостей побудови засобів базового контролю та можливостей інтеграції IoT-сенсорів для збору телеметрії (спираючись на результати аналізу з підрозділу 1.1).

2. Формування архітектури ІСД. Застосування розробленої теоретико-множинної моделі та дворівневої (Edge-Cloud) архітектури інтелектуальної системи діагностування (підрозділ 2.1). На цьому етапі визначається розподіл функцій: які алгоритми локального донавчання реалізуються безпосередньо на контролерах граничного рівня (Edge), а які дані акумулюються для федеративного навчання цифрового двійника на сервері (Cloud).

3. Вибір базової моделі (БМ). На основі аналізу специфіки об'єкта та обраної архітектури ІСД застосовується розроблений класифікатор моделей (підрозділ 2.2). Залежно від типу фрагментності, наявності засобів технічного обслуговування (СТОiP) та інших ознак, з комплексу базових моделей (БМ1–

БМ5) обирається найбільш релевантна за допомогою багатокритеріального ранжування і інструменті «ПОЛПЕДР-Альтернатива» [13].

4. Синтез та розрахунок марковської моделі. Обрана базова модель розгортається у мультифрагментну марковську модель (МФММ), що дозволяє врахувати динаміку зміни достовірності діагностування (D) внаслідок еволюції цифрового двійника (підрозділи 3.1–3.3). Здійснюється параметризація моделі (визначення інтенсивностей λ , μ) та формування системи диференціальних рівнянь Колмогорова-Чепмена.

5. Верифікація обчислень (V&V) та обґрунтування інструментарію. Для гарантування математичної точності результатів (особливо для критичних систем типу SMR) застосовується розроблений алгоритм обґрунтування вибору інструментарію (Tool Justification) та метод перехресної верифікації (підрозділ 3.4). Це дозволяє гарантувати необхідну точність обчислення під час моделювання.

6. Аналіз результатів та прийняття рішень. На основі розрахованої функції готовності $A(t)$ аналізуються ключові метрики: стаціонарна готовність (A_{steady}), мінімальне значення готовності (A_{min}) та час програшу у готовності відносно моделі без донавчання (T_{loss}). За результатами порівняння динамічної та статичної моделей формується висновок щодо економічної і технічної доцільності впровадження конкретної стратегії донавчання цифрового двійника.

Особливості практичної реалізації інформаційної технології. Оскільки практичне виконання наведеної послідовності вимагає роботи зі складними кіберфізичними архітектурами, опрацювання великих просторів станів та розв'язання жорстких систем диференціальних рівнянь, ручне застосування запропонованих методів є неефективним. Для автоматизації наведеної послідовності та впровадження теоретичних розробок, пропонується комплекс засобів, який концептуально поділяється на такі дві складові.

1. Інфраструктурна ІТ-складова (Архітектура розгортання): еталонна кіберфізична архітектура ІСД на основі Цифрового двійника. Вона не передбачає фізичного створення обладнання автором, а регламентує використання

конкретних платформ, хмарних сервісів, Edge-фреймворків та протоколів передачі даних (розглядається у підрозділі 4.2).

2. Програмна складова (Інструментальний засіб R-SIM): спеціалізований програмний симулятор, який виступає математичним ядром системи та безпосередньо автоматизує процеси синтезу МФММ (кроки 3-4), верифікації обчислень (крок 5) та візуалізації результатів для прийняття рішень (крок 6). Цей інструмент буде апробований у підрозділі 4.4 на МФММ4.

4.2 Інформаційна технологія та архітектура розгортання ІСД на основі цифрового двійника для промислового обладнання із використанням індустріального інтернету речей

4.2.1 Концепція еталонного розгортання та зв'язок з теоретичною моделлю

У підрозділі 2.1 було запропоновано теоретико-множинну модель та високорівневу логічну архітектуру інтелектуальної системи діагностування (ІСД) з функцією донавчання. Однак практична імплементація цифрового двійника для промислового обладнання (наприклад, верстатів з ЧПК чи циркуляційного насосу малих модульних реакторів) є багатовимірною інженерною задачею. Створення повномасштабного фізичного екземпляра такого ЦД у межах виробничого підприємства вимагає значних інвестицій, інтеграції з наявними SCADA-системами та виходить за межі обсягу даного дисертаційного дослідження.

З огляду на це, у даному підрозділі пропонується розроблена еталонна ІТ-архітектура розгортання інформаційної технології ІСД. Вона деталізує перехід від математичних моделей до більш деталізованої архітектури, інструментів, стандартів обміну даними, хмарних сервісів і патернів, дозволяючи іншим розробникам реалізувати та розгорнути запропоновану логіку донавчання на власній апаратній базі.

Вимоги до системи. Розроблення еталонної архітектури базується на визначенні функціональних та нефункціональних вимог, які регламентують поведінку обчислювальних вузлів та мережевої інфраструктури (таблиця 4.1).

Проекція теоретичної моделі на ІТ-архітектуру. Для забезпечення цілісності дослідження теоретико-множинна модель $IDS=\langle M,O,D,DT,L,E \rangle$ (формула 2.1) спроектована на реальні апаратно-програмні вузли. Відповідність між множинами та компонентами архітектури наведено у таблиці 4.2.

Таблиця 4.1 – Основні функціональні та нефункціональні вимоги до архітектури ІСД промислового ІоТ обладнання з донавчанням

Функціональні вимоги	Нефункціональні вимоги
Безперервний збір телеметрії з об'єкта діагностування	Затримки: Операційний контроль ≤ 10 мс; edge ≤ 200 мс; передача в хмару – асинхронна; федеративне донавчання – офлайн (години/дні)
Накопичення історії відмов	
Перерахунок готовності обладнання	
Виявлення та локалізація відмов засобами оперативного контролю і тестового діагностування	Надійність і QoS: підтвердження доставки (QoS 1/2), цілісність і впорядкованість даних, резервування MQTT-брокера
Формування і передачу керуючих впливів для коригування допусків параметрів і тестових послідовностей	Кібербезпека: Ізоляція контурів управління від зовнішніх мереж, TLS, X.509, захист каналів передачі ΔW (дельти ваг)
Забезпечення локального і федеративного навчання діагностичних моделей ЦД	Масштабованість: $\geq 10^3$ – 10^4 вузлів; до 10^5 – 10^6 msg/s; підключення нових вузлів без зміни ядра (UNS-парадигма)

Таблиця 4.2 – Проекція теоретико-множинної моделі ІСД на її архітектуру

Елемент теоретичної моделі (Розділ 2)	Реалізація в ІТ-архітектурі (Розділ 4)
1	2
Множина режимів роботи (М)	Блок керування (4) та Мультиплексор вхідних даних (2), що розміщені на польовому рівні (L0-L1). Вони діють як скінченний автомат, здійснюючи апаратну диспетчеризацію потоків телеметрії залежно від поточного режиму роботи.

1	2
Множина станів об'єкта (O)	Фізично у станах перебуває лише об'єкт діагностування (рівень 0–1). Поточний стан визначається Класифікатором технічного стану (DTedge) та використовується Sel для апостеріорного перерахунку у сервісі розрахунку готовності
Засоби діагностування (D)	Оперативний контроль (Oc) та Тестове діагностування (Td) реалізовані на рівні ПЛК/FPGA як замкнений діагностичний контур з реакцією до 10 мс. Адаптація допусків ΔP та послідовностей ΔT_{seq} надходить через Edge MQTT Broker.
Множина цифрового двійника (DT)	Розподілена сутність: DTedge (Inference Engine · ONNX Runtime, Flower Client · PyTorch, InfluxDB Edge/SQLite) на Edge для локальної адаптації; DTcloud (Flower ServerApp · SageMaker/Azure ML, граф топології · Azure Digital Twins/AWS IoT TwinMaker) у хмарі для глобальної агрегації. Рівні об'єднані через UNS (Central MQTT Broker · MQTT Bridge).
Процеси донавчання (L)	Flower Client (PyTorch) обчислює $\Delta W \rightarrow$ Flower SuperNode $\rightarrow$ SuperLink $\rightarrow$ Flower ServerApp агрегує (FedAvg/FedProx) $\rightarrow$ оновлена модель повертається по тому ж каналу до SuperNode $\rightarrow$ Flower Client. ΔP та ΔT_{seq} публікуються Класифікатором через Edge MQTT Broker до Oc/Td.
Оцінювання готовності (E)	Сервіс розрахунку готовності на Edge: апостеріорний вибір фрагментної матриці (Sel), розв'язок системи Колмогорова методом Рунге-Кутти (Calc), публікація $A(t)$ у UNS.

4.2.2 Структурна організація еталонної ІТ-архітектури

Розроблена еталонна ІТ-архітектура реалізує триярусну ієрархію обчислень: польовий рівень (Field, рівні 0–1), граничний рівень (Edge, рівень 2) та хмарний рівень (Cloud, рівні 3–4). Розподіл функцій між рівнями визначається компромісом між затримкою реакції та обчислювальною ємністю. Загальна структура архітектури наведена на рисунку 4.1.

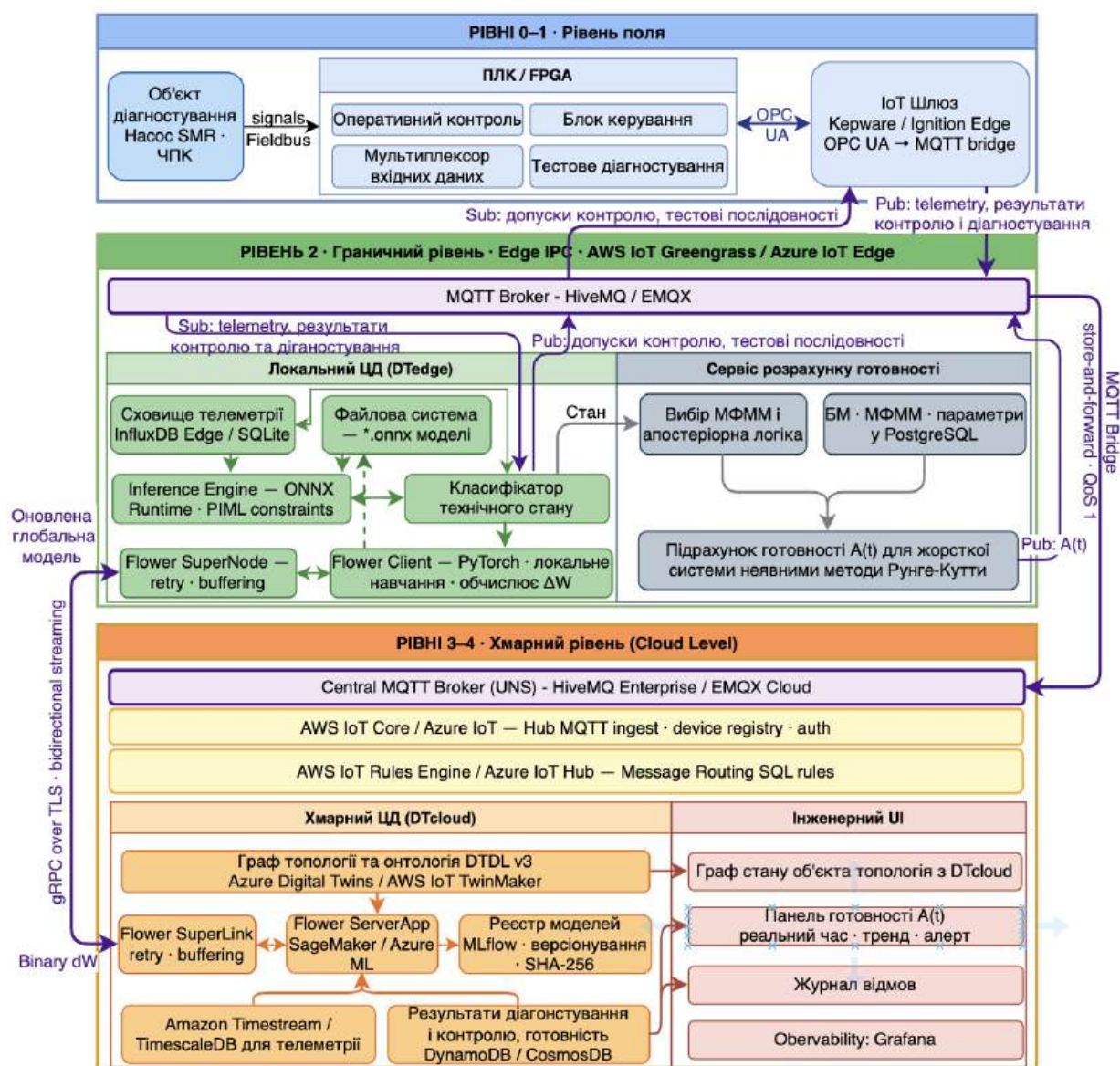


Рисунок 4.1 – Еталонна ІТ-архітектура ІСД промислового ІоТ обладнання на основі цифрового двійника із донавчанням

Польовий рівень (Field Level, рівні 0–1) утворює ізольований контур функціональної безпеки відповідно до IEC 61508. Об'єкт діагностування (насос SMR/ЧПК) передає сигнали сенсорів (Fieldbus) до контролера ПЛК/FPGA. У межах контролера функціонують чотири апаратні компоненти, що відповідають множині D теоретико-множинної моделі: блок оперативного контролю (Oc), блок тестового діагностування (Td), мультиплексор вхідних даних (Mx) та блок керування (Ctrlsys). Реакція замкненого діагностичного контуру на цьому рівні не

перевищує 10 мс. IoT-шлюз (Kepware / Ignition Edge) транслює дані з OPC UA у MQTT та передає їх до Edge MQTT брокера.

Граничний рівень (Edge Level, рівень 2) розгортається на промисловому комп'ютері (Edge IPC – Industrial PC) безпосередньо поруч з об'єктом і реалізує два паралельних шляхи обробки даних.

1. Hot Path – мілісекундна петля адаптації. Edge MQTT брокер (EMQX / HiveMQ) доставляє телеметрію до локального цифрового двійника (DTedge). Inference Engine на базі ONNX Runtime виконує inference з урахуванням фізично-інформованих обмежень (PIML constraints) та передає результат до класифікатора технічного стану, який визначає поточний стан об'єкта з множини $\{Su, Sh, Sd\}$ та публікує оновлені допуски ΔP і тестові послідовності ΔT_{seq} назад до Oc/Td через Edge MQTT брокер. Повний цикл Hot Path від отримання телеметрії до публікації оновлених допусків не перевищує 200 мс (відповідно до вимог, таблиця 4.1).

2. Cold Path – асинхронна передача накопичених даних. Класифікатор технічного стану записує оброблену телеметрію та діагностичні події до локального сховища (InfluxDB Edge / SQLite), збагачуючи кожен запис поточним станом об'єкта. Сховище виконує функцію буфера Cold Path для подальшої передачі до хмари через MQTT Bridge за наявності мережевого з'єднання.

Окремим сервісом є сервіс розрахунку готовності, що функціонує на Edge IPC. Він отримує від класифікатора ідентифікований стан, виконує апостеріорний вибір фрагментної матриці (Sel), розв'язує систему диференціальних рівнянь Колмогорова-Чепмена неявним методом Рунге-Кутти та публікує значення готовності $A(t)$ у MQTT.

Федеративне навчання на Edge рівні реалізується компонентами Flower Client (PyTorch) та Flower SuperNode. Після ідентифікації прихованої відмови або отримання нової версії глобальної моделі Flower Client виконує локальне донавчання, обчислює дельти вагових коефіцієнтів ΔW та передає їх через Flower SuperNode до хмари по каналу gRPC з TLS 1.3.

Хмарний рівень (Cloud Level, рівні 3–4) відповідає за ресурсоємну агрегацію знань від усього парку обладнання. Центральний (Central) MQTT

брокер (HiveMQ Enterprise / EMQX Cloud) приймає повідомлення від Edge рівня через MQTT Bridge зі збереженням гарантії доставки QoS 1. AWS IoT Core / Azure IoT Hub виконує функції шлюзу хмарного рівня: аутентифікацію пристроїв на основі X.509-сертифікатів та прийом повідомлень від Central MQTT брокера для подальшої маршрутизації. AWS IoT Rules Engine / Azure IoT Hub Message Routing маршрутизує потоки за типом: часові ряди телеметрії направляються до сховища часових рядів (Amazon Timestream / TimescaleDB), структуровані діагностичні події – до документно-орієнтованого сховища (DynamoDB / CosmosDB), поточні значення показника готовності $A(t)$ – до інженерного UI через WebSocket.

Глобальний цифровий двійник (DTcloud) реалізує чотири функціональні підсистеми. Підсистема онтологічного представлення об'єкта забезпечує зберігання графу топології та семантичних зв'язків між вузлами обладнання засобами мови DTDL v3 (Azure Digital Twins / AWS IoT TwinMaker). Підсистема федеративного навчання (Flower ServerApp, SageMaker / Azure ML) виконує агрегацію дельт вагових коефіцієнтів ΔW від усіх екземплярів DTedge за алгоритмом FedAvg/FedProx та формує оновлену глобальну модель. Підсистема версіонування моделей (MLflow) забезпечує реєстрацію артефактів із верифікацією цілісності на основі SHA-256, керування життєвим циклом моделі (Staging $\rightarrow$ Production $\rightarrow$ Archived). Оновлена глобальна модель розповсюджується на Edge рівень по тому самому двонаправленому gRPC-каналі (Flower SuperLink $\rightarrow$ SuperNode $\rightarrow$ Flower Client), де Edge верифікує SHA-256 підпис перед завантаженням вагових коефіцієнтів до середовища виконання.

Інженерний UI реалізує операторський інтерфейс моніторингу стану ІСД. Панель готовності відображає поточне значення $A(t)$ у режимі реального часу через WebSocket-з'єднання з підпискою на відповідний топик. Граф стану об'єкта візуалізує топологію з DTcloud із відображенням поточного стану кожного вузла. Журнал відмов забезпечує ретроспективний аналіз відмов з прив'язкою до версії моделі та раунду федеративного навчання. Підсистема технічної спостережуваності (Observability) забезпечує моніторинг стану інфраструктурних

компонентів – доступності сервісів, затримок MQTT Bridge, метрик FL-раундів засобами Prometheus / Grafana.

4.2.3 Модель даних та інтерфейси взаємодії

Єдиний простір імен (Unified Namespace, UNS) реалізує централізовану шину обміну повідомленнями на базі протоколу MQTT v5 (ISO/IEC 20922) з топологією publish/subscribe. UNS виступає основним інтеграційним шаром між Edge та Cloud рівнями: компоненти від IoT-шлюзу до хмарних сервісів взаємодіють через публікацію та підписку на топіки, що забезпечує слабку зв'язаність компонентів та можливість горизонтального масштабування парку обладнання без зміни архітектури. Взаємодія на польовому рівні (ПЛК → IoT-шлюз) здійснюється через OPC UA (IEC 62541) з подальшою трансляцією у MQTT. Канал федеративного навчання реалізований окремо – через двонаправлений gRPC-стрім (Flower SuperNode ↔ SuperLink, TLS 1.3), оскільки передача бінарних артефактів моделей не відповідає семантиці MQTT.

Ієрархічна структура топіків UNS побудована за схемою `plant/{site}/asset/{id}/{signal}` і наведена у таблиці 4.3. Кожен топік має визначеного публікувача та одного чи кількох підписників.

