

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»
Факультет автоматизації виробництва, інформаційних
та управлінських технологій
Кафедра інформаційних технологій та аналітики даних

«Допущено до захисту»
Гарант ОПП

Ірина ГЕТЬМАН

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

за підсумками виконання
освітньо-професійної програми
«Комп'ютерні науки»
за спеціальністю 122 Комп'ютерні науки

**на тему «Програмно-методичний комплекс для інформаційної
підтримки запобігання термошоку доменних печей при екстремому
знеструмленні»**

Керівник роботи

Олександр КОСТІКОВ

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

Здобувач

Олександр СМІК

Підсумкова оцінка за атестацію			
--------------------------------	--	--	--

Голова ЕК

Антон КУДРЯВЦЕВ

ЗАПОРІЖЖЯ 2026

Факультет	ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»
Кафедра	автоматизації виробництва, інформаційних та управлінських технологій
Ступінь вищої освіти	інформаційних технологій та аналітики даних
Спеціальність	бакалавр
ОПП	122 Комп'ютерні науки
	Комп'ютерні науки

ЗАТВЕРДЖУЮ
Гарант ОПП

_____ Ірина ГЕТЬМАН

«26» лютого 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Смика Олександра Сергійовича

(прізвище, ім'я, по батькові здобувача)

1. Тема роботи «Програмно-методичний комплекс для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні»

керівник роботи, _____ Костіков Олександр Анатолійович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом Університету від 23.02.2026 р. №41/23.02.2026

2. Термін подання роботи 16.06.2026 р.

3. Вихідні дані до роботи Навчальна література, державні стандарти (зокрема ДСТУ 8302:2015, ДСТУ 2156-93, ІЕС 61511) та методичні вказівки з дисциплін «Системний аналіз», «Інженерія програмного забезпечення», «Бази даних», «Проектування інформаційних систем», «Методологія та організація наукових досліджень»; наукові публікації з тематики моделювання теплових процесів у доменних печах, методів моніторингу та прогнозування температурних полів, автоматизації металургійного виробництва; технічна документація з розробки ПЗ на Python та платформі FastAPI; документація з використання Power BI для візуалізації даних; внутрішні нормативні документи ПРАТ «КАМЕТ-СТАЛЬ» (технологічні інструкції, регламенти дій персоналу при екстремому знеструмленні); результати аналізу технологічного процесу доменного виробництва, виконаного під час переддипломної практики; результати власного проектування, алгоритмічної реалізації, чисельного моделювання, тестування програмного прототипу та економічного обґрунтування впровадження ПМК.

4. Зміст пояснювальної записки (перелік питань) Реферат. Зміст. Перелік умовних позначень. Вступ. Розділ 1. Аналіз стану питання, концепцій з проблеми, що розглядається. Розділ 2. Розробка моделі об'єкта та методики дослідження. Розділ 3. Розробка проекту програмно-методичного комплексу та його реалізація. Розділ 4. Економічні розрахунки. Висновки. Список використаних джерел. Додатки.

5. Перелік графічного (демонстраційного) матеріалу (з точним зазначенням обов'язкових креслень): Актуальність, мета, об'єкт, предмет та завдання дослідження. Характеристика предметної області: технологічна схема доменної печі та критичні зони. Діаграма послідовності подій при екстремому знеструмленні. Архітектура програмно-методичного комплексу (трирівнева). UML-діаграми: діаграма прецедентів використання, діаграма діяльності, діаграма класів, діаграма розгортання. Схема бази

даних SQLite (ER-діаграма). Фрагменти програмного коду (ініціалізація БД, прогнозування, оцінка ризику). Дашборди Power BI (4 сторінки) у трьох режимах роботи (штатний, коливання, аварія) із умовним форматуванням. Результати економічних розрахунків (витрати на розробку, річна економія, NPV, ROI, строк окупності). Висновки до роботи.

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх.