Таблиця 4.3 – Структура топіків Єдиного простору імен (MQTT Namespace)

Топік MQTT	Публікує	Підписується	Опис
1	2	3	4
<code>plant/{site}/asset/{id}/telemetry</code>	IoT Шлюз	Класифікатор технічного стану, Rules Engine (Cloud)	Сигнали сенсорів
<code>plant/{site}/asset/{id}/control/oc_results</code>	Ос (ПЛК)	Класифікатор технічного стану	Результати оперативного контролю

1	2	3	4
plant/{site}/asset/{id}/ control/td_results	Td (ПЛК)	Класифікатор технічного стану	Результати тестового діагностування
plant/{site}/asset/{id}/ control/tolerances	Класифікатор (DTedge)	Ос (ПЛК)	Оновлені допуски ΔP
plant/{site}/asset/{id}/ control/test_sequences	Класифікатор (DTedge)	Td (ПЛК)	Скориговані тестові послідовності ΔTseq
plant/{site}/asset/{id}/ diagnostic_event	Класифікатор (DTedge)	Сервіс розрахунку готовності, Rules Engine (Cloud)	Діагностична подія {state, component, ts, confidence}
plant/{site}/asset/{id}/ availability	Сервіс розрахунку готовності (DTedge)	Rules Engine (Cloud) → UI (WebSocket)	Поточне значення A(t) {A_t, ts, model_ver}

Ключові міжрівневі інтерфейси та структури даних наведені у таблиці 4.4. Вибір протоколу для кожного інтерфейсу визначався компромісом між гарантіями доставки, затримкою та типом даних, що передаються.

Описана архітектура є еталонною – вона визначає інтерфейси, протоколи та топологію взаємодії компонентів, але не прив'язує реалізацію до конкретного хмарного провайдера чи інструмента. Вибір між AWS IoT Core та Azure IoT Hub (чи аналогами інших хмарних провайдерів), між Amazon Timestream та TimescaleDB, між EMQX та HiveMQ визначається наявною інфраструктурою підприємства та вимогами до даних. Незмінними залишаються ізоляція польового контуру безпеки (IEC 61508), єдиний простір імен як єдина точка інтеграції між рівнями, та двонаправлений gRPC-канал федеративного навчання як єдиний механізм передачі вагових коефіцієнтів між DTedge і DTcloud.

Таблиця 4.4 – Ключові інтерфейси та структури даних

Інтерфейс	Протокол	Напрямок	Структура даних	Примітка
Field → Edge (OPC UA)	OPC UA(IEC 62541)	ПЛК → IoT Шлюз	{node_id, value, quality, timestamp}	Режим опитування (polling) або підписки
Field (OPC UA) → Edge (MQTT)	MQTT v5, QoS 1	IoT Шлюз → Edge Broker	{asset_id, signals[], ts_unix_ms}	Трансляція OPC UA → MQTT на шлюзі
Edge → Cloud (telemetry)	MQTT Bridge, QoS 1	Edge Broker → Central Broker	{asset_id, signals[], ts_unix_ms}	TLS 1.3
Edge → Cloud (Федеративне навчання – завантаження дельт ваг)	gRPC по TLS 1.3	SuperNode → SuperLink	{tensors: bytes[], num_examples: int, metrics: map}	Flower, двонаправлений gRPC стрім
Cloud → Edge Завантаження глобальної моделі	gRPC по TLS 1.3	SuperLink → SuperNode	{ parameters: {tensors: bytes[]}, config: {round: int, model_ver: str, sha256: str} }	gRPC стрім. Edge верифікує SHA-256 перед завантаженням
Edge → Cloud (Відправка A(t))	MQTT v5, QoS 1	розрахунок готовності → Edge Broker → Central Broker	{A_t: float, ts: datetime, model_ver: str, state: Su/Sh/Sd}	Топік: plant/{site}/asset/{id}/availability
UI / Observability	REST API + WebSocket	Rules Engine → UI	{asset_id, A_t, state, trend[], alerts[]}	WebSocket для live A(t), REST для журналу відмов

4.3 Програмний засіб «R-SIM» для оцінювання надійності промислового обладнання систем індустриального інтернету речей з використанням ІСД та донавчанням

З метою автоматизації розроблених у розділах 2 та 3 математичних моделей та методу, а також для впровадження їх у практику проектування систем індустриального інтернету речей, було розроблено спеціалізований програмний

засіб «R-SIM» (Reliability Simulator). Цей інструмент є системою підтримки прийняття рішень для інженерів з надійності.

Програмний засіб призначений для оцінювання готовності промислового обладнання систем індустріального інтернету речей (наприклад, компонентів управління малих модульних реакторів SMR, високоточних верстатів з ЧПК), яке використовує інтелектуальні системи діагностування (ІСД) на основі цифрових двійників із функціями локального або федеративного донавчання.

Основним призначенням інструменту є вирішення двох ключових проблем моделювання таких систем. По-перше, він автоматизує побудову складних мультифрагментних марковських моделей (МФММ) на основі результатів розділу 2.2, позбавляючи дослідника необхідності ручного формування матриць при зміні показників достовірності діагностування. По-друге, інструмент автоматично вирішує проблему математичної «жорсткості» за методом, описаним у розділі 3.4.

4.3.1 Проектування архітектури та опис функціоналу

Для формалізації вимог до програмного засобу, опису логіки перетворення вхідних даних та взаємодії його компонентів було використано методологію функціонального моделювання IDEF0. Розроблена ієрархічна модель дозволяє простежити зв'язок між теоретичними результатами дисертації та їх програмною реалізацією.

На контекстній діаграмі рівня А-0 (рис. 4.2) відображено загальний процес оцінювання готовності. На вхід системи подаються параметри моделювання (інтенсивності відмов, відновлення, початкова достовірність діагностування і т.д.) та обрана базова модель об'єкта. Керуючими впливами, які регламентують роботу алгоритмів, виступають отримані у попередніх розділах наукові результати: спосіб технічного діагностування (НР1), класифікатор та комплекс базових моделей (НР2), а також метод верифікації та валідації (НР3). Механізмами реалізації процесу виступають модулі програмного засобу: графічний інтерфейс

(RSIM UI) та обчислювальне ядро (RSIM core). На виході формуються функція готовності $A(t)$, звіт про верифікацію математичних обчислень та інтерпретація вихідних показників у вигляді певних рекомендацій щодо впровадження донавчання в ІСД при таких вхідних даних.

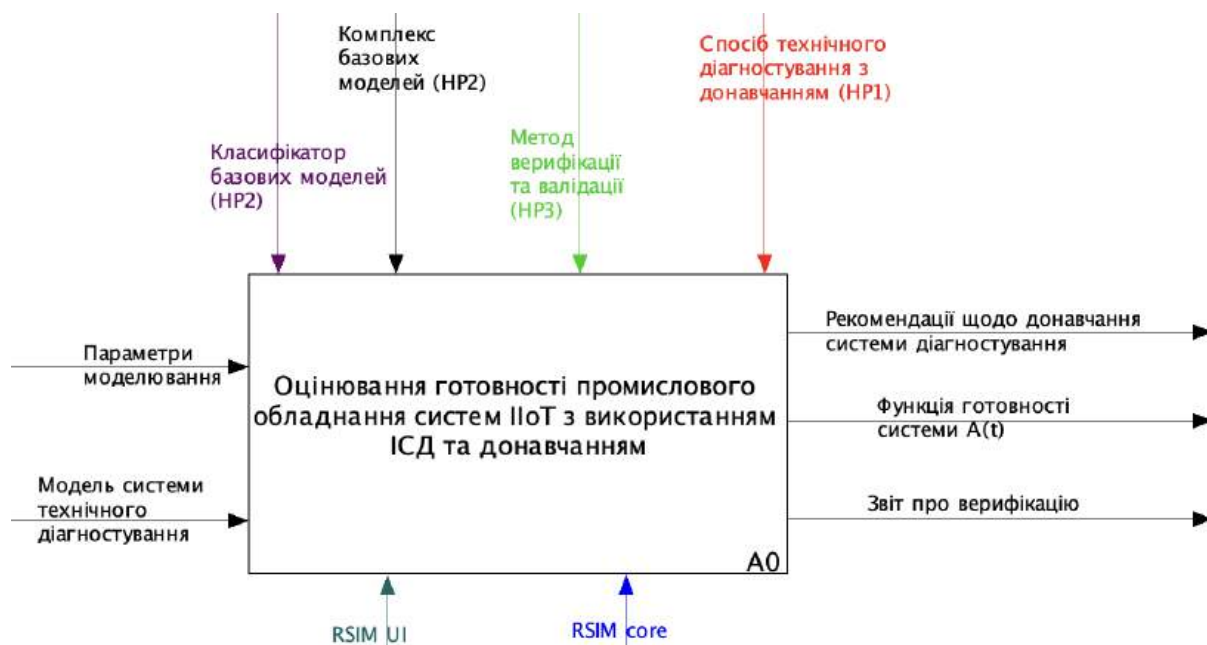


Рисунок 4.2 – Функціональне моделювання «R-SIM», ієрархічна діаграма IDEF0 рівня A-0

Декомпозиція діаграми на рівні A0 (рис. 4.3) розкриває внутрішню послідовність обробки даних, що складається з таких чотирьох основних етапів.

1. Обрати модель та параметри (Блок A1): Користувач взаємодіє з класифікатором (HP2) для вибору базової моделі та параметри в залежності від ознак об'єкта діагностування (врахування обслуговування, різних причин донавчання).

2. Синтезувати та валідувати матрицю інтенсивностей Q (Блок A2): Обчислювальне ядро автоматично генерує простір станів та перехідні коефіцієнти. На цьому етапі розраховується індекс жорсткості, який як керуючий параметр передається на наступні етапи для вибору методу розв'язання диференціальних рівнянь. Слід зазначити, що будується Q для кожної з моделей у порівняльній тріаді.

3. Просимулювати (Блок A3): Виконується безпосереднє розв'язання системи диференціальних рівнянь. Використовуючи дані про жорсткість матриці, система отримує динаміку ймовірностей перебування системи у різних станах на в різні моменти часу. Також розраховується стаціонарна готовність.

4. Проаналізувати результати симуляції та налаштувань моделі (Блок A4): Здійснюється фінальна обробка отриманих масивів даних, їх верифікація та генерація звітів.

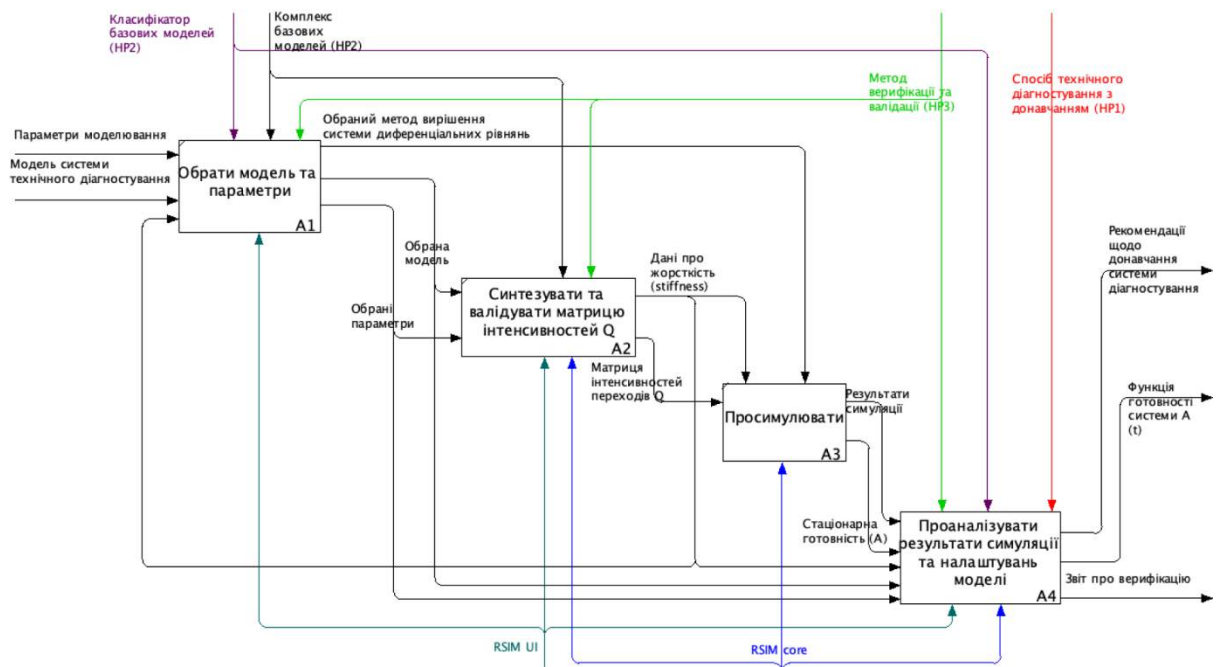


Рисунок 4.3 – Функціональне моделювання «R-SIM», ієрархічна діаграма IDEF0 рівня A0

Додатково декомпозиємо етап аналізу до рівня A4 (рис. 4.4). Етап рівня A4 включає підрахунок підсумкової функції готовності $A(t)$ (Блок A41), обов'язкову верифікацію результатів симуляції (Блок A42), де результати чисельного моделювання звіряються з аналітичними розв'язками згідно з методом HP3, а також формування KPI-звіту та інтерпретацію результатів (Блок A43), де система генерує висновок про доцільність донавання з огляду на втрати часу та приріст достовірності діагностування.

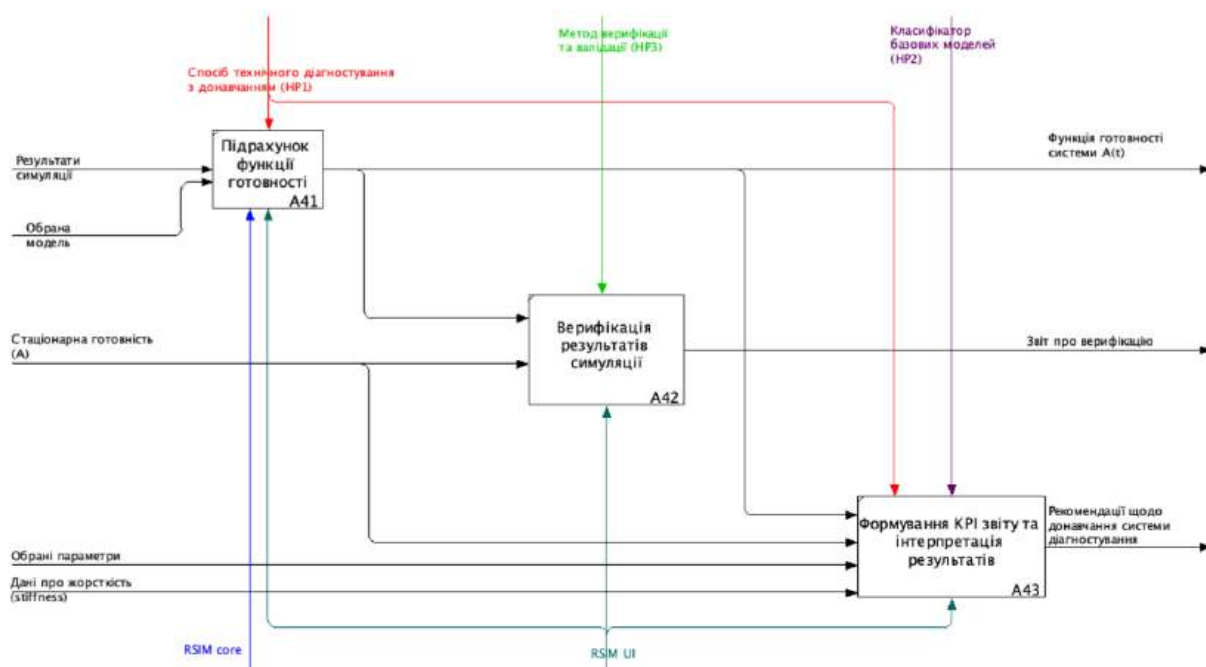


Рисунок 4.4 – Функціональне моделювання «R-SIM», ієрархічна діаграма IDEF0 рівня A4.

Архітектура програмного засобу R-SIM. Для реалізації описаної функціональної моделі архітектуру програмного засобу R-SIM (рис. 4.5) було побудовано за принципом розділення відповідальності. Вона складається з наступних трьох логічних блоків.

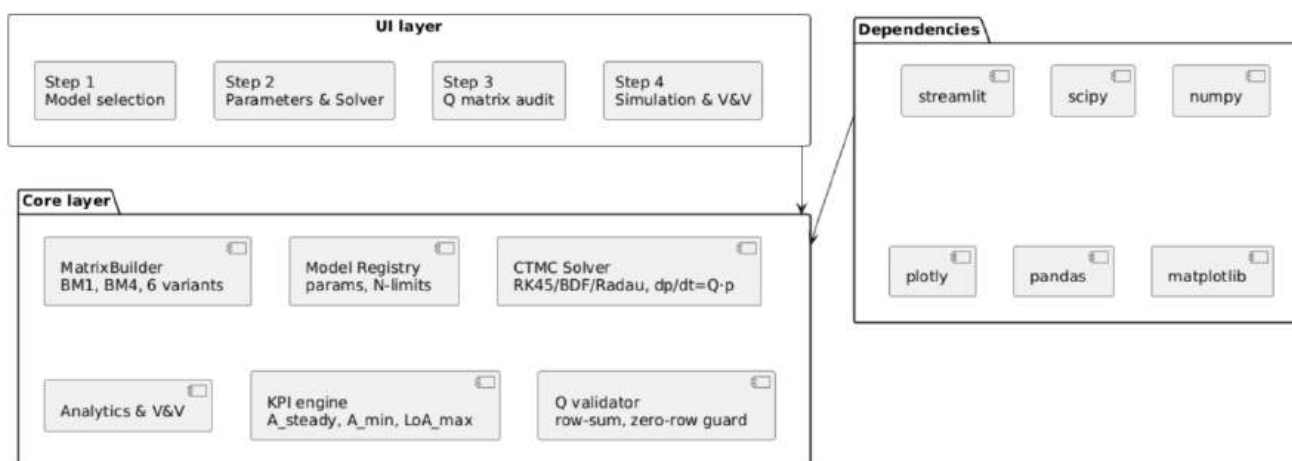


Рисунок 4.5 – архітектура «R-SIM»

1. Інтерфейс користувача (R-SIM UI). Реалізований як клієнт за допомогою Python-фреймворку Streamlit. Вибір цього інструменту зумовлений

можливістю швидкого створення інтерактивних інженерних веб-дашбордів без необхідності розроблення складного розділеного фронтенду та бекенду. Він відповідає виключно за інтерактивну взаємодію з користувачем через покрокову конфігурацію (wizard) та динамічну візуалізацію дашбордів. Інтерфейс ізолює інженера від математичної складності, приймаючи експлуатаційні параметри та відображаючи кінцеві KPI та статуси верифікації. Має 4 кроки: вибір моделі, вибір параметрів і методу розв'язання диференціальних рівнянь, верифікація матриці інтенсивностей переходів Q , симулювання.

2. Обчислювальне ядро (R-SIM core). Окремий рівень додатку, у якому зосереджена основна бізнес-логіка, що відповідає за математичні розрахунки та моделювання. Виконує побудову матриці інтенсивностей переходів Q , верифікує її (сума елементів кожного рядка нульова: $\sum_{j=1}^n Q_{ij} = 0$), обчислює жорсткість, розв'язує систему диференціальних рівнянь обраним способом, обчислює ключові метрики, обчислює стаціонарне значення готовності. Ці операції виконуються для кожної моделі з порівняльної тріади. Також містить компоненти для перевірки чи готовність сходиться з аналітичним розв'язком ($|A_{sim} - A_{an}| \leq \varepsilon$, де $\varepsilon = 0,01$). Виконує інтерпретацію метрик. Розв'язання диференціальних рівнянь виконується у функції `solve_ivp` пакету SciPy, конфігуруючи параметр методу розв'язання залежно від визначеного індексу жорсткості (RK45, BDF або Radau).

3. Залежності (python пакети з бібліотеками). Для роботи веб-інтерфейсу (UI) використовується фреймворк Streamlit. Для математичного забезпечення використовуються наступні бібліотеки: NumPy для матричних операцій та SciPy для лінійної алгебри і розв'язання систем рівнянь необхідним методом. Своєю чергою, структурування масивів результатів та генерації графіків і схеми базових моделей реалізується за допомогою бібліотек Pandas, Plotly та Matplotlib.

4.3.2 Приклади роботи та особливості застосування програмного засобу

Процес оцінювання готовності промислового обладнання в середовищі ІоТ з донавчанням ІСД у розробленому програмному засобі «R-SIM» реалізовано у

вигляді інтерактивного покрокового алгоритму (wizard). Інтерфейс складається з чотирьох послідовних етапів.

Крок 1. Вибір базової моделі. На першому етапі користувач здійснює вибір базової моделі (з комплексу БМ1–БМ5) відповідно до розробленого у підрозділі 2.2 комплексу (рис. 4.6). Головною особливістю програмного засобу є те, що для забезпечення об'єктивного порівняльного аналізу система автоматично формує та готує до симуляції відразу порівняльну тріаду моделей: ідеальну (Ideal), статичну без навчання (NoLearning) та динамічну із донавчанням (Learning). На екрані динамічно візуалізуються структурні графи обраної тріади.

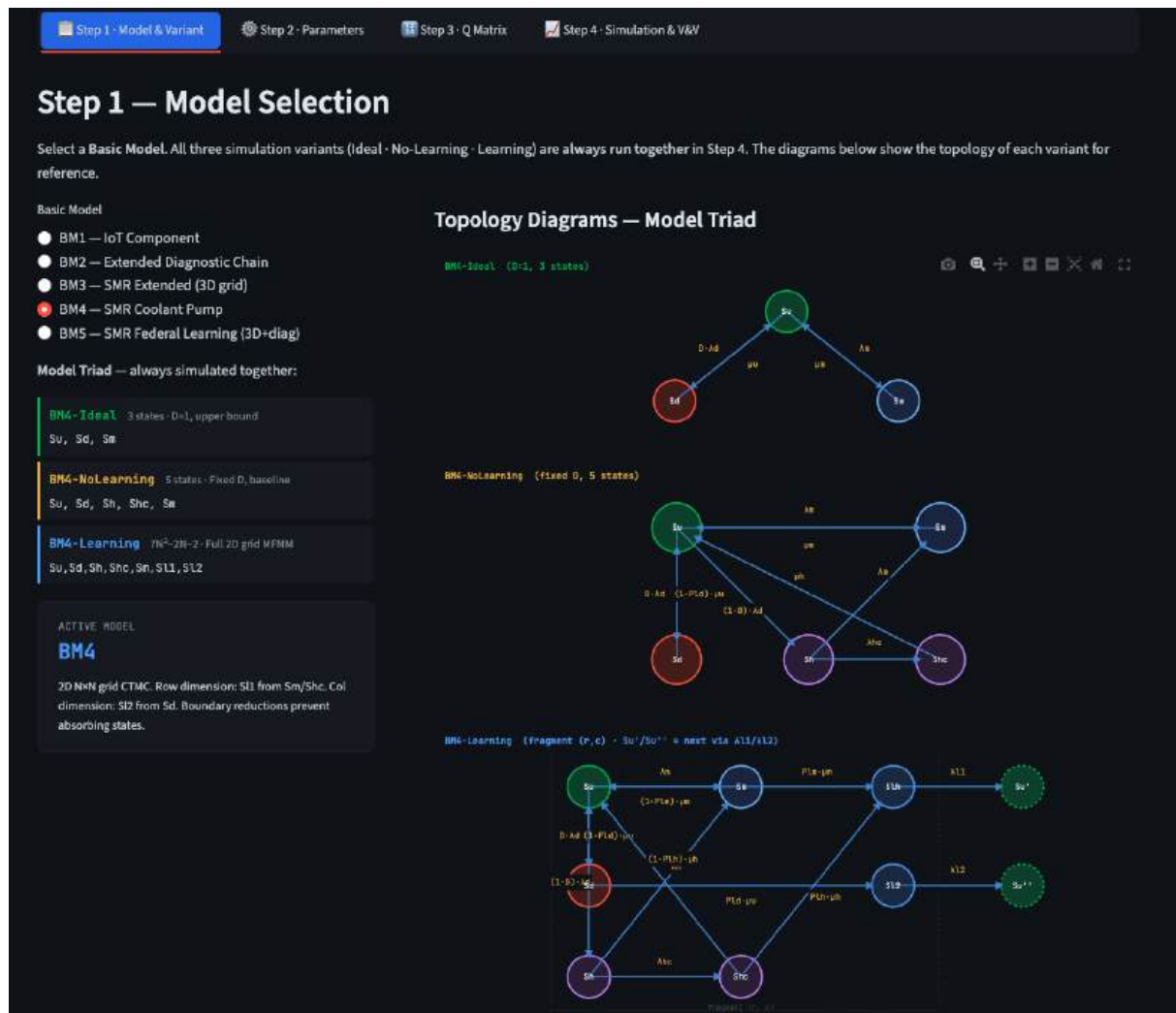


Рисунок 4.6 – Крок 1, вибір базової моделі (BM4)

Крок 2. Налаштування параметрів та обґрунтування інструментарію.

Другий крок призначений для введення експлуатаційних параметрів обладнання: інтенсивностей виникнення відмов (λ_d), ремонту (μ_r), показників донавання (λ_l) початкову D , та інших (рис. 4.7). На цьому етапі реалізується метод вибору чисельного методу розв'язання диференціальних рівнянь (НРЗ). Аналізуються введені параметри, розраховується індекс математичної жорсткості $s(x)$ і автоматично призначає оптимальний інструмент розв'язання диференціальних рівнянь (RK45 для нежорстких систем, або BDF/Radau для жорстких).

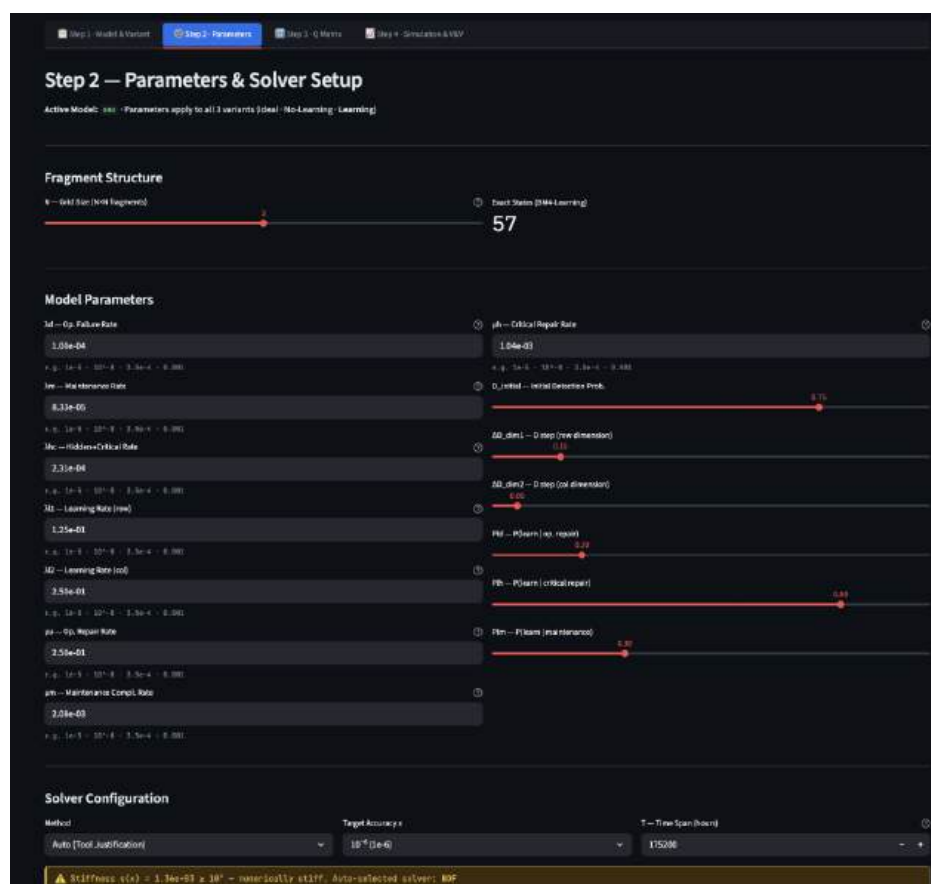


Рисунок 4.7 – Крок 2, параметризація для VM4, врахування жорсткості

Крок 3. Верифікація матриць інтенсивностей переходів Q . Перед запуском симуляцій програма здійснює верифікацію матриці інтенсивностей переходів Q : рівність суми елементів кожного рядка нулю та відсутність ізольованих нульових рядків. Згенеровані матриці виводяться на екран у вигляді

таблиць для візуального контролю експертом та можуть бути експортовані для подальшого зовнішнього аналізу (рис. 4.8).

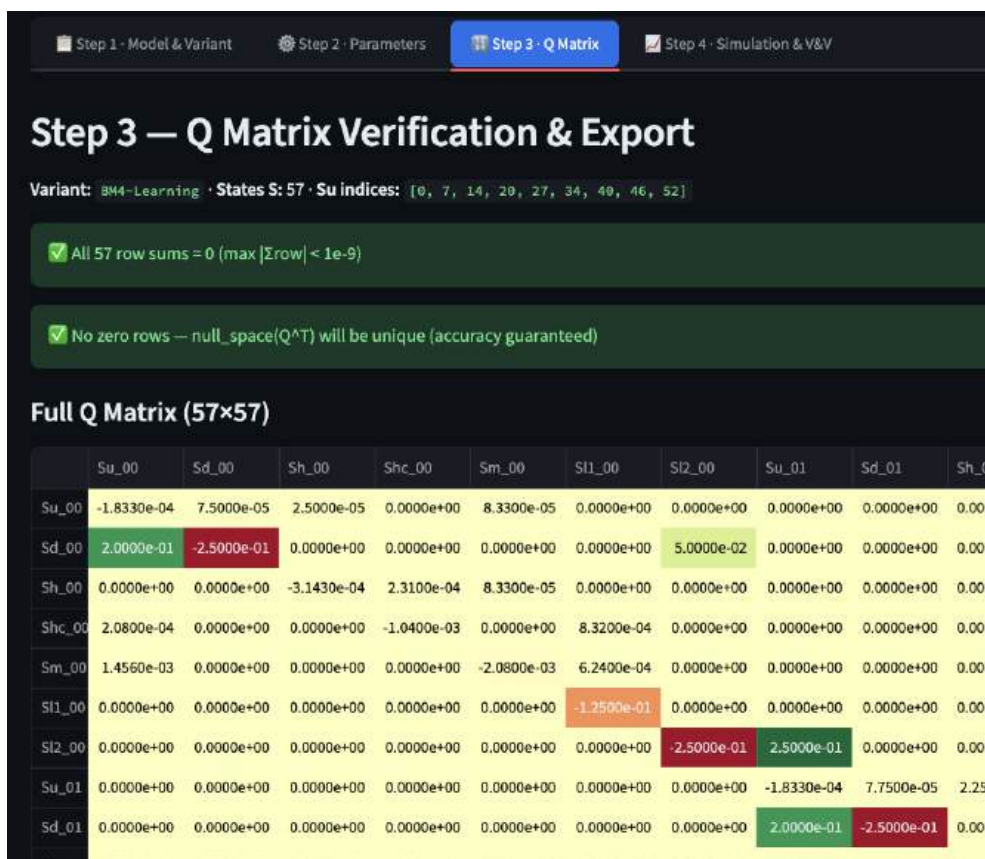


Рисунок 4.8 – Крок 3, верифікація Q для VM4 та її експорт

Крок 4. Симуляція, перехресна верифікація та генерація висновків.

Четвертий фінальний крок ініціює розв'язання задачі та інтерпретацію результатів. Інструмент будує динаміку функцій готовності $A(t)$ для порівняльної тріади моделей. Вираховує ключові метрики аналогічні розглянутим у розділі (рис. 4.9). Ще на цьому кроці виконується такі 2 функції (рис. 4.9, рис. 4.10).

1. В автоматичному режимі порівнюється результат, отриманий чисельним методом (симуляцією), із точним аналітичним стаціонарним розв'язанням ($\pi Q=0$). У разі математичної збіжності система видає підтвердження (статус «converged»), що показує, чи достатній проміжок для симулювання було обрано, щоб побачити динаміку на всіх етапах.

2. Підсистема розраховує п'ять ключових метрик (Asteady, Amin, LoAmax, Tloss, Tstd), порівнює криві статичної моделі та моделі з донавчанням, і генерує кількісно текстовий висновок. Сигналізує якщо готовність ІСД з донавчанням немає покращень у стаціонарному значенні, або якщо мінімально перевищує, чи є суттєвий програш на старті (рис. 4.10).

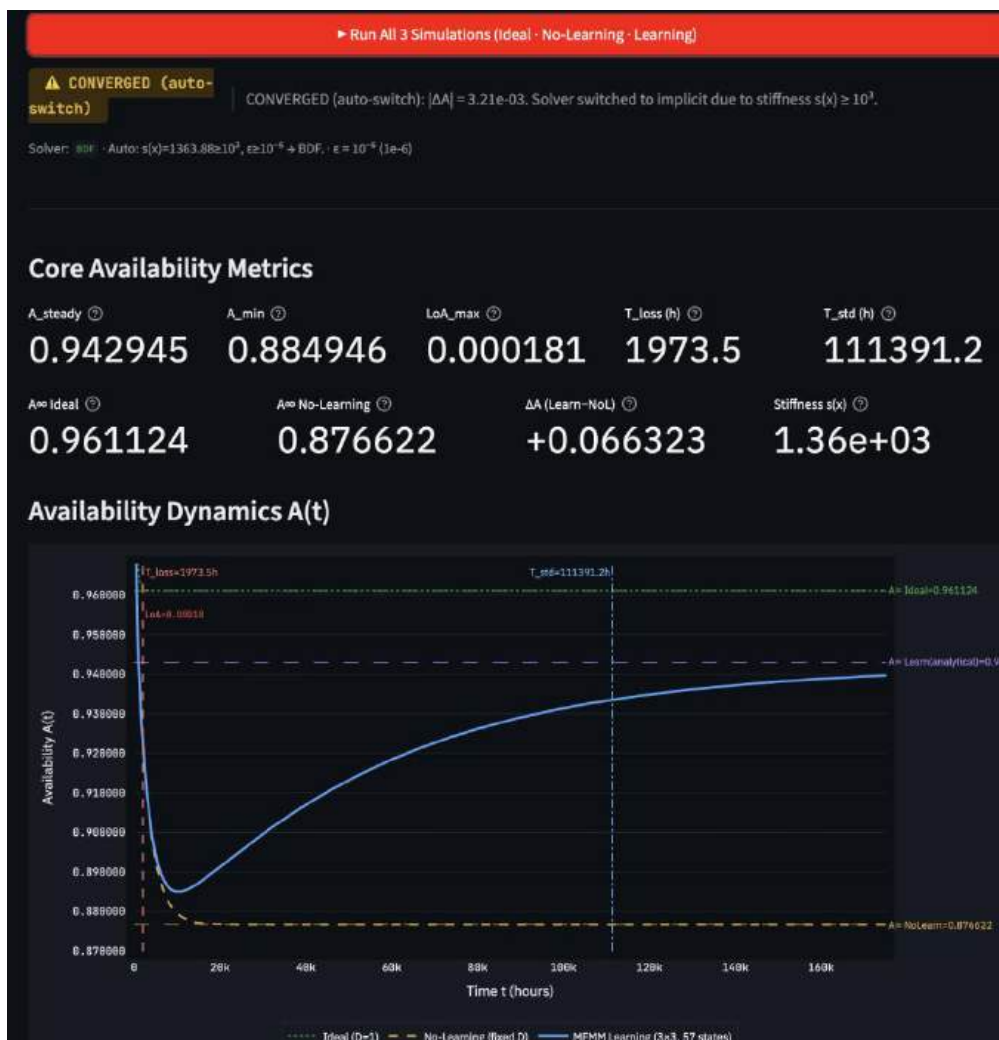


Рисунок 4.9 – Крок 4, приклад симуляції результатів для BM4, розрахунок метрик

Таким чином, розроблений програмний засіб «R-SIM», спираючись на отримані в роботі теоретичні моделі (НР2) та методи верифікації (НР3), дозволяє інженерам із надійності обґрунтовувати технічні рішення щодо доцільності розгортання ІСД на основі цифрових двійників із донавчанням для конкретних зразків промислового обладнання з ІІТ.

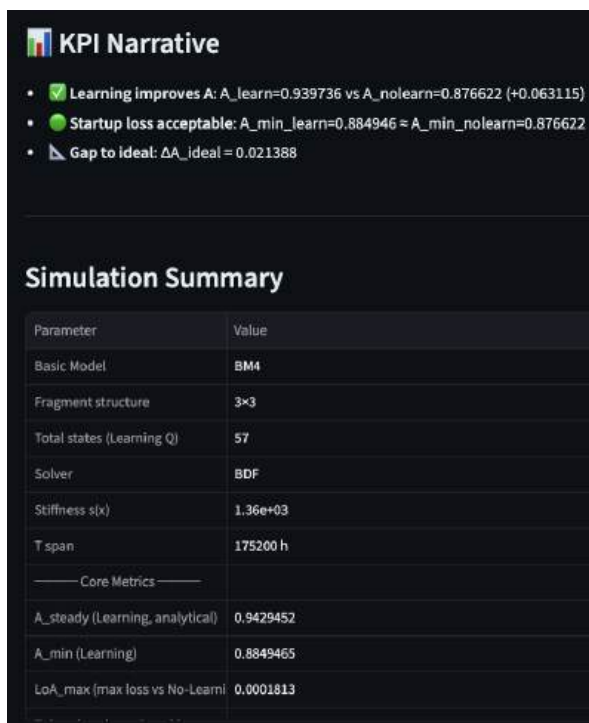


Рисунок 4.10 – Крок 4, приклад симуляції для BM4, звіт по метриках

4.4 Апробація МФММ4 та програмного засобу для обґрунтування надійності обладнання малих модульних реакторів із ІСД з донавчанням

Для практичної апробації однієї з розроблених марковських моделей та програмного засобу «R-SIM», як об'єкт дослідження обрано обладнання малих модульних реакторів (Small Modular Reactors, SMR), зокрема циркуляційний насос. Таке обладнання є репрезентативним прикладом критичних кіберфізичних систем, де інтеграція технологій індустріального інтернету речей (IIoT) в інформаційно-керуючі системи (I&C) відіграє важливу роль у забезпеченні безпеки та готовності системи [1].

Перспективним інструментом для забезпечення цієї мети є технологія цифрових двійників (ЦД). Як було детально розглянуто у нашій роботі [1], ЦД здатний не лише виконувати функції предиктивного технічного обслуговування чи донавчати засоби діагностування, але й виступати безпечним віртуальним полігоном для безперервної верифікації та валідації (V&V) оновлених алгоритмів керування і діагностування без ризику для фізичного обладнання. Особливого

значення ці функції набувають на етапі експлуатації V-подібного життєвого циклу систем I&C (рис. 4.11), що є частиною запропонованої у комплексної методології життєвого циклу цифрових двійників [1].