Розділ	Прізвище, ініціали та посада консультанта
1	Костіков О.А., доц. каф. ІНТЕХАД
2	Шпунт Д.А., майстер з ремонту устаткування доменного цеху ПРАТ «Камет-Сталь»
3	Костіков О.А., доц. каф. ІНТЕХАД
4	Гетьман І.А., доц. каф. ІНТЕХАД

7. Дата видачі завдання 26.02.2026

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи
1	Розділ 1. Аналіз стану питання, концепцій проблеми термошоку доменних печей при екстремому знеструмленні та сучасних інформаційних технологій	26.02.2026 - 20.03.2026
2	Розділ 2. Розробка математичної (теплової) моделі футерівки доменної печі та методики дослідження	21.03.2026 - 10.04.2026
3	Розділ 3. Проектування та реалізація програмно-методичного комплексу для інформаційної підтримки запобігання термошоку	11.04.2026 - 25.05.2026
5	Розділ 4. Економічна доцільність розробки програмно-методичного комплексу	26.05.2026 - 05.06.2026
6	Висновки, перелік посилань, вступ, зміст, реферат	06.06.2026 - 12.06.2026
7	Подання завершеної роботи. Перевірка на академічний плагіат	13.06.2026 - 16.06.2026
8	Остаточне оформлення роботи, презентаційного матеріалу	16.06.2026 - 18.06.2026
9	Рецензування завершеної роботи. Захист	18.06.2026 - 23.06.2026

Здобувач

(Олександр СМІК)

Керівник роботи

(Олександр КОСТІКОВ)

РЕФЕРАТ

Смик О.С. Розробка програмно-методичного комплексу для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра комп'ютерних наук за спеціальністю 122 «Комп'ютерні науки». - ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» МОН України, м. Запоріжжя, 2026.

Мета роботи - розробка структурної моделі та алгоритмічного забезпечення програмно-методичного комплексу для запобігання термошоку футерівки доменних печей при екстремому знеструмленні.

Об'єкт дослідження - процеси інформаційної підтримки управління доменними печами в аварійних ситуаціях.

Предмет дослідження - методи та алгоритми моніторингу, прогнозування та підтримки прийняття рішень для запобігання термошоку футерівки.

Методологія - системний аналіз, математичне моделювання, об'єктно-орієнтоване проєктування з використанням UML, порівняльний аналіз технологій, економічна оцінка ефективності.

У роботі проведено аналіз технологічного процесу доменного виробництва на ПРАТ «КАМЕТ-СТАЛЬ», визначено критичні елементи печі. Розроблено математичну модель теплового стану футерівки з коефіцієнтами теплопереносу для трьох зон ($\alpha = 0,95; 0,85; 0,75$) та методику бальної оцінки ризику з п'ятьма рівнями. Створено програмний прототип на Python + FastAPI + SQLite + Power BI з REST API, симуляцією датчиків, прогнозуванням та генерацією рекомендацій. Виконано техніко-економічне обґрунтування: витрати - 225 000 грн, річна економія - понад 42 млн грн, строк окупності - менше 1 місяця.

Розроблений комплекс дозволяє зменшити ймовірність термошоку на 40 - 60%, скоротити час прийняття рішень персоналом та зменшити тривалість простою печі на 20 - 40%. Система придатна до масштабування. Результати можуть бути використані на ПРАТ «КАМЕТ-СТАЛЬ» та інших металургійних підприємствах.

КЛЮЧОВІ СЛОВА: ПРОГРАМНО-МЕТОДИЧНИЙ КОМПЛЕКС, ДОМЕННА ПІЧ, ТЕРМОШОК ФУТЕРІВКИ, ЗНЕСТРУМЛЕННЯ, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, ОЦІНКА РИЗИКУ, UML, FASTAPI, POWER BI.

ABSTRACT

Smyk O.S. Development of a Software-Methodological Complex for Information Support to Prevent Thermal Shock of Blast Furnaces during Emergency Power Outage.

Qualification work for obtaining the degree of Bachelor of Computer Science in the specialty 122 Computer Science. - LLC «TECHNICAL UNIVERSITY «METINVEST POLYTECHNIC» MES of Ukraine, Zaporizhzhia, 2026.

The aim of this work is to develop a structural model and algorithmic support for a software-methodological complex to prevent thermal shock of blast furnace lining during emergency power outage.

The object of research is the processes of information support for blast furnace control in emergency situations.

The subject of research is methods and algorithms for monitoring, forecasting and decision-making support to prevent thermal shock of blast furnace lining.

Methodology - system analysis, mathematical modeling, object-oriented design using UML, comparative analysis of technologies, economic evaluation of efficiency.

The study includes an analysis of the technological process at KAMET-STEEL PJSC and identification of critical furnace elements. A mathematical model of the lining thermal state with heat transfer coefficients for three zones ($\alpha = 0.95$; 0.85 ; 0.75) and a point-based risk assessment methodology with five risk levels were developed. A software prototype was created using Python + FastAPI + SQLite + Power BI with REST API, sensor simulation, forecasting and recommendation generation. A feasibility study was performed: costs - UAH 225,000, annual savings - over UAH 42 million, payback period - less than 1 month.