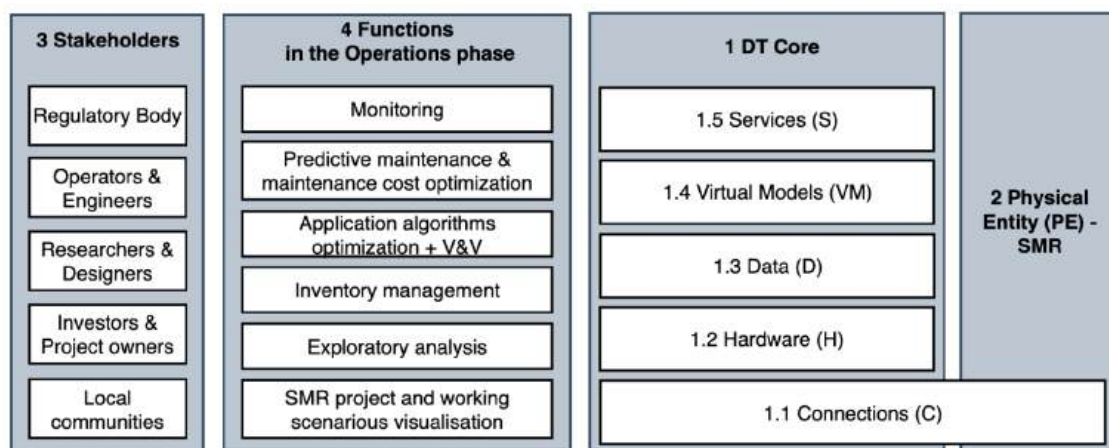


Рисунок 4.11 – Структура ЦД I&C, стейкхолдери та можливі функції виконання таким цифровим двійником

Крім того, на відміну від традиційних АЕС, концепція SMR передбачає перехід до економіки серійного виробництва та розгортання реакторів у вигляді парку (флоту) стандартизованих модулів. Це створює передумови для обміну телеметричними даними та донавчання алгоритмів цифрового двійника на основі досвіду усього флоту [2]. Відповідно, для моделювання інтелектуальної системи діагностування такого обладнання найбільш доцільним є використання матричної мультифрагментної Базової моделі 4 (БМ4) і відповідної МФММ4 (рис. 4.12). Ця модель найкраще відповідає специфіці експлуатації компонентів SMR, оскільки враховує періодичне ТО, ризики ескалації прихованої відмови до аварії, а також дозволяє врахувати різні причини донавчання цифрового двійника.

Параметризація. Через відсутність довгострокової статистики експлуатації SMR (так званий «deployment gap»), використовувалися емпіричні проксі-дані щодо надійності циркуляційних насосів (герметичного типу) діючих реакторів покоління та доступні специфікації проектів SMR (наприклад, BWRX-300, NuScale) [3-7]. Так, відповідно до досліджень, середній час між відмовами (MTBF) циркуляційних насосів першого контуру становить близько 100 000 годин

(~11 років) [3], тоді як середній час ремонту (MTTR) коливається в межах від 100 до 500 годин [3, 4]. Інтенсивність планового технічного обслуговування (ТО) жорстко узгоджується з циклами перевантаження палива, які для проекту BWRX-300 становлять 12 місяців [5], а для модулів NuScale – 21 місяць [6]. При цьому тривалість планової зупинки на обслуговування приймається на рівні 10 днів [6].

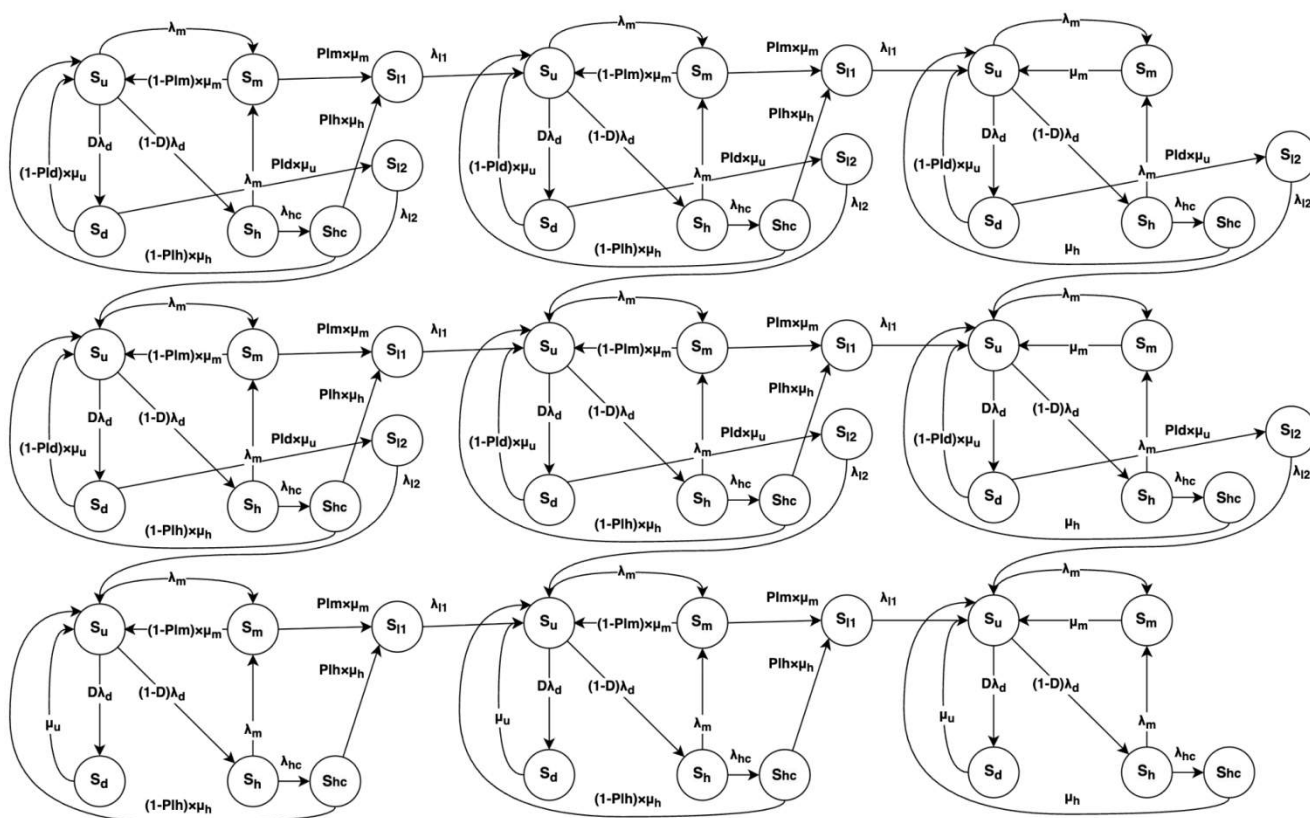


Рисунок 4.12 – МФММ4 для 3-х ітерацій донавчання по кожному з її видів

Окремо параметризовано переходи, пов'язані з виникненням та ескалацією прихованих відмов. Згідно з галузевими звітами щодо експлуатації насосного обладнання (зокрема, щодо деградації ущільнень), час ескалації латентного дефекту до критичного стану сильно варіюється і може тривати від кількох годин до тижнів або місяців [7]. У разі неможливості виявлення дефекту та його переходу у критичну відмову, час аварійного відновлення обладнання (без попередньої підготовки) суттєво зростає і може сягати 33–37 днів [4]. На основі зазначених даних розраховано базові інтенсивності переходів та ймовірнісні параметри, які зведено у таблицю 4.5.

Таблиця 4.5 – Базові значення параметрів для симуляції насоса SMR

№	Параметр	Значення	Обґрунтування
1	D	Dstart: 0.75-0.95 Dmax = 0.95-0.99	Достовірність діагностування. Початкова ймовірність (Dstart) лежить у межах 0.75–0.95 і поступово еволюціонує до максимального рівня D =1. Також ΔD після S11 у 3 рази більше ніж після S12.
2	Pld	0.1-0.3	Ймовірність потреби у донавчанні після явної відмови. Лише зрідка після штатного виявлення відмови не вистачає діагностичної інформації, що вимагає перенавчання. Або щось змінюється суттєве в конфігурації
3	Plm	0.1-0.4	Ймовірність потреби у донавчанні після ТО. Технічне обслуговування найчастіше завершується з відомим результатом і рідко вимагає оновлення моделі.
4	Plh	0.7-1.0	Ймовірність потреби у донавчанні після критичної прихованої відмови. Такі відмови майже завжди є новими деградаційними патернами і вимагають обов'язкового донавчання ЦД.
5	μ_u	1/100 – 1/500 (1/год)	Інтенсивність ремонту (відновлення). Середній час ремонту насоса після виявленої відмови становить від 100 до 500 годин [3, 4].
6	μ_h	1/480 – 1/960 (1/год)	Інтенсивність ремонту після критичної прихованої відмови. Аварійне відновлення без попередньої підготовки є складнішим і займає від 20 до 40 днів [4].
7	μ_m	1/120 – 1/480 (1/год)	Інтенсивність відновлення після планового ТО. Регламентні роботи тривають від 5 до 20 днів, з усередненим значенням близько 10 днів [6].
8	λ_d	1/50 000 – 1/150 000 (1/ год)	Інтенсивність відмов. Середній час між відмовами (MTBF) циркуляційних насосів становить близько 100 000 годин або ~11 років [3].
9	λ_m	1/8640 – 1/17280 (1/ год)	Інтенсивність проведення планового ТО. Зазвичай жорстко прив'язана до циклу перевантаження палива, що становить 12–24 місяці [5, 6].
10	λ_{hc}	1/6 – 1/4320 (1/ год)	Інтенсивність ескалації прихованої відмови. Час переходу прихованого дефекту в критичний аварійний стан сильно варіюється і може складати від 6 годин до 6 місяців [7].
11	λ_{l1}	1/8 (1/ год)	Інтенсивність донавчання після прихованих відмов. Оновлення алгоритмів після пізно виявлених дефектів є складнішим і займає 4–8 годин.
12	λ_{l2}	1(1/ год)	Інтенсивність донавчання після явних відмов. Адаптація моделі внаслідок випадкового виявлення або браку інформації є швидшою (1–4 години).

Симуляційне моделювання у програмному засобі «R-SIM». Викликом при моделюванні такого складного обладнання SMR як циркуляційний насос є математична «жорсткість» системи. Для вирішення проблеми при виконання симулюванні на довгостроковому горизонті (понад 10 років), обчислювальне ядро «R-SIM» автоматично застосувало неявні L-стабільні методи розв'язання.

Для комплексного аналізу поведінки системи було сформовано серію цільових симуляційних тестів. Аналіз базується на індикаторах, що були сформульовані у розділі 3.2.3: $A_{\min}$ (мінімальне значення готовності під час перехідних процесів); T_{loss} (час «навчального боргу», втрачений на простої для донавчання порівняно зі статичною системою); та A_{steady} (стаціонарна готовність після донавчання); $LoA_{\max}$ (максимальна втрата готовності у порівнянні з ІСД без донавчання); T_{std} (час, необхідний системі для подолання 95 % діапазону свого відновлення після початкового просідання). Додатково, розраховується показник відносного зменшення неготовності (Relative Unavailability Reduction, ΔU_{rel}). Цей індикатор показує, яку частку загального часу простою обладнання вдалося усунути завдяки впровадженню алгоритмів донавчання. Показник обчислюється за формулою:

$$U_{\text{rel}} = \frac{U_{\text{stat}} - U_{\text{learn}}}{U_{\text{stat}}} \times 100\% = \frac{(1 - A_{\text{stat}}) - (1 - A_{\text{learn}})}{1 - A_{\text{stat}}} \times 100\%, \quad (4.1)$$

де U_{stat} та U_{learn} – стаціонарна неготовність статичної системи (без навчання) та динамічної системи (з донавчанням) відповідно, що визначаються через розраховані коефіцієнти готовності як $U=1-A$.

Відправною точкою для порівняння є *базовий сценарій* (Сценарій 1), який спирається на медіанні значення параметрів з таблиці 4.5: $D_{\text{start}}=0.75$, $P_{\text{ld}}=0.2$, $P_{\text{lh}}=0.8$, $P_{\text{m}}=0.3$, $\mu_{\text{u}}=1/300$, $\mu_{\text{h}}=1/960$, $\mu_{\text{m}}=1/240$, $\lambda_{\text{d}}=10^{-5}$, $\lambda_{\text{m}}=8.333 \times 10^{-5}$, $\lambda_{\text{hc}}=1/2000$, $\lambda_{\text{l1}}=1/8$, $\lambda_{\text{l2}}=1/4$. Далі, шляхом почергової зміни декількох параметрів за умови незмінності інших, було сформовано серію цільових тестів. Результати експериментів для порівняльної тріади моделей зведено у таблицю 4.6.

Таблиця 4.6 – Результати симуляцій з почерговою зміною параметрів відносно базового сценарію (baseline)

№	Змінений параметр та опис сценарію	Asteady (analytical)	Tsd (год)	LoAmax	Amin	Tloss (год)	ΔU_{rel}
1	Базові налаштування	0.9764331	1401752	9.44×10^{-5}	0.972257	10012.5	15.47%
2	Швидке донавання: $\lambda_{l1} = 2$; $\lambda_{l2} = 4$;	0.9764331	1426783	0	0.972448	0	15.47%
3	Повільне донавання: $\lambda_{l1} = 0.1$; $\lambda_{l2} = 0.05$;	0.9764331	1416771	1.67×10^{-4}	0.972184	10012.5	15.47%
4	Швидке донавання l1 і повільне l2: $\lambda_{l1} = 2$; $\lambda_{l2} = 0.05$;	0.9764331	1456821	0	0.972419	0	15.47%
5	Швидке донавання l2 і повільне l1: $\lambda_{l1} = 0.1$; $\lambda_{l2} = 4$;	0.9764331	1391739	1.44×10^{-4}	0.972207	10012.5	15.47%
6	Середньо-висока D: Dstart = 0.85	0.9773001	170212	9.23×10^{-5}	0.974395	10012.5	11.77%
7	Висока D: Dstart = 0.95. Damx = 0.99	0.9773001	155194	1.41×10^{-4}	0.976381	15018.8	3.68%
8	Довге відновлення: $\mu_u = 1/500$, $\mu_h = 1/960$, $\mu_m = 1/480$	0.9559746	1421777	1.11×10^{-4}	0.952231	10012.5	8.01%
9	Швидке відновлення: $\mu_u = 1/100$, $\mu_h = 1/480$, $\mu_m = 1/120$	0.9881214	1381727	9.07×10^{-5}	0.984269	10012.5	25.06%
10	Повільне відновлення після критичної відмови та швидке відновлення інших типів: $\mu_u = 1/100$, $\mu_h = 1/960$, $\mu_m = 1/120$;	0.9879206	1386733	7.5×10^{-5}	0.983334	10012.5	28.30%

Аналіз результатів симуляційного моделювання. Таким чином, отримані результати для 10 цільових сценаріїв підтверджують базову гіпотезу дослідження, яка вже була розглянута у розділах 3.2 та 3.3 (рис 4.13). Додатково, аналіз чутливості (таблиця 4.6) дозволив виявити наступні три ключові експлуатаційні закономірності.



Рисунок 4.13 – Приклад результатів симуляції для тесту №3 з повільним донавчанням

1. Мінімізація «навчального боргу» залежить від швидкості оновлення алгоритмів після прихованих відмов (λ_{l1}). Прискорення виключно λ_{l1} (сценарій 4) зводить T_{loss} до нуля. Натомість прискорення лише λ_{l2} (після виявлених відмов) залишає простої на рівні 10 000 годин. Це також зумовлено припущенням, що приріст D з S11 є більшим, бо покриває більш чутливі приховані відмови.

2. Нелінійна залежність ефекту від початкової достовірності (D_{start}). Розгортання ЦД є економічно та технічно виправданим переважно для нових флотів SMR ($D_{start} = 0.75 - 0.85$), де відносний виграш ΔU_{rel} сягає 11.7–15.4%. Для зрілих систем із високою достовірністю ($D_{start}=0.95$) ефект від донавчання

падає до 3.68%, а час T_{loss} зростає понад 15 000 годин, що робить його недоцільним.

3. Обмеження СТОiP та ефект контрастного відновлення. При повільному ремонті (сценарій 8) ІСД здатна зменшити неготовність лише на 8.01%, при швидкому – на понад 25%. Водночас підтверджено ефект контрастності (сценарій 10): що тривалішим є аварійний ремонт μh порівняно зі штатним, то вищою є доцільність ІСД з донавчанням (максимум $\Delta U_{rel} = 28.30\%$), оскільки його головною задачею є превентивне уникнення найдорожчих критичних простоїв.

4.5 Аналіз результатів впровадження

Отримані результати дисертаційної роботи впроваджено наступним чином.

– У навчальному процесі кафедри кібербезпеки та інтелектуальних інформаційних технологій ХАІ (лекціях та практичних заняттях з навчальної дисципліни «Теорія і методи інтернет обчислень» у розрізі побудови інтелектуальних систем діагностування з донавчанням на основі розподілених цифрових двійників) для бакалаврів, магістрів і аспірантів, що навчаються за спеціальностями 123 – Комп’ютерна інженерія та 125 – Кібербезпека і захист інформації (акт впровадження від 10 березня 2026), що сприяло покращенню фундаментальності викладання матеріалу із сучасних парадигм індустріального інтернету речей (IIoT) із застосуванням перспективних технологій цифрових двійників та федеративного навчання.

– У науково-дослідницьких роботах кафедри (архітектуру ІСД, комплекс базових моделей та метод V&V), виконаних на замовлення МОН України, зокрема: «Методи, засоби та технології моделювання, розроблення, розгортання та забезпечення гарантоздатності мобільних інтелектуальних систем для об’єктів критичної інфраструктури» (№ Д/Р 0124U003250, 2024-теперішній час), «Наукові засади і методи забезпечення гарантоздатності флотів БПЛА інтелектуальних систем моніторингу потенційно небезпечних і військових

об'єктів» (№ Д/Р 0121U112172, 2021–2023 рр.), «Методи, моделі та інформаційні технології підвищення надійності та безпечності складних ІТ-систем на етапах розроблення та впровадження» (№ Д/Р 0121U113842, 2021–2023 рр.), «Методи, програмно-апаратні засоби та технології забезпечення гарантоздатності інтелектуальних систем індустриального інтернету речей» (№ Д/Р 0122U001065, 2022–2023 рр.) (акт впровадження від 9 березня 2026).

– На підприємстві ТОВ «НВП «РАДІКС» шляхом застосування методики використання цифрових двійників для підвищення ефективності процесів верифікації та валідації (V&V) інформаційно-керуючих систем і обґрунтування готовності обладнання у малих модульних реакторів з використанням засобів діагностування з донавчанням (акт впровадження від 26 лютого 2026). Впровадження даних методів дозволило підвищити достовірність оцінювання готовності систем безпеки малих модульних реакторів (ММР), зменшити кількість помилок на ранніх етапах проектування та зменшити ризики, пов'язані з дефіцитом експлуатаційних даних нових реакторів.

Таким чином, аналіз результатів впровадження розробленого програмно-математичного апарату підтверджує практичну значущість та інженерну доцільність застосування технології цифрових двійників з функцією донавчання для підвищення надійності та експлуатаційної готовності критичного промислового обладнання.

4.6 Висновки до четвертого розділу

На підставі викладеного матеріалу, аналізу розроблених технологій, програмних засобів та результатів їх апробації і впровадження, сформульовано такі висновки.

1. Розроблено еталонну ІТ-архітектуру розгортання інтелектуальної системи діагностування (ІСД) з донавчанням на базі цифрових двійників (ЦД). Запропонована дворівнева (Edge-Cloud) архітектура забезпечує проекцію створеної теоретико-множинної моделі на фізичні програмно-апаратні

компоненти індустріального інтернету речей (IIoT) із визначенням ключових протоколів та інтерфейсів взаємодії та забезпечує федеративне донавчання засобів ІСД для покращення достовірності діагностування.

2. Створено інструментальний програмний засіб «R-SIM» для моделювання готовності промислового IIoT обладнання, що керується отриманими науковими результатами. Засіб автоматизує процеси побудови простору станів мультифрагментних марковських моделей (МФММ) з HP2, ефективно вирішує проблему математичної «жорсткості» при розв'язанні систем диференціальних рівнянь (шляхом автоматичного вибору чисельного методу на основі індексу жорсткості), а також забезпечує перехресну верифікацію результатів симуляції з HP3.

3. Проведено практичну апробацію мультифрагментних моделей та ПЗ «R-SIM» на прикладі обладнання малих модульних реакторів з використанням моделі МФММ4. За результатами моделювання 10 цільових сценаріїв кількісно підтверджено базову гіпотезу дослідження: впровадження донавчання ЦД формує тимчасовий «навчальний борг» (просідання готовності на час налаштування), який у довгостроковій перспективі компенсується відносним зменшенням неготовності системи (до 28,3% за тестами). Доведено, що ефективність контуру донавчання асиметрично залежить від швидкості реакції саме на приховані відмови, має нелінійну зворотну залежність від початкової достовірності діагностування та лімітується фізичними можливостями служб технічного обслуговування.

4. Проаналізовано результати впровадження, що підтвердило високу практичну значущість та інженерну доцільність розроблених методів, моделей і технологій.

Література до четвертого розділу

1. Kharchenko V., Shcheglov V., Ivasiuk O., Morozova O. Digital Twin-Based Lifecycle Methodology for Ensuring Safety of NPP/SMR I&C Systems.

Technologies. 2026. Vol. 14, no. 1, article no. 46. P. 1–33. DOI: 10.3390/technologies14010046.

2. Chmielewska-Śmietanko D., Smoliński T., Bartela Ł., Chmielewski A. G. Prospects and Trends in the Development of Small Modular Nuclear Reactors. *Energies*. 2025. Vol. 18, no. 22, article no. 5970. DOI: 10.3390/en18225970.

3. Milivojević S., Riznić J. R. The empirical failure rate and repair rate of PWR primary coolant pumps. *Reliability Engineering & System Safety*. 1989. Vol. 24, no. 3. P. 267–273. DOI: 10.1016/0951-8320(89)90044-6.

4. Cadwallader L. C. *Review of maintenance and repair times for components in technological facilities*. Idaho National Laboratory, 2012. URL: <http://www.inl.gov> (дата звернення: 28.02.2026).

5. GE Vernova. *BWRX-300 general description*. 2025. URL: https://www.gevernova.com/content/dam/gevernova-nuclear/global/en_us/document (дата звернення: 28.02.2026).

6. NuScale Power. *The NuScale power module technical specifications*. 2026. URL: <https://www.nuscalepower.com/hubfs/NPM-technical-specifications.pdf?hsLang=en> (дата звернення: 28.02.2026).

7. Boles G. *Nuclear maintenance applications center: The physical basis of abnormal seal leakoff mitigation*. EPRI, 2012. URL: <https://www.epri.com/research/products/1025765> (дата звернення: 28.02.2026).

ВИСНОВКИ

У дисертаційній роботі наведено теоретичне узагальнення та нове вирішення актуальної науково-прикладної задачі, що полягає у розробленні моделей, методу та засобів оцінювання і забезпечення надійності промислового обладнання систем індустріального інтернету речей на основі інтелектуальних систем діагностування з використанням цифрових двійників та режимів донавчання. При цьому було отримано такі наукові та практичні результати.

1. Вперше запропоновано теоретико-множинну модель та архітектуру інтелектуальної системи діагностування промислового обладнання систем індустріального інтернету речей з використанням цифрового двійника, яка, на відміну від існуючих, забезпечує локальне та федеративне донавчання, коригування тестових послідовностей і параметрів оперативного контролю на базі виявлених прихованих відмов, а також можливість динамічного обчислення функції готовності, що надає змогу знизити ймовірність прихованих відмов та підвищити керованість процесами обслуговування..

2. Удосконалено базові та мультифрагментні моделі готовності відновлювального промислового обладнання систем індустріального інтернету речей, які, на відміну від існуючих, утворюють множину, сформовану на основі розробленого онтологічного класифікатора, та враховують зміну достовірності діагностування, зумовлену донавчанням цифрового двійника після прояву різних типів відмов, що дозволяє обирати структуру моделі залежно від особливостей об'єкта і процесів діагностування та підвищити достовірність оцінювання готовності системи.

3. Удосконалено метод верифікації та валідації мультифрагментних марковських моделей інтелектуальних систем діагностування промислового обладнання з використанням індустріального інтернету речей, який відрізняється використанням багатоступеневої перехресної перевірки результатів за допомогою двох незалежних методологій (на основі графів станів та структурно-автоматних моделей), а також врахуванням математичної «жорсткості» матриці

інтенсивностей для вибору чисельного методу розв'язання, що дозволяє уникнути обчислювальних похибок та підвищити достовірність оцінювання готовності.

4. Практичні результати полягають у доведенні теоретичних положень дисертаційної роботи до конкретної інформаційної технології (еталонна архітектура розгортання ІСД на основі ЦД для промислового ІоТ обладнання) та інструментальних засобів, зокрема розроблено програмний засіб «R-SIM». Цей засіб автоматизує процеси побудови простору станів мультифрагментних марковських моделей, вирішує проблему математичної «жорсткості» систем диференціальних рівнянь та дозволяє оцінити динаміку готовності у порівнянні із аналогічною системою без донавчання. Додатково достовірність отриманих результатів підтверджено апробацією розробленого програмного засобу «R-SIM» на розробленій МФММ4 для циркуляційного насоса SMR з використанням референсних параметрів з реального обладнання.

5. Результати дисертаційної роботи успішно впроваджено: у промисловий сектор на підприємстві ТОВ «НВП «РАДІКС» шляхом застосування розробленої методики для підвищення ефективності процесів верифікації та валідації інформаційно-керуючих систем і обґрунтування готовності систем діагностування малих модульних реакторів; у навчальний процес Національного аерокосмічного університету «ХАІ»; при виконанні державних науково-дослідних проєктів (№ Д/Р 0124U003250, № Д/Р 0121U112172, № Д/Р 0121U113842, № Д/Р 0122U001065) у Національному аерокосмічному університеті «ХАІ».

6. Основні результати дисертації опубліковано у 7 наукових працях, у тому числі: 2 статті у наукових фахових виданнях України (категорія «Б»); 1 стаття в англomовному журналі (категорія «А»), що індексується у наукометричній базі даних Scopus (квартиль Q1); 1 колективна монографія; 2 публікації у працях міжнародних конференцій, матеріали яких включено до бази даних Scopus і апробація у 1 національній науковій конференції.

7. Подальші дослідження доцільно зосередити на: розширенні математичного апарату мультифрагментних моделей для більш адекватного врахування впливу кібератак та федеративного донавчання; врахуванні динаміки

зміни достовірності діагностування і його донавчання за різними фасетами (типами) відмов; науково-практичній реалізації ІСД на основі ЦД відповідно до розробленої теоретико-множинної моделі та еталонної архітектури; дослідженні конкретних алгоритмів машинного навчання для виявлення прихованих відмов на обладнанні ІоТ, а також комплексному врахуванні використання зовнішніх засобів резервного тестування (наприклад, тепловізійного чи акустичного контролю) для формування мультимодальних датасетів донавчання цифрового двійника.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Грищук А. М. Необхідність зміни технологічного укладу на основі драйверів економічного зростання. *Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Економіка і управління*. 2020. Т. 31 (70), № 2 (1). С. 81–88. URL: [http://nbuv.gov.ua/UJRN/UZTNU_econ_2020_31\(70\)_2\(1\)_16](http://nbuv.gov.ua/UJRN/UZTNU_econ_2020_31(70)_2(1)_16) (дата звернення: 01.02.2026).
2. Bigliardi B., Bottani E., Casella G. Enabling technologies, application areas and impact of industry 4.0: a bibliographic analysis. *Procedia Manufacturing*. 2020. Vol. 42. P. 322–326. DOI: 10.1016/j.promfg.2020.02.086.
3. Islam M. T., Sepanloo K., Woo S., Woo S. H., Son Y.-J. A Review of the Industry 4.0 to 5.0 Transition: Exploring the Intersection, Challenges, and Opportunities of Technology and Human–Machine Collaboration. *Machines*. 2025. Vol. 13, no. 4, article no. 267. DOI: 10.3390/machines13040267.
4. Alabadi M., Habbal A., Wei X. Industrial Internet of Things: Requirements, Architecture, Challenges, and Future Research Directions. *IEEE Access*. 2022. Vol. 10. P. 66374–66400. DOI: 10.1109/ACCESS.2022.3185049.
5. Boyes H., Hallaq B., Cunningham J., Watson T. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry*. 2018. Vol. 101. P. 1–12. DOI: 10.1016/j.compind.2018.04.015.
6. Hermeto R. T., Gallais A., Theoleyre F. Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: A survey. *Computer Communications*. 2017. Vol. 114. P. 84–105. DOI: 10.1016/j.comcom.2017.10.004.
7. What is IIoT and which industries does it impact? *CCLAB*. URL: <https://www.cclab.com/news/what-is-iiot-and-which-industries-does-it-impact> (дата звернення: 01.02.2026).
8. Afrin S., Rafa S., Kabir M., Farah T., Alam M. S., Lameesa A., Ahmed S., Gandomi A. Industrial Internet of Things: Implementations, challenges, and potential

solutions across various industries. *Computers in Industry*. 2025. Vol. 170, article no. 104317. DOI: 10.1016/j.compind.2025.104317.

9. Jarwar M. A., Watson J., Ali S. Modeling Industrial IoT Security Using Ontologies: A Systematic Review. *IEEE Open Journal of the Communications Society*. 2025. DOI: 10.1109/OJCOMS.2025.3532224.

10. Endres H., Indulska M., Ghosh A., Baiyere A., Broser S. Industrial Internet of Things (IIoT) *Business Model Classification*. *40th International Conference on Information Systems (ICIS)* : Proc. Int. Conf., Munich, Germany, 2019. URL: https://www.researchgate.net/publication/342261722_Industrial_Internet_of_Things_IIoT_Business_Model_Classification (дата звернення: 01.02.2026).

11. Lin S.-W., Mellor S., Munz H., Barnstedt E. Architecture Alignment and Interoperability. An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper. URL: https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf (дата звернення: 01.02.2026).

12. ДСТУ ISO/IEC 30141:2019. Інтернет речей. Еталонна архітектура (ISO/IEC 30141:2018, IDT). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=83590 (дата звернення: 01.02.2026).

13. Bochra G., Abderrahim S. The Importance of Semantic Interoperability in The Internet of Things (IOT). *SWIoT+MSW@KGSWC* : Proc. Conf., 2023. URL: <https://api.semanticscholar.org/CorpusID:269043727> (дата звернення: 01.02.2026).

14. Moreira J. L., Daniele L. M., Ferreira Pires L., van Sinderen M. J., Wasielewska K., Szmeja P., Pawlowski W., Ganzha M., Paprzycki M. Towards IoT platforms' integration: Semantic Translations between W3C SSN and ETSI SAREF. *SEMANTiCS conference 2017* : Proc. Int. Conf., Amsterdam, Netherlands, 2017. URL: <http://ceur-ws.org/Vol-2063/sisiot-paper3.pdf> (дата звернення: 01.02.2026).

15. Cornejo Lupa M., Aguilera A., Dongo I. Developing an Ontology to Enhance Semantic Interoperability in IoT Environment. *CLEI Electronic Journal*. 2025. Vol. 28. DOI: 10.19153/cleiej.28.3.4.

16. Stryzhak O., Horborukov V., Prychodniuk V., Franchuk O., Chepkov R. Decision-making System Based on The Ontology of The Choice Problem. *Journal of Physics: Conference Series*. 2021. Vol. 1828, no. 1. P. 012007. DOI: 10.1088/1742-6596/1828/1/012007.
17. Mrabet H., Belguith S., Alhomoud A., Jemai A. A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. *Sensors*. 2020. Vol. 20, no. 13, article no. 3625. DOI: 10.3390/s20133625.
18. Gautam V. L., Lanjewar U. A. Cloud, Fog, and Edge Computing: A Comparative Analysis of Architectures, Applications, Security Challenges and Performance. *International Journal of Computer Science Trends and Technology (IJCST)*. 2025. P. 130–137. URL: <https://www.ijcstjournal.org/conference-issue/ICAMT2025/23.pdf> (дата звернення: 01.02.2026).
19. ДСТУ 2860-94. Надійність техніки. Терміни та визначення. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=25034 (дата звернення: 01.02.2026).
20. ДСТУ EN 61508-1:2019. Функційна безпечність електричних, електронних, програмованих електронних систем, пов'язаних із безпекою. Частина 1. Загальні вимоги (EN 61508-1:2010, IDT; IEC 61508-1:2010, IDT). URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=84383 (дата звернення: 01.02.2026).
21. Owen B., Smith J. Signal Integrity Challenges in Harsh Industrial Environments Under Cyber Interference. 2025. URL: https://www.researchgate.net/publication/398394492_Signal_Integrity_Challenges_in_Harsh_Industrial_Environments_Under_Cyber_Interference (дата звернення: 01.02.2026).
22. Securing IIoT Systems: An Analysis of Key Vulnerabilities and Threats in Manufacturing Industries. *CCLAB*. URL: <https://www.cclab.com/news/securing-iiot-systems-an-analysis-of-key-vulnerabilities-and-threats-in-manufacturing-industries> (дата звернення: 01.02.2026).

23. Kohn L. T., Corrigan J. M., Donaldson M. S. Why Do Errors Happen? *Institute of Medicine (US) Committee on Quality of Health Care in America*. 2000. URL: <https://www.ncbi.nlm.nih.gov/books/NBK225171/> (дата звернення: 01.02.2026).
24. Industrial IoT Enhancing Equipment Reliability and Uptime. 2025. URL: <https://oxmaint.com/blog/post/iiot-equipment-reliability-uptime> (дата звернення: 02.02.2026).
25. Benhanifia A., Cheikh Z. B., Oliveira P. M., Valente A., Lima J. Systematic review of predictive maintenance practices in the manufacturing sector. *Intelligent Systems with Applications*. 2025. Vol. 26, article no. 200501. DOI: 10.1016/j.iswa.2025.200501.
26. ДСТУ 2389-94. Технічне діагностування та контроль технічного стану. Терміни та визначення. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=104134 (дата звернення: 02.02.2026).
27. Gonzalez-Jimenez D., del-Olmo J., Poza J., Garramiola F., Madina P. Data-Driven Fault Diagnosis for Electric Drives: A Review. *Sensors*. 2021. Vol. 21, no. 12, article no. 4024. DOI: 10.3390/s21124024.
28. Yan W., Wang J., Lu S., Zhou M., Peng X. A Review of Real-Time Fault Diagnosis Methods for Industrial Smart Manufacturing. *Processes*. 2023. Vol. 11, no. 2, article no. 369. DOI: 10.3390/pr11020369.
29. Leite D., Andrade E., Rativa D., Maciel A. M. A. Fault Detection and Diagnosis in Industry 4.0: A Review on Challenges and Opportunities. *Sensors*. 2025. Vol. 25, no. 1, article no. 60. DOI: 10.3390/s25010060.
30. Rodriguez-Aguilar R., Marmolejo-Saucedo J.-A., Köse U. Development of a Digital Twin Driven by a Deep Learning Model for Fault Diagnosis of Electro-Hydrostatic Actuators. *Mathematics*. 2024. Vol. 12, no. 19, article no. 3124. DOI: 10.3390/math12193124.
31. Ademujimi T., Prabhu V. Model-Driven Bayesian Network Learning for Factory-Level Fault Diagnostics and Resilience. *Sustainability*. 2024. Vol. 16, no. 2, article no. 513. DOI: 10.3390/su16020513.

32. Kumar S., Raj K. K., Cirrincione M., Cirrincione G., Franzitta V., Kumar R. R. A Comprehensive Review of Remaining Useful Life Estimation Approaches for Rotating Machinery. *Energies*. 2024. Vol. 17, no. 22, article no. 5538. DOI: 10.3390/en17225538.
33. Carino J. A. та ін. Fault Detection and Identification Methodology Under an Incremental Learning Framework Applied to Industrial Machinery. *IEEE Access*. 2018. Vol. 6. P. 49755–49766. DOI: 10.1109/ACCESS.2018.2868430.
34. Chen B., Shen C., Shi J., Kong L., Tan L., Wang D., Zhu Z. Continual learning fault diagnosis: A dual-branch adaptive aggregation residual network for fault diagnosis with machine increments. *Chinese Journal of Aeronautics*. 2023. Vol. 36, no. 6. P. 361–377. DOI: 10.1016/j.cja.2022.08.019.
35. Chen X., Gao C., Delbruck T., Liu S.-C. EILE: Efficient Incremental Learning on the Edge. *Artificial Intelligence Circuits and Systems (AICAS)* : Proc. 3rd IEEE Int. Conf., Washington DC, USA, 2021. P. 1–4. DOI: 10.1109/AICAS51828.2021.9458554.
36. Neupane D., Bouadjenek M. R., Dazeley R., Aryal S. Data-driven machinery fault diagnosis: A comprehensive review. *Neurocomputing*. 2025. Vol. 627, article no. 129588. DOI: 10.1016/j.neucom.2025.129588.
37. Savaglio C., Barbuto V., Mangione F., Fortino G. Generative Digital Twins: A Novel Approach in the IoT Edge-Cloud Continuum. *IEEE Internet of Things Magazine*. 2025. Vol. 8, no. 1. P. 42–48. DOI: 10.1109/IOTM.001.2400035.
38. Щеглов В. Р., Морозова О. І. Методи та технології розроблення цифрових двійників для гарантоздатних систем індустріального інтернету речей. *Системи управління, навігації та зв'язку*. 2022. Вип. 4 (70). С. 127–137. DOI: 10.26906/SUNZ.2022.4.127.
39. Щеглов В. Р., Морозова О. І. Застосування технології цифрових двійників відносно безпілотних літальних апаратів. *Авіація, промисловість, суспільство* : матеріали наук.-практ. конф., м. Харків, 2024. Харків, 2024. С. 47–51.

40. Гаєвський В. В., Кунуп Т. В., Морозова О. І., Щеглов В. Р. Розділ 11. Інтелектуальні системи інтернету речей з використанням технологій цифрових двійників. *Methods and technologies of ensuring quality and safety of intelligent systems* : кол. монографія / за заг. ред. В. С. Харченка, О. І. Морозової. Харків : Міністерство освіти і науки України, Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ», 2023. С. 193–224.

41. Bisanti G., Mainetti L., Montanaro T., Patrono L., Sergi I. Digital twins for aircraft maintenance and operation: A systematic literature review and an IoT-enabled modular architecture. *Internet of Things*. 2023. Vol. 24, article no. 100991. DOI: 10.1016/j.iot.2023.100991.

42. ISO 23247-1:2021(en). *Automation systems and integration – Digital twin framework for manufacturing – Part 1: Overview and general principles*. URL: <https://www.iso.org/obp/ui/en/#iso:std:iso:23247:-1:ed-1:v1:en> (дата звернення: 02.02.2026).

43. The DTC Glossary. *Digital Twin Consortium*. URL: <https://www.digitaltwinconsortium.org/glossary/glossary/#digital-twin> (дата звернення: 02.02.2026).

44. Kharchenko V., Shcheglov V., Ivasiuk O., Morozova O. Digital Twin-Based Lifecycle Methodology for Ensuring Safety of NPP/SMR I&C Systems. *Technologies*. 2026. Vol. 14, no. 1, article no. 46. P. 1–33. DOI: 10.3390/technologies14010046.

45. Grieves M. Digital Model, Digital Shadow, Digital Twin. 2023. URL: https://www.researchgate.net/publication/369830792_Digital_Model_Digital_Shadow_Digital_Twin (дата звернення: 02.02.2026).

46. Grieves M. Origins of the Digital Twin Concept. 2016. DOI: 10.13140/RG.2.2.26367.61609.

47. Singh M., Fuenmayor E., Hinchy E. P., Qiao Y., Murray N., Devine D. Digital Twin: Origin to Future. *Applied System Innovation*. 2021. Vol. 4, no. 2, article no. 36. DOI: 10.3390/asi4020036.

48. Schuh G., Anderl R., Dumitrescu R., Krüger A., ten Hompel M. Industrie 4.0 Maturity Index. Managing the Digital Transformation of Companies – UPDATE 2020. 2020. URL: <https://en.acatech.de/publication/industrie-4-0-maturity-index-update-2020/> (дата звернення: 02.02.2026).
49. Leon-Medina J. X., Tibaduiza D. A., Parés N., Pozo F. Digital twin technology in wind turbine components: A review. *Intelligent Systems with Applications*. 2025. Vol. 26, article no. 200535. DOI: 10.1016/j.iswa.2025.200535.
50. Coelho M., Houck K., Browning L., Sabharwall P., Folmar C., Cavaluzzi J., McClure P., Dunker J. Early Fault Detection in Nuclear Systems: A Digital Engineering Approach. *IEEE Open Journal of Systems Engineering*. 2025. P. 1–12. DOI: 10.1109/OJSE.2025.3562518.
51. Sadeghi A., Bellavista P., Song W., Yazdani-Asrami M. Digital Twins for Condition and Fleet Monitoring of Aircraft: Toward More-Intelligent Electrified Aviation Systems. *IEEE Access*. 2024. Vol. 12. P. 99806–99832. DOI: 10.1109/ACCESS.2024.3371902.
52. Wang Q., Zhang X., Lu J., Xiao G., Ren Y., Li W. Digital Twin-Driven Physically Constrained Generative Adversarial Network for Industrial Boiler Fault Diagnosis. *IEEE Transactions on Instrumentation and Measurement*. 2025. Vol. 74, article no. 3530215. DOI: 10.1109/TIM.2025.3558806.
53. Shcheglov V., Kharchenko V., Volochiy B., Morozova O., Zaitseva E. Availability Assessing Intelligent Diagnostic System with Relearning after Complex Equipment Failures: Structural-Automata Model-based Methodology of Markov's Analysis. *Dependable Systems, Services and Technologies (DESSERT'2024)* : Proc. 14th IEEE Int. Conf., Athens, Greece, 2024. P. 1–8. DOI: 10.1109/DESSERT65323.2024.11122169.
54. Kabir S. An overview of fault tree analysis and its application in model based dependability analysis. *Expert Systems with Applications*. 2017. Vol. 77. P. 114–135. DOI: 10.1016/j.eswa.2017.01.058.

55. Keefe K., Sanders W. Reliability Analysis with Dynamic Reliability Block Diagrams in the Möbius Modeling Tool. *EAI Endorsed Transactions on Security and Safety*. 2016. Vol. 3. DOI: 10.4108/eai.14-12-2015.2262723.
56. Kharchenko V., Odarushchenko O., Butenko V., Popov P. Selecting mathematical software for dependability assessment of computer systems described by stiff Markov chains. *CEUR Workshop Proceedings*. 2013. Vol. 1000. P. 146–162. URL: https://www.researchgate.net/publication/289567556_Selecting_mathematical_software_for_dependability_assessment_of_computer_systems_described_by_stiff_Markov_chains (дата звернення: 02.02.2026).
57. Malhotra M., Muppala J. K., Trivedi K. S. Stiffness-tolerant methods for transient analysis of stiff Markov chains. *Microelectronics Reliability*. 1994. Vol. 34, no. 11. P. 1825–1841. DOI: 10.1016/0026-2714(94)90137-6.
58. Fonseca i Casas P., Ruiz-Martin C. Building Models in Pairs for Cross-Verification Using SDL and DEVS. *Advanced Theory and Simulations*. 2024. Vol. 7. DOI: 10.1002/adts.202300839.
59. Щеглов В. Р. Інформаційна технологія оцінювання готовності інтелектуальних систем діагностування з донавчанням на основі цифрових двійників. *Measuring and computing devices in technological processes*. 2026. № 1. С. 87–96. DOI: 10.31891/2219-9365-2026-85-11.
60. Cohen B. Set Theory as a Semantic Framework for Object Oriented Modeling. *Object-Oriented Technologies*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1998. P. 161–168. DOI: 10.1007/3-540-69687-3_34.
61. Tazaki E., Amagasa M. Structural modeling in a class of systems using fuzzy sets theory. *IEEE Conference on Decision and Control including the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications* : Proc. Conf., New Orleans, LA, USA, 1977. P. 1408–1413. DOI: 10.1109/CDC.1977.271524.
62. Shcheglov V., Kharchenko V., Borges J., Morozova O. Availability Markov Models of Complex Equipment with Relearning Diagnostic Means after Detecting Latent Failures. *Dependable Systems, Services and Technologies*

(DESSERT'2023) : Proc. 13th IEEE Int. Conf., Athens, Greece, 2023. P. 1–7. DOI: 10.1109/DESSERT61349.2023.10416539.

63. Горборуков В. В., Франчук О. В., Приходнюк В. В., Кирієнко О. В. Трансдисциплінарна інформаційно-аналітична платформа підтримки процесів оцінювання. *Наукові записки НаУКМА*. Комп'ютерні науки. 2022. Т. 5. С. 79–84. DOI: 10.18523/2617-3808.2022.5.79-84.

64. Горборуков В. В., Приходнюк В. В., Франчук О. В. Алгоритм конкурентної нормалізації в системі рейтингового оцінювання інтелектуальних досягнень. *Наукові записки Малої академії наук України*. 2022. № 1 (23). С. 3–12. DOI: 10.51707/2618-0529-2022-23-01.

65. Інформаційно-аналітична система підтримки прийняття рішень «ПОЛІЕДР-Альтернатива». URL: <https://e-devel.ulif.org.ua/sppr-int> (дата звернення: 08.04.2026).

66. Volochiy B., Mandziy B., Ozirkovsky L. Extending the features of software for reliability analysis of fault-tolerant systems. *Computational Problems of Electrical Engineering*. 2012. Vol. 2, no. 2. P. 113–121.

67. Hairer E., Wanner G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Vol. 14. Springer Science & Business Media, 1996. URL: <https://link.springer.com/book/10.1007/978-3-642-05221-7> chains (дата звернення: 03.03.2026).

68. Bobbio A., Trivedi K. S. An aggregation technique for the transient analysis of stiff Markov chains. *IEEE Transactions on Computers*. 1986. Vol. C-35, no. 9. P. 803–814. URL: <https://ieeexplore.ieee.org/document/1676840> (дата звернення: 03.03.2026).

69. Volochiy S., Fedasyuk D., Chohey R. Formalized development of the state transition graphs using the Erlang phase method. *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications* : Proc. 9th IEEE Int. Conf., Bucharest, Romania, Sept. 21–23, 2017. P. 1098–1101. DOI: 10.1109/IDAACS.2017.8095255.

70. Strielkina A., Volochiy S., Kharchenko V. Discrete-Continuous Stochastic Model of Insulin Pump Functioning for Health IoT System Using Erlang Phase Method. *ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer* : Proc. 15th Int. Conf., Volume II: Workshops, Kherson, Ukraine, Jun. 12–15, 2019. P. 793–808. URL: http://ceur-ws.org/Vol-2393/paper_402.pdf (дата звернення: 03.03.2026).
71. Chmielewska-Śmietanko D., Smoliński T., Bartela Ł., Chmielewski A. G. Prospects and Trends in the Development of Small Modular Nuclear Reactors. *Energies*. 2025. Vol. 18, no. 22, article no. 5970. DOI: 10.3390/en18225970.
72. Milivojević S., Riznić J. R. The empirical failure rate and repair rate of PWR primary coolant pumps. *Reliability Engineering & System Safety*. 1989. Vol. 24, no. 3. P. 267–273. DOI: 10.1016/0951-8320(89)90044-6.
73. Cadwallader L. C. *Review of maintenance and repair times for components in technological facilities*. Idaho National Laboratory, 2012. URL: <http://www.inl.gov> (дата звернення: 28.02.2026).
74. GE Vernova. *BWRX-300 general description*. 2025. URL: https://www.gevernova.com/content/dam/gevernova-nuclear/global/en_us/document (дата звернення: 28.02.2026).
75. NuScale Power. *The NuScale power module technical specifications*. 2026. URL: <https://www.nuscalepower.com/hubfs/NPM-technical-specifications.pdf?hsLang=en> (дата звернення: 28.02.2026).
76. Boles G. *Nuclear maintenance applications center: The physical basis of abnormal seal leakoff mitigation*. EPRI, 2012. URL: <https://www.epri.com/research/products/1025765> (дата звернення: 28.02.2026).

ДОДАТОК А. АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ

Затверджую

Проректор з науково-педагогічної роботи
Національного аерокосмічного університету
«Харківський авіаційний інститут»



Андрій ГУМЕННИЙ

2026 року

АКТ ВПРОВАДЖЕННЯ

наукових результатів дисертаційної роботи

Щеглова Владислава Романовича, виконаної на здобуття наукового ступеня
доктора філософії, у навчальному процесі кафедри кібербезпеки та
інтелектуальних інформаційних технологій

Комісія у складі: голови комісії – декана факультету радіоелектроніки, комп'ютерних систем та інфокомунікацій к.т.н. Олексія ОДОКІСНКА, і членів – професора кафедри кібербезпеки та інтелектуальних інформаційних технологій, к.т.н. Клайда ФУРМАНОВА, професора кафедри кібербезпеки та інтелектуальних інформаційних технологій, к.т.н. Дмитра УЗУНА, доцента кафедри кібербезпеки та інтелектуальних інформаційних технологій, к.т.н. В'ячеслава ДУЖОГО, встановила, що наукові результати, отримані під час дослідження забезпечення та оцінки надійності промислового обладнання з використанням індустріального інтернету речей і цифрових двійників, а саме:

- теоретико-множинна модель, архітектура та метод функціонування інтелектуальної системи діагностування промислового обладнання систем індустріального інтернету речей на основі цифрового двійника,
- комплекс базових та мультифрагментних моделей готовності відновлювального промислового обладнання систем індустріального інтернету речей,
- метод верифікації та валідації мультифрагментних марковських моделей інтелектуальних систем діагностування промислового обладнання з

використанням індустріального інтернет речей, реалізовані у навчальному процесі кафедри кібербезпеки та інтелектуальних інформаційних технологій у вигляді:

- лекційного матеріалу і практичних занять з використання інструментальних засобів та методів при оцінюванні та забезпеченні надійності промислового обладнання, зокрема, систем індустріального інтернету речей, побудови інтелектуальних систем діагностування на основі цифрових двійників із режимами донавчання відповідно до вимог у навчальній дисципліні «Теорія і методи інтернет обчислень» (4 години).

Це дозволило покращити фундаментальність викладання матеріалу із сучасних парадигм Інтернет-обчислень, а саме, технологій індустріального інтернету речей, крайових і хмарних обчислень та розроблення цифрових двійників, наочність та практичну спрямованість навчального процесу, якість підготовки фахівців за напрямками комп'ютерних інженерії, кібербезпеки та захисту інформації.

Голова комісії

Члени комісії



Олексій ОДОКІЄНКО

Клайд ФУРМАНОВ

Дмитро УЗУН

В'ячеслав ДУЖИЙ

Затверджую

Проректор з наукової роботи

Національного аерокосмічного університету

«Харківський авіаційний інститут»

д-р наук з держ. упр., професор

Світлана ДОМБРОВСЬКА

2026 року



АКТ ВПРОВАДЖЕННЯ

наукових результатів дисертаційної роботи Щеглова Владислава Романовича,
виконаної на здобуття наукового ступеня доктора філософії,
у науково-дослідних проєктах Національного аерокосмічного університету
«Харківський авіаційний інститут»

Комісія у складі: голови – декана факультету радіоелектроніки, комп'ютерних систем та інфокомунікацій к.т.н. Олексія ОДОКІЄНКА і членів – професора кафедри кібербезпеки та інтелектуальних інформаційних технологій, д.т.н. Германа ФЕСЕНКА, професора кафедри кібербезпеки та інтелектуальних інформаційних технологій, к.т.н. Олександра ОРЕХОВА, доцента кафедри кібербезпеки та інтелектуальних інформаційних технологій, к.т.н. Артема ТЕЦЬКОГО, встановила, що наукові результати, а саме:

- теоретико-множинну модель, архітектуру та метод функціонування інтелектуальної системи діагностування промислового обладнання систем індустриального інтернету речей на основі цифрового двійника, які, на відміну від існуючих, забезпечують локальне та федеративне донавчання, коригування тестових послідовностей і параметрів оперативного контролю на базі виявлених прихованих відмов, а також динамічний перерахунок функції готовності, що надає змогу знизити ймовірність прихованих відмов та підвищити керованість процесами обслуговування;

- комплекс базових та мультифрагментних моделей готовності відновлювального промислового обладнання систем індустриального інтернету речей, який, на відміну від існуючих, формується на основі розробленого класифікатора та враховує динаміку зміни достовірності діагностування, зумовлену локальним або федеративним донавчанням цифрового двійника після

прояву різних типів відмов, що дозволяє обирати структуру моделі залежно від особливостей об'єкта і процесів діагностування та підвищити достовірність оцінювання готовності;

– удосконалений метод верифікації та валідації мультифрагментних марковських моделей інтелектуальних систем діагностування промислового обладнання з використанням індустріального інтернет речей, який відрізняється використанням багатоступеневої перехресної перевірки результатів за допомогою двох незалежних методологій (на основі графів станів та структурно-автоматних моделей), а також врахуванням математичної "жорсткості" (stiffness) матриці інтенсивностей для вибору чисельного методу розв'язання, що дозволяє уникнути обчислювальних похибок та підвищити достовірність оцінювання готовності, реалізовані у вигляді наукових положень і розробок, використаних при виконанні науково-дослідних проєктів за замовленням Міністерства освіти і науки України:

– «Методи, засоби та технології моделювання, розроблення, розгортання та забезпечення гарантоздатності мобільних інтелектуальних систем для об'єктів критичної інфраструктури» (№ Д/Р 0124U003250, 2024–теперішній час);

– «Наукові засади і методи забезпечення гарантоздатності флотів БПЛА інтелектуальних систем моніторингу потенційно небезпечних і військових об'єктів» (№ Д/Р 0121U112172, 2021–2023 рр.);

– «Методи, моделі та інформаційні технології підвищення надійності та безпечності складних ІТ-систем на етапах розроблення та впровадження» (№ Д/Р 0121U113842, 2021–2023 рр.);

– «Методи, програмно-апаратні засоби та технології забезпечення гарантоздатності інтелектуальних систем індустріального інтернету речей» (№ Д/Р 0122U001065, 2022–2023 рр.);

Це дозволило підвищити показники надійності та точності оцінювання готовності для інтелектуальних систем та критичних інфраструктур, які досліджувалися в рамках виконання НДР впродовж 2022-2025 рр.

Голова комісії

Члени комісії



Олексій ОДОКІЄНКО

Герман ФЕСЕНКО

Олександр ОРСХОВ

Артем ТЕЦЬКИЙ

ЗАТВЕРДЖУЮ
 Генеральний директор
 ТОВ "НВП "РАДІКС"
 Катерина БАХМАЧ
 2026 року



АКТ ВПРОВАДЖЕННЯ

наукових результатів дисертаційної роботи Щеглова Владислава Романовича за темою "Моделі, метод та засоби інтелектуальної системи діагностування промислового обладнання з використанням індустриального інтернету речей і цифрових двійників", виконаної на здобуття наукового ступеня доктора філософії, у науково-дослідних проєктах ТОВ "НВП "РАДІКС"

Комісія у складі: голови комісії – операційного директора ТОВ "НВП "РАДІКС", PhD Олександра Івасюка, і членів - заступника операційного директора ТОВ "НВП "РАДІКС", PhD Наталії Рвачової, провідного наукового співробітника ТОВ "НВП "РАДІКС", PhD Вікторії Кулініч встановила, що у ТОВ "НВП "РАДІКС" використано наукові результати дисертаційної роботи за темою "Моделі, метод та засоби інтелектуальної системи діагностування промислового обладнання з використанням індустриального інтернету речей і цифрових двійників", а саме:

Методику використання цифрового двійника для підтримки життєвого циклу розробки критичних інформаційно-керуючих систем (ІКС), включно з етапами проєктування, верифікації, кваліфікації та валідації;

теоретико-множинну модель цифрового двійника ІКС, яка формалізує взаємодію фізичних компонентів, віртуальних моделей, потоків даних та стейкхолдерів, дозволяючи зменшити витрати на розробку нових та модернізацію існуючих ІКС;

формалізовану архітектуру цифрового двійника ІКС, що забезпечує узгоджене використання моделей системи, середовища та сценаріїв відмов для підвищення функційної безпечності;

метод оцінювання показника готовності інтелектуальних діагностичних систем із використанням багатофрагментних марковських моделей, що враховують адаптивні властивості цифрових двійників, механізми їх довчання (relearning) та зниження ризиків прихованих відмов;

Застосування зазначених результатів забезпечує:

- методичну основу для використання цифрових двійників як інструменту інженерної підтримки життєвого циклу ІКС;
- підвищення ефективності процесів верифікації та валідації (V&V) за рахунок автоматизації тестування на базі цифрових моделей;
- скорочення часу аналізу та обґрунтування змін алгоритмів керування без порушення вимог функційної безпечності;
- обґрунтування надійнісних характеристик перспективних систем діагностики для малих модульних реакторів (ММР);
- скорочення часу реалізації проєктів щодо модернізації існуючих рішень завдяки оптимізації етапів тестування;

Впровадження даних методів дозволило підвищити достовірність оцінювання готовності систем безпеки ММР, зменшити кількість помилок на ранніх етапах проєктування та зменшити ризики, пов'язані з дефіцитом експлуатаційних даних нових реакторів.

Голова комісії

Члени комісії



Олександр ІВАСЮК

Наталія РВАЧОВА

Вікторія КУЛІНІЧ

ДОДАТОК Б. ЛІСТИНГИ КОДІВ ПРОГРАМНИХ ЗАСОБІВ

У даному додатку наведено вихідний код основного математичного модуля `rsim_core.py`, який містить реалізацію генераторів простору станів для мультифрагментних марковських моделей (БМ1–БМ5), алгоритмів розрахунку стаціонарних станів, а також розв'язання систем диференціальних рівнянь. Код компонентів графічного інтерфейсу (UI) не наводиться. Повний код доступний на [github](https://github.com/Veglad/phd_availability_calculator_rsim) за посиланням: https://github.com/Veglad/phd_availability_calculator_rsim.

```
rsim_core.py - R-SIM Mathematical Core
=====
PhD Decision-Support Tool

Contains all CTMC model builders, solvers, analytical steady-state,
validation, V&V auditor, and availability metrics.
Zero dependency on Streamlit - importable standalone for testing.

Models implemented
-----
BM1 : linear 1D-chain MFMM (2 / 3 / (N-1)·4+2 states)
BM2 : extended diagnostic chain (3 / 5 / (N-1)·6+5 states)
BM3 : 3D N×N×N grid CTMC (3 / 5 / 8N³-3N² states)
BM4 : 2D N×N grid CTMC (3 / 5 / 7N²-2N-2 states)
BM5 : 3D N×N×N grid + diagonal federal (3 / 5 / 8N³-3N²+(N-1)³ states)

Each model has three variants: Ideal · NoLearning · Learning.
"""

import re as _re
import warnings

import numpy as np
from scipy.integrate import solve_ivp
from scipy.linalg import null_space

warnings.filterwarnings("ignore")

# =====
# Scientific notation parser
# =====

def parse_sci(text: str) -> float:
    """
    Parse rate parameter strings. Supported formats:
    1e-6 | 1E-6 | 10^-6 | 3.5e-4 | 1.2*10^3
    1/480 | 1/12000 | 2/3 (fraction a/b)
    0.001 | plain floats
    Returns float or raises ValueError.
    """
    t = text.strip().replace(" ", "")
    frac = _re.match(
        r'^([+-]?[\d.]+(?:[eE][+-]?\d+)?)/([+-]?[\d.]+(?:[eE][+-]?\d+)?)$', t)
    if frac:
        num, den = float(frac.group(1)), float(frac.group(2))
        if den == 0:
            raise ValueError("Division by zero")
```

```

        return num / den
    t = t.replace("x", "*").replace("X", "*").replace("\u00d7", "*")
    t = _re.sub(r'10\^[+-]?\d+', lambda m: f"1e{m.group(1)}", t)
    t = _re.sub(r'\^[+-]?\d+', lambda m: f"e{m.group(1)}", t)
    t = t.replace("*1e", "e").replace("*1E", "e")
    return float(t)

# =====
# BM1 - IoT Component (linear 1D-chain)
# =====

def build_bm1_ideal(p: dict):
    """
    2 states: Su=0, Sd=1
    Su -> Sd : lambda_d
    Sd -> Su : mu_u
    """
    Q = np.zeros((2, 2))
    Q[0, 1] = p["lambda_d"]
    Q[1, 0] = p["mu_u"]
    for i in range(2):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sd"]

def build_bm1_nolearn(p: dict):
    """
    3 states: Su=0, Sd=1, Sh=2
    Su -> Sd : D * lambda_d
    Su -> Sh : (1-D) * lambda_d
    Sh -> Sd : lambda_d (diagnostic sweep rate, not D*lambda_d)
    Sd -> Su : mu_u
    """
    lam = p["lambda_d"]
    mu = p["mu_u"]
    D = p["D_start"]
    Q = np.zeros((3, 3))
    Q[0, 1] = D * lam
    Q[0, 2] = (1.0 - D) * lam
    Q[2, 1] = lam
    Q[1, 0] = mu
    for i in range(3):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sd", "Sh"]

def build_bm1_learning(p: dict, N: int):
    """
    (N-1) * 4 + 2 states. Linear 1D-chain MFMM.

    Fragment  $i \in 1..N-1$  : Sui, Sdi, Shi, Sli (4 states)
    Fragment N (reduced) : SuN, SdN (2 states, DN=1)

    Di = Dstart + (i-1) * ΔD, ΔD = (1 - Dstart) / (N-1)

    Within fragment  $i < N$ :
    Sui -> Sdi : Di * lambda_d
    Sui -> Shi : (1-Di) * lambda_d
    Shi -> Sdi : Di * lambda_d
    Sdi -> Sui : (1-L) * mu_u
    Sdi -> Sli : L * mu_u
    Inter-fragment:
    Sli -> Su{i+1} : lambda_l
    Last fragment N:
    SuN -> SdN : lambda_d

```

```

    Sd_N -> Su_N : mu_u
    """
    lam_d = p["lambda_d"]
    mu_u = p["mu_u"]
    lam_l = p["lambda_l"]
    L = p["L"]
    D_start = p["D_start"]
    delta_D = (1.0 - D_start) / (N - 1) if N > 1 else 0.0

    states = []
    offsets = {}
    for i in range(1, N):
        offsets[i] = len(states)
        states += [f"Su_{i}", f"Sd_{i}", f"Sh_{i}", f"Sl_{i}"]
    offsets[N] = len(states)
    states += [f"Su_{N}", f"Sd_{N}"]

    S = len(states)
    Q = np.zeros((S, S))

    for i in range(1, N):
        Di = D_start + (i - 1) * delta_D
        b = offsets[i]
        Su, Sd, Sh, Sl = b, b+1, b+2, b+3
        Q[Su, Sd] = Di * lam_d
        Q[Su, Sh] = (1.0 - Di) * lam_d
        Q[Sh, Sd] = Di * lam_d
        Q[Sd, Su] = (1.0 - L) * mu_u
        Q[Sd, Sl] = L * mu_u
        Q[Sl, offsets[i + 1]] = lam_l

    bN = offsets[N]
    Q[bN, bN+1] = lam_d
    Q[bN+1, bN] = mu_u

    for j in range(S):
        Q[j, j] = -Q[j, :].sum()

    su_indices = [offsets[i] for i in range(1, N + 1)]
    return Q, states, su_indices

# =====
# BM2 - Extended Diagnostic Chain (linear 1D-chain)
# =====

def build_bm2_ideal(p: dict):
    """
    3 states: Su=0, Sed=1, Sd=2 (Pc=1, D=1)
    Su -> Sed : lambda_d
    Sed -> Sd : lambda_ed
    Sd -> Su : mu_u
    """
    Q = np.zeros((3, 3))
    Q[0, 1] = p["lambda_d"]
    Q[1, 2] = p["lambda_ed"]
    Q[2, 0] = p["mu_u"]
    for i in range(3):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sed", "Sd"]

def build_bm2_nolearn(p: dict):
    """
    5 states: Su=0, Sh=1, Sed=2, Sad=3, Sd=4
    Su -> Sh : (1-Pc) * lambda_d

```

```

    Su -> Sed : Pc · lambda_d
    Sh -> Su  : mu_m           (direct return, no learning)
    Sed -> Sad : (1-D) · lambda_ed
    Sed -> Sd  : D · lambda_ed
    Sad -> Su  : mu_am
    Sd  -> Su  : mu_u
    """
    lam_d = p["lambda_d"]; lam_ed = p["lambda_ed"]
    mu_u  = p["mu_u"];      mu_m  = p["mu_m"]; mu_am = p["mu_am"]
    Pc    = p["Pc"];        D     = p["D_start"]
    Q = np.zeros((5, 5))
    Q[0, 1] = (1.0 - Pc) * lam_d
    Q[0, 2] = Pc * lam_d
    Q[1, 0] = mu_m
    Q[2, 3] = (1.0 - D) * lam_ed
    Q[2, 4] = D * lam_ed
    Q[3, 0] = mu_am
    Q[4, 0] = mu_u
    for i in range(5):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sh", "Sed", "Sad", "Sd"]

def build_bm2_learning(p: dict, N: int):
    """
    (N-1) · 6 + 5 states. Linear 1D-chain MFMM with extended diagnostics.

    Fragment i < N : Su, Sh, Sed, Sad, Sd, Sl (6 states)
    Fragment N      : Su, Sh, Sed, Sad, Sd    (5 states, no Sl)

    Within fragment i < N:
        Su -> Sh : (1-Pc) · lambda_d
        Su -> Sed : Pc · lambda_d
        Sh -> Sl : mu_m           (always triggers learning)
        Sed -> Sad : (1-D) · lambda_ed
        Sed -> Sd  : D · lambda_ed
        Sad -> Sl : mu_am           (always triggers learning)
        Sd -> Sl : mu_u           (always triggers learning)
        Sl -> Su_{i+1} : Pl · lambda_l
        Sl -> Su_i : (1-Pl) · lambda_l
    Last fragment N: Sh/Sad/Sd -> Su directly (no Sl).
    """
    lam_d = p["lambda_d"]; lam_ed = p["lambda_ed"]
    lam_l = p["lambda_l"]; mu_u  = p["mu_u"]
    mu_m  = p["mu_m"];      mu_am = p["mu_am"]
    Pc    = p["Pc"];        D     = p["D_start"]; Pl = p["Pl"]

    NORM = ["Su", "Sh", "Sed", "Sad", "Sd", "Sl"]
    LAST = ["Su", "Sh", "Sed", "Sad", "Sd"]

    offsets = {}
    states = []
    for i in range(1, N):
        offsets[i] = len(states)
        states += [f"{s}_{i}" for s in NORM]
    offsets[N] = len(states)
    states += [f"{s}_{N}" for s in LAST]

    S = len(states)
    Q = np.zeros((S, S))

    def idx(frag, name):
        lst = NORM if frag < N else LAST
        return offsets[frag] + lst.index(name)

    for i in range(1, N):

```

```

    Su = idx(i, "Su"); Sh = idx(i, "Sh")
    Sed = idx(i, "Sed"); Sad = idx(i, "Sad")
    Sd = idx(i, "Sd"); Sl = idx(i, "Sl")
    Q[Su, Sh] = (1.0 - Pc) * lam_d
    Q[Su, Sed] = Pc * lam_d
    Q[Sh, Sl] = mu_m
    Q[Sed, Sad] = (1.0 - D) * lam_ed
    Q[Sed, Sd] = D * lam_ed
    Q[Sad, Sl] = mu_am
    Q[Sd, Sl] = mu_u
    Q[Sl, idx(i + 1, "Su")] = Pl * lam_l
    Q[Sl, Su] = (1.0 - Pl) * lam_l

    Su = idx(N, "Su"); Sh = idx(N, "Sh")
    Sed = idx(N, "Sed"); Sad = idx(N, "Sad"); Sd = idx(N, "Sd")
    Q[Su, Sh] = (1.0 - Pc) * lam_d
    Q[Su, Sed] = Pc * lam_d
    Q[Sh, Su] = mu_m
    Q[Sed, Sad] = (1.0 - D) * lam_ed
    Q[Sed, Sd] = D * lam_ed
    Q[Sad, Su] = mu_am
    Q[Sd, Su] = mu_u

    for j in range(S):
        Q[j, j] = -Q[j, :].sum()

    su_indices = [offsets[i] for i in range(1, N + 1)]
    return Q, states, su_indices

# =====
# BM3 - SMR Extended (3D N×N×N grid)
# =====

def build_bm3_ideal(p: dict):
    """3 states: Su=0, Sd=1, Sm=2 (D=1, no hidden branch)."""
    Q = np.zeros((3, 3))
    Q[0, 1] = p["lambda_d"]
    Q[0, 2] = p["lambda_m"]
    Q[1, 0] = p["mu_u"]
    Q[2, 0] = p["mu_m"]
    for i in range(3):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sd", "Sm"]

def build_bm3_nolearn(p: dict):
    """5 states: Su=0, Sd=1, Sh=2, Shc=3, Sm=4 (fixed D, no Sl)."""
    lam_d = p["lambda_d"]; lam_m = p["lambda_m"]
    lam_hc = p["lambda_hc"]
    mu_u = p["mu_u"]; mu_m = p["mu_m"]; mu_h = p["mu_h"]
    D = p["D_start"]
    Q = np.zeros((5, 5))
    Q[0, 1] = D * lam_d
    Q[0, 2] = (1.0 - D) * lam_d
    Q[0, 4] = lam_m
    Q[1, 0] = mu_u
    Q[2, 3] = lam_hc
    Q[2, 4] = lam_m
    Q[3, 0] = mu_h
    Q[4, 0] = mu_m
    for i in range(5):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sd", "Sh", "Shc", "Sm"]

```

```

def _bm3_frag_states(r: int, c: int, d: int, N: int) -> list:
    """
    Local states for fragment (r,c,d).
    Boundary reductions prevent zero rows in Q:
        r=N-1 => S11 absent (Sm -> Su directly)
        c=N-1 => S12 absent (Sd -> Su directly)
        d=N-1 => S13 absent (Shc -> Su directly)
    Total states:  $8N^3 - 3N^2$ 
    """
    base = ["Su", "Sd", "Sh", "Shc", "Sm"]
    if r < N - 1: base.append("S11")
    if c < N - 1: base.append("S12")
    if d < N - 1: base.append("S13")
    return base

def build_bm3_learning(p: dict, N: int):
    """
    3D N×N×N grid CTMC. Three independent learning dimensions:
        S11 <- Sm      (maintenance)      -> Su_(r+1, c, d)    dim r
        S12 <- Sd      (detected failure) -> Su_(r, c+1, d)    dim c
        S13 <- Shc     (critical hidden)  -> Su_(r, c, d+1)    dim d

    D(r,c,d) = min(0.99, D_initial
                    + delta_D_dim1 · r/(N-1)
                    + delta_D_dim2 · c/(N-1)
                    + delta_D_dim3 · d/(N-1))
    """
    lam_d = p["lambda_d"]; lam_m = p["lambda_m"]; lam_hc = p["lambda_hc"]
    lam_l1 = p["lambda_l1"]; lam_l2 = p["lambda_l2"]; lam_l3 = p["lambda_l3"]
    mu_u = p["mu_u"]; mu_m = p["mu_m"]; mu_h = p["mu_h"]
    Plm = p["Plm"]; Pld = p["Pld"]; Plh = p["Plh"]
    D_init = p["D_initial"]
    ddim1 = p["delta_D_dim1"]
    ddim2 = p["delta_D_dim2"]
    ddim3 = p["delta_D_dim3"]

    frag_local = {}
    frag_off = {}
    all_labels = []

    for r in range(N):
        for c in range(N):
            for d in range(N):
                local = _bm3_frag_states(r, c, d, N)
                frag_local[(r, c, d)] = local
                frag_off[(r, c, d)] = len(all_labels)
                for s in local:
                    all_labels.append(f"{s}_{r}{c}{d}")

    S = len(all_labels)
    Q = np.zeros((S, S))

    def gidx(r, c, d, name):
        lst = frag_local[(r, c, d)]
        if name not in lst:
            return None
        return frag_off[(r, c, d)] + lst.index(name)

    denom = max(1, N - 1)
    for r in range(N):
        for c in range(N):
            for d in range(N):
                last_r = (r == N - 1)
                last_c = (c == N - 1)
                last_d = (d == N - 1)

```

```

D = min(0.99, D_init
        + ddim1 * (r / denom)
        + ddim2 * (c / denom)
        + ddim3 * (d / denom))

Su = gidx(r, c, d, "Su")
Sd = gidx(r, c, d, "Sd")
Sh = gidx(r, c, d, "Sh")
Shc = gidx(r, c, d, "Shc")
Sm = gidx(r, c, d, "Sm")
S11 = gidx(r, c, d, "S11")
S12 = gidx(r, c, d, "S12")
S13 = gidx(r, c, d, "S13")

Q[Su, Sd] = D * lam_d
Q[Su, Sh] = (1.0 - D) * lam_d
Q[Su, Sm] = lam_m
Q[Sh, Shc] = lam_hc
Q[Sh, Sm] = lam_m

if not last_c:
    Q[Sd, Su] = (1.0 - Pld) * mu_u
    Q[Sd, S12] = Pld * mu_u
else:
    Q[Sd, Su] = mu_u

if not last_d:
    Q[Shc, Su] = (1.0 - Plh) * mu_h
    Q[Shc, S13] = Plh * mu_h
else:
    Q[Shc, Su] = mu_h

if not last_r:
    Q[Sm, Su] = (1.0 - Plm) * mu_m
    Q[Sm, S11] = Plm * mu_m
else:
    Q[Sm, Su] = mu_m

if not last_r:
    Q[S11, gidx(r + 1, c, d, "Su")] = lam_l1
if not last_c:
    Q[S12, gidx(r, c + 1, d, "Su")] = lam_l2
if not last_d:
    Q[S13, gidx(r, c, d + 1, "Su")] = lam_l3

for j in range(S):
    Q[j, j] = -Q[j, :].sum()

su_indices = [gidx(r, c, d, "Su")
               for r in range(N) for c in range(N) for d in range(N)]
return Q, all_labels, frag_off, frag_local, su_indices

# =====
# BM4 - SMR Coolant Pump (2D N×N grid)
# =====

def build_bm4_ideal(p: dict):
    """3 states: Su=0, Sd=1, Sm=2 (D=1, no hidden branch)."""
    Q = np.zeros((3, 3))
    Q[0, 1] = p["lambda_d"]
    Q[0, 2] = p["lambda_m"]
    Q[1, 0] = p["mu_u"]
    Q[2, 0] = p["mu_m"]
    for i in range(3):

```

```

        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sd", "Sm"]

def build_bm4_nolearn(p: dict):
    """5 states: Su=0, Sd=1, Sh=2, Shc=3, Sm=4 (fixed D, no S1)."""
    lam_d = p["lambda_d"]; lam_m = p["lambda_m"]
    lam_hc = p["lambda_hc"]
    mu_u = p["mu_u"]; mu_m = p["mu_m"]; mu_h = p["mu_h"]
    D = p["D_start"]
    Q = np.zeros((5, 5))
    Q[0, 1] = D * lam_d
    Q[0, 2] = (1.0 - D) * lam_d
    Q[0, 4] = lam_m
    Q[1, 0] = mu_u
    Q[2, 3] = lam_hc
    Q[2, 4] = lam_m
    Q[3, 0] = mu_h
    Q[4, 0] = mu_m
    for i in range(5):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sd", "Sh", "Shc", "Sm"]

def _bm4_frag_state_names(r: int, c: int, N: int) -> list:
    """
    Local states for fragment (r,c).
    Boundary reductions prevent zero rows in Q:
        r=N-1 => S11 absent (Sm/Shc -> Su directly)
        c=N-1 => S12 absent (Sd -> Su directly)
Total states:  $7N^2 - 2N - 2$ 
    """
    base = ["Su", "Sd", "Sh", "Shc", "Sm"]
    if r < N - 1: base.append("S11")
    if c < N - 1: base.append("S12")
    return base

def build_bm4_learning(p: dict, N: int):
    """
    2D N×N grid CTMC. Two learning dimensions:
        S11 <- Sm/Shc (maintenance / critical hidden) -> Su_(r+1, c) dim r
        S12 <- Sd (detected failure) -> Su_(r, c+1) dim c

    D(r,c) = min(0.99, D_initial
                  + delta_D_dim1 · r/(N-1)
                  + delta_D_dim2 · c/(N-1))
    """
    lam_d = p["lambda_d"]; lam_m = p["lambda_m"]; lam_hc = p["lambda_hc"]
    lam_l1 = p["lambda_l1"]; lam_l2 = p["lambda_l2"]
    mu_u = p["mu_u"]; mu_m = p["mu_m"]; mu_h = p["mu_h"]
    Plm = p["Plm"]; Pld = p["Pld"]; Plh = p["Plh"]
    D_init = p["D_initial"]
    ddim1 = p["delta_D_dim1"]
    ddim2 = p["delta_D_dim2"]

    frag_local = {}
    frag_off = {}
    all_labels = []

    for r in range(N):
        for c in range(N):
            local = _bm4_frag_state_names(r, c, N)
            frag_local[(r, c)] = local
            frag_off[(r, c)] = len(all_labels)
            for s in local:

```

```

        all_labels.append(f"{s}_{r}{c}")

S = len(all_labels)
Q = np.zeros((S, S))

def gidx(r, c, name):
    lst = frag_local[(r, c)]
    if name not in lst:
        return None
    return frag_off[(r, c)] + lst.index(name)

denom = max(1, N - 1)
for r in range(N):
    for c in range(N):
        last_r = (r == N - 1)
        last_c = (c == N - 1)

        D = min(0.99, D_init
                + ddim1 * (r / denom)
                + ddim2 * (c / denom))

        Su = gidx(r, c, "Su")
        Sd = gidx(r, c, "Sd")
        Sh = gidx(r, c, "Sh")
        Shc = gidx(r, c, "Shc")
        Sm = gidx(r, c, "Sm")
        Sl1 = gidx(r, c, "Sl1")
        Sl2 = gidx(r, c, "Sl2")

        Q[Su, Sd] = D * lam_d
        Q[Su, Sh] = (1.0 - D) * lam_d
        Q[Su, Sm] = lam_m
        Q[Sh, Shc] = lam_hc
        Q[Sh, Sm] = lam_m

        if not last_c:
            Q[Sd, Su] = (1.0 - Pld) * mu_u
            Q[Sd, Sl2] = Pld * mu_u
        else:
            Q[Sd, Su] = mu_u

        if not last_r:
            Q[Shc, Su] = (1.0 - Plh) * mu_h
            Q[Shc, Sl1] = Plh * mu_h
        else:
            Q[Shc, Su] = mu_h

        if not last_r:
            Q[Sm, Su] = (1.0 - Plm) * mu_m
            Q[Sm, Sl1] = Plm * mu_m
        else:
            Q[Sm, Su] = mu_m

        if not last_r:
            Q[Sl1, gidx(r + 1, c, "Su")] = lam_l1
        if not last_c:
            Q[Sl2, gidx(r, c + 1, "Su")] = lam_l2

    for j in range(S):
        Q[j, j] = -Q[j, :].sum()

su_indices = [gidx(r, c, "Su") for r in range(N) for c in range(N)]
return Q, all_labels, frag_off, frag_local, su_indices

```

```
# =====
```

```

# BM5 - SMR Federal Learning (3D N×N×N grid + diagonal Slf)
# =====

def build_bm5_ideal(p: dict):
    """3 states: Su=0, Sd=1, Sm=2 (D=1, no hidden branch)."""
    Q = np.zeros((3, 3))
    Q[0, 1] = p["lambda_d"]
    Q[0, 2] = p["lambda_m"]
    Q[1, 0] = p["mu_u"]
    Q[2, 0] = p["mu_m"]
    for i in range(3):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sd", "Sm"]

def build_bm5_nolearn(p: dict):
    """5 states: Su=0, Sd=1, Sh=2, Shc=3, Sm=4 (fixed D, no Sl)."""
    lam_d = p["lambda_d"]; lam_m = p["lambda_m"]
    lam_hc = p["lambda_hc"]
    mu_u = p["mu_u"]; mu_m = p["mu_m"]; mu_h = p["mu_h"]
    D = p["D_start"]
    Q = np.zeros((5, 5))
    Q[0, 1] = D * lam_d
    Q[0, 2] = (1.0 - D) * lam_d
    Q[0, 4] = lam_m
    Q[1, 0] = mu_u
    Q[2, 3] = lam_hc
    Q[2, 4] = lam_m
    Q[3, 0] = mu_h
    Q[4, 0] = mu_m
    for i in range(5):
        Q[i, i] = -Q[i, :].sum()
    return Q, ["Su", "Sd", "Sh", "Shc", "Sm"]

def _bm5_frag_states(r: int, c: int, d: int, N: int) -> list:
    """
    Local states for fragment (r,c,d).
    Same boundary reductions as BM3, plus:
        Slf present only when r < N-1 AND c < N-1 AND d < N-1
        (diagonal jump requires all three dims to have room)
    Total states:  $8N^3 - 3N^2 + (N-1)^3$ 
    """
    base = ["Su", "Sd", "Sh", "Shc", "Sm"]
    if r < N - 1: base.append("Sl1")
    if c < N - 1: base.append("Sl2")
    if d < N - 1: base.append("Sl3")
    if r < N - 1 and c < N - 1 and d < N - 1:
        base.append("Slf")
    return base

def build_bm5_learning(p: dict, N: int):
    """
    BM5 = BM3 + federal learning state Slf from Sm.
    Slf gives a diagonal jump: (r,c,d) -> (r+1, c+1, d+1).

    Sm routing (federal priority):
        if Slf present (all dims < N-1):
            Sm -> Slf : Plf · mu_m
            Sm -> Sl1 : (1-Plf) · Plm · mu_m (if r < N-1)
            Sm -> Su : (1-Plf) · (1-Plm) · mu_m
        else:
            Sm -> Sl1 : Plm · mu_m (if r < N-1)
            Sm -> Su : (1-Plm) · mu_m
    """

```

```

lam_d = p["lambda_d"]; lam_m = p["lambda_m"]; lam_hc = p["lambda_hc"]
lam_l1 = p["lambda_l1"]; lam_l2 = p["lambda_l2"]; lam_l3 = p["lambda_l3"]
lam_lf = p["lambda_lf"]
mu_u = p["mu_u"]; mu_m = p["mu_m"]; mu_h = p["mu_h"]
Plm = p["Plm"]; Pld = p["Pld"]; Plh = p["Plh"]
Plf = p["Plf"]
D_init = p["D_initial"]
ddim1 = p["delta_D_dim1"]
ddim2 = p["delta_D_dim2"]
ddim3 = p["delta_D_dim3"]

frag_local = {}
frag_off = {}
all_labels = []

for r in range(N):
    for c in range(N):
        for d in range(N):
            local = _bm5_frag_states(r, c, d, N)
            frag_local[(r, c, d)] = local
            frag_off[(r, c, d)] = len(all_labels)
            for s in local:
                all_labels.append(f"{s}_{r}{c}{d}")

S = len(all_labels)
Q = np.zeros((S, S))

def gidx(r, c, d, name):
    lst = frag_local[(r, c, d)]
    if name not in lst:
        return None
    return frag_off[(r, c, d)] + lst.index(name)

denom = max(1, N - 1)
for r in range(N):
    for c in range(N):
        for d in range(N):
            last_r = (r == N - 1)
            last_c = (c == N - 1)
            last_d = (d == N - 1)
            has_slf = (not last_r) and (not last_c) and (not last_d)

            D = min(0.99, D_init
                    + ddim1 * (r / denom)
                    + ddim2 * (c / denom)
                    + ddim3 * (d / denom))

            Su = gidx(r, c, d, "Su")
            Sd = gidx(r, c, d, "Sd")
            Sh = gidx(r, c, d, "Sh")
            Shc = gidx(r, c, d, "Shc")
            Sm = gidx(r, c, d, "Sm")
            Sl1 = gidx(r, c, d, "Sl1")
            Sl2 = gidx(r, c, d, "Sl2")
            Sl3 = gidx(r, c, d, "Sl3")
            Slf = gidx(r, c, d, "Slf")

            Q[Su, Sd] = D * lam_d
            Q[Su, Sh] = (1.0 - D) * lam_d
            Q[Su, Sm] = lam_m
            Q[Sh, Shc] = lam_hc
            Q[Sh, Sm] = lam_m

            if not last_c:
                Q[Sd, Su] = (1.0 - Pld) * mu_u
                Q[Sd, Sl2] = Pld * mu_u

```

```

else:
    Q[Sd, Su] = mu_u

if not last_d:
    Q[Shc, Su] = (1.0 - Plh) * mu_h
    Q[Shc, Sl3] = Plh * mu_h
else:
    Q[Shc, Su] = mu_h

if has_slf:
    Q[Sm, Slf] = Plf * mu_m
    if not last_r:
        Q[Sm, Sl1] = (1.0 - Plf) * Plm * mu_m
        Q[Sm, Su] = (1.0 - Plf) * (1.0 - Plm) * mu_m
    else:
        Q[Sm, Su] = (1.0 - Plf) * mu_m
else:
    if not last_r:
        Q[Sm, Sl1] = Plm * mu_m
        Q[Sm, Su] = (1.0 - Plm) * mu_m
    else:
        Q[Sm, Su] = mu_m

if not last_r:
    Q[Sl1, gidx(r + 1, c, d, "Su")] = lam_l1
if not last_c:
    Q[Sl2, gidx(r, c + 1, d, "Su")] = lam_l2
if not last_d:
    Q[Sl3, gidx(r, c, d + 1, "Su")] = lam_l3
if has_slf:
    Q[Slf, gidx(r + 1, c + 1, d + 1, "Su")] = lam_lf

for j in range(S):
    Q[j, j] = -Q[j, :].sum()

su_indices = [gidx(r, c, d, "Su")
               for r in range(N) for c in range(N) for d in range(N)]
return Q, all_labels, frag_off, frag_local, su_indices

# =====
# Unified dispatcher
# =====

def build_model(model_key: str, p: dict, N: int):
    """
    Returns (Q, labels, su_indices).
    su_indices: global row indices of all Su states.
    Availability A(t) = P(t)[:, su_indices].sum(axis=1)
    """
    dispatch = {
        "BM1-Ideal": lambda: (*build_bm1_ideal(p), ) + ([0],),
        "BM1-NoLearning": lambda: (*build_bm1_nolearn(p), ) + ([0],),
        "BM2-Ideal": lambda: (*build_bm2_ideal(p), ) + ([0],),
        "BM2-NoLearning": lambda: (*build_bm2_nolearn(p), ) + ([0],),
        "BM3-Ideal": lambda: (*build_bm3_ideal(p), ) + ([0],),
        "BM3-NoLearning": lambda: (*build_bm3_nolearn(p), ) + ([0],),
        "BM4-Ideal": lambda: (*build_bm4_ideal(p), ) + ([0],),
        "BM4-NoLearning": lambda: (*build_bm4_nolearn(p), ) + ([0],),
        "BM5-Ideal": lambda: (*build_bm5_ideal(p), ) + ([0],),
        "BM5-NoLearning": lambda: (*build_bm5_nolearn(p), ) + ([0],),
    }
    if model_key in dispatch:
        Q, lbl, su = dispatch[model_key]()
        return Q, lbl, su

```

```

if model_key == "BM1-Learning":
    Q, lbl, su = build_bm1_learning(p, N)
    return Q, lbl, su
if model_key == "BM2-Learning":
    Q, lbl, su = build_bm2_learning(p, N)
    return Q, lbl, su
if model_key == "BM3-Learning":
    Q, lbl, _, _, su = build_bm3_learning(p, N)
    return Q, lbl, su
if model_key == "BM4-Learning":
    Q, lbl, _, _, su = build_bm4_learning(p, N)
    return Q, lbl, su
if model_key == "BM5-Learning":
    Q, lbl, _, _, su = build_bm5_learning(p, N)
    return Q, lbl, su

raise ValueError(f"Unknown model_key: {model_key!r}")

# =====
# Stiffness
# =====

def compute_stiffness(Q: np.ndarray) -> float:
    """s(x) = max|Q_ii| / min(|Q_ii| where Q_ii != 0)"""
    d = np.abs(np.diag(Q))
    nz = d[d > 1e-15]
    if len(nz) < 2:
        return 1.0
    return float(nz.max() / nz.min())

# =====
# Solver router
# =====

def select_solver(choice: str, stiffness: float, eps: float):
    """Returns (method, reason_str, was_auto_switched)."""
    if choice != "Auto (Tool Justification)":
        MAP = {
            "RK45 - Explicit, Fast": ("RK45", "Manual: RK45.", False),
            "BDF - L-stable, Stiff": ("BDF", "Manual: BDF.", False),
            "Radau - Implicit, High Precision": ("Radau", "Manual: Radau.", False),
        }
        return MAP[choice]
    if stiffness < 1e3:
        return "RK45", f"Auto: s(x)={stiffness:.2f}<103 → RK45.", False
    if eps >= 1e-6:
        return "BDF", f"Auto: s(x)={stiffness:.2f}≥103, ε≥10-6 → BDF.", True
    return "Radau", f"Auto: s(x)={stiffness:.2f}≥103, ε<10-6 → Radau.", True

# =====
# ODE solver dp/dt = Q^T · p
# =====

def solve_ctmc(Q: np.ndarray, t_end: float, method: str,
               n_points: int = 800):
    """
    Solves Kolmogorov-Chapman forward equations.
    Initial condition: p(0) = e_0 (system starts in Su).
    Returns (t_array, P_matrix) where P[k, :] is state dist at t[k].
    """
    S = Q.shape[0]
    p0 = np.zeros(S); p0[0] = 1.0
    sol = solve_ivp(
        fun=lambda t, p: Q.T @ p,

```

```

        t_span=[0.0, t_end],
        y0=p0,
        method=method,
        t_eval=np.linspace(0.0, t_end, n_points),
        rtol=1e-9,
        atol=1e-11,
    )
    if not sol.success:
        raise RuntimeError(f"solve_ivp failed ({method}): {sol.message}")
    return sol.t, sol.y.T # (n_points,), (n_points, S)

def availability_from_P(P: np.ndarray, su_indices: list) -> np.ndarray:
    """A(t) = sum over all Su-state columns."""
    return P[:, su_indices].sum(axis=1)

# =====
# Analytical steady-state  $\pi Q = 0, \Sigma \pi = 1$ 
# =====

def analytical_steady_state(Q: np.ndarray, su_indices: list):
    """
    Returns (A_steady, pi_vector).

    Accuracy guard: verifies Q has no zero rows before calling null_space.
    Zero rows indicate absorbing states - boundary reductions incomplete.
    """
    zero_rows = np.where(np.abs(Q).sum(axis=1) < 1e-15)[0]
    if len(zero_rows) > 0:
        raise ValueError(
            f"Q has zero row(s) at indices {zero_rows.tolist()}. "
            "Absorbing states detected - boundary reductions incomplete. "
            "Steady-state is undefined or not unique."
        )
    ns = null_space(Q.T)
    if ns.shape[1] != 1:
        raise ValueError(
            f"null_space dimension = {ns.shape[1]} (expected 1). "
            "Q may not be irreducible. Check model parameters."
        )
    pi = np.abs(ns[:, 0])
    pi /= pi.sum()
    return float(pi[su_indices].sum()), pi

# =====
# Q matrix validation
# =====

def validate_Q(Q: np.ndarray):
    """Returns (is_valid, messages_list). Checks all row sums = 0."""
    row_sums = Q.sum(axis=1)
    msgs = []
    ok = True
    for i, rs in enumerate(row_sums):
        if abs(rs) > 1e-9:
            msgs.append(f"Row {i}: sum={rs:.2e} FAIL")
            ok = False
        else:
            msgs.append(f"Row {i}: sum={rs:.2e} OK")
    return ok, msgs

# =====
# V&V Auditor
# =====

```

```

def vv_auditor(A_sim: float, A_ana: float, switched: bool,
               tol: float = 0.01):
    """
    Returns (status, message). status ∈ {'green', 'yellow', 'red'}.

    Compares simulated steady-state A_sim with analytical A_ana.
    """
    diff = abs(A_sim - A_ana)
    if diff > tol:
        return ("red",
               f"DIVERGED: |A_sim({A_sim:.6f}) - A_analytical({A_ana:.6f})| "
               f"= {diff:.2e} > tol={tol:.2e}. Increase T or N.")
    if switched:
        return ("yellow",
               f"CONVERGED (auto-switch): |ΔA| = {diff:.2e}. "
               "Solver switched to implicit due to stiffness s(x) ≥ 10³.")
    return ("green",
           f"CONVERGED: |A_sim({A_sim:.6f}) - A_analytical({A_ana:.6f})| "
           f"= {diff:.2e} < tol={tol:.2e}.")

# =====
# Core Availability Metrics
# =====

def compute_core_metrics(t, A_learn, A_nolearn, A_ideal,
                        A_steady_learn, t_std_tol: float = 0.95):
    """
    Compute six core availability metrics.

    Parameters
    -----
    t          : time array
    A_learn    : A(t) of Learning model
    A_nolearn  : A(t) of No-Learning model
    A_ideal    : A(t) of Ideal model (unused internally, reserved)
    A_steady_learn : analytical steady-state of Learning model (πQ=0)
    t_std_tol  : fraction for T_std threshold (default 0.95)

    Returns
    -----
    dict with keys:
        A_steady - analytical steady-state availability (Learning)
        A_min    - minimum A(t) of Learning curve (worst startup dip)
        LoA_max  - max(A_nolearn(t) - A_learn(t)) (deepest gap)
        T_loss   - time until Learning ≥ No-Learning (catch-up time)
        T_std    - first t where A_learn ≥ A_min + (A_steady - A_min)·t_std_tol
                   and stays there (transition time)
        DUrel    - relative unavailability reduction (%):
                   (U_nolearn - U_learn) / U_nolearn × 100
    """
    t1 = np.asarray(t, dtype=float)
    A1 = np.asarray(A_learn, dtype=float)
    An = np.asarray(A_nolearn, dtype=float)

    A_steady = float(A_steady_learn)
    A_min = float(A1.min())

    # LoA_max
    gap = An - A1
    LoA_max = float(gap.max()) if gap.max() > 0 else 0.0
    # T_loss - first crossing where Learning ≥ No-Learning after initial dip
    below = A1 < An
    if not below.any():
        T_loss = 0.0

```

```

else:
    crossings = np.where(np.diff(below.astype(int)) == -1)[0]
    T_loss = float(tl[-1]) if len(crossings) == 0 \
        else float(tl[crossings[0] + 1])

# T_std - first t where A_learn reaches t_std_tol of the [A_min, A_steady] range
threshold = A_min + (A_steady - A_min) * t_std_tol
above_thresh = A1 >= threshold
if not above_thresh.any():
    T_std = float("inf")
else:
    last_below = np.where(~above_thresh)[0]
    if len(last_below) == 0:
        T_std = 0.0
    else:
        idx = last_below[-1] + 1
        T_std = float(tl[idx]) if idx < len(tl) else float(tl[-1])

# DUrel - relative unavailability reduction
A_nolearn_ss = float(A_n[-1])
U_stat = 1.0 - A_nolearn_ss
U_learn = 1.0 - A_steady
DUrel = ((U_stat - U_learn) / U_stat * 100.0) \
    if U_stat > 1e-12 else float("nan")

return dict(
    A_steady=A_steady,
    A_min=A_min,
    LoA_max=LoA_max,
    T_loss=T_loss,
    T_std=T_std,
    DUrel=DUrel,
)

# =====
# KPI narrative report
# =====

def kpi_report(t, A_learn, A_nolearn, A_ideal):
    """Returns a Markdown string with a qualitative KPI summary."""
    Al, An, Ai = float(A_learn[-1]), float(A_nolearn[-1]), float(A_ideal[-1])
    Am_l = float(A_learn.min()); Am_n = float(A_nolearn.min())
    delta = Al - An

    lines = ["### 📊 KPI Narrative"]
    if Al < An - 1e-4:
        lines.append(f"- 🚩 **A degraded by learning path**: "
            f"A_learn={Al:.6f} < A_nolearn={An:.6f} (Δ={delta:.6f})")
    elif abs(delta) < 0.001:
        lines.append(f"- 🟦 **A near the same**: |ΔA|={abs(delta):.6f} < 0.001. "
            "Learning has marginal steady-state impact.")
    else:
        lines.append(f"- ✅ **Learning improves A**: "
            f"A_learn={Al:.6f} vs A_nolearn={An:.6f} (+{delta:.6f})")

    if Am_l < Am_n * 0.85:
        lines.append(f"- 🚫 **Significant startup loss**: "
            f"A_min_learn={Am_l:.6f} ≪ A_min_nolearn={Am_n:.6f}")
    else:
        lines.append(f"- 🟢 **Startup loss acceptable**: "
            f"A_min_learn={Am_l:.6f} ≈ A_min_nolearn={Am_n:.6f}")

    lines.append(f"- 📏 **Gap to ideal**: ΔA_ideal = {abs(Al - Ai):.6f}")
    return "\n".join(lines)

```