The developed complex allows reducing the probability of thermal shock by 40 - 60%, reducing personnel decision-making time and reducing furnace downtime by 20 - 40%. The system is scalable. The results can be used at KAMET-STEEL PJSC and other metallurgical enterprises.

KEYWORDS: SOFTWARE-METHODOLOGICAL COMPLEX, BLAST FURNACE, LINING THERMAL SHOCK, POWER OUTAGE, MATHEMATICAL MODELING, RISK ASSESSMENT, UML, FASTAPI, POWER BI.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ СТАНУ ПИТАННЯ, КОНЦЕПЦІЙ З ПРОБЛЕМИ, ЩО РОЗГЛЯДАЄТЬСЯ.....	14
1.1 Аналіз предметної області, сучасних принципів моделювання. Обґрунтування актуальності досліджень.	14
1.1.1 Загальна характеристика ПРАТ «КАМЕТ-СТАЛЬ» та доменного цеху.....	14
1.1.2 Технологічний процес роботи доменної печі та системи автоматизації.....	16
1.1.3 Термошок футерівки доменної печі при екстремому знеструмленні: сутність, причини та наслідки	20
1.1.4 Глосарій предметної області	27
1.2 Аналіз сучасних інформаційних технологій, технологій та засобів розробки програмного забезпечення	29
1.2.1 Аналіз поточних інформаційних систем та алгоритмів на підприємстві.....	29
1.2.2 Порівняльний аналіз сучасних методів моніторингу та прогнозування температурного стану	32
1.2.3 Переваги та недоліки існуючих підходів	34
1.2.4 Висновки щодо проблем та напрямків удосконалення.....	36
2 КРИТИЧНИЙ АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ	38
2.1 Обґрунтування вибору методів теоретичних та експериментальних досліджень, програмного забезпечення	38
2.1.1 Теоретичні методи дослідження	38
2.1.2 Експериментальні методи дослідження.....	39
2.1.3 Інструментальні методи та програмне забезпечення	40

2.2 Математична модель об'єкта дослідження.....	42
2.2.1 Загальна постановка задачі	42
2.2.2 Ключові параметри моделі	43
2.2.3 Рівняння прогнозування температури.....	44
2.2.4 Розрахунок температурних градієнтів	45
2.2.5 Критерії термошоку.....	46
2.3 Розробка методики дослідження.....	47
2.3.1 Фактори, що впливають на термошок	47
2.3.2 Методика оцінки ризику (бальна система)	49
2.3.3 Методика обробки результатів дослідження	51
2.3.4 План чисельного експерименту	52
2.4 Технічне завдання на створення програмно-методичного комплексу.....	54
3. РОЗРОБКА ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ ТА ЙОГО РЕАЛІЗАЦІЯ	57
3.1 Розробка логічної моделі ПМК.....	57
3.1.1 Діаграма прецедентів використання.....	58
3.1.2 Діаграма діяльності	61
3.1.3 Діаграма класів	63
3.1.4 Діаграма розгортання.....	67
3.2 Розробка фізичної моделі проєкту ПМК	71
3.2.1 Загальна архітектура системи.....	71
3.2.2 Вибір технологічного стеку	73
3.2.3 Структура бази даних	74
3.2.4 Особливості роботи в умовах знеструмлення.....	78
3.3 Особливості реалізації програмних компонентів.....	80
3.3.1 Загальна структура backend-компоненту	81
3.3.2 Модуль збору даних та симуляція сенсорів.....	82
3.3.3 Реалізація математичної моделі прогнозування	85

3.3.4 Реалізація алгоритму оцінки ризику	86
3.3.5 API-ендпоінти	86
3.3.6 Консольне меню для керування сценаріями	89
3.3.7 Інтеграція з Power BI	90
3.4 Елементи інтерфейсу ПМК	91
3.4.1 Обґрунтування вибору Power BI	92
3.4.2 Структура дашбордів	92
3.4.3 Умовне форматування	93
3.4.4 Джерела даних та оновлення	93
3.4.5 Скріншоти дашбордів.....	94
3.4.6 Переваги та обмеження	94
4 ЕКОНОМІЧНІ РОЗРАХУНКИ.....	95
4.1 Техніко-економічне обґрунтування	95
4.2 Витрати на розробку та впровадження.....	95
4.2.1 Одноразові витрати (капітальні вкладення).....	95
4.2.2 Поточні витрати (експлуатаційні).....	96
4.3 Розрахунок очікуваного економічного ефекту	96
4.3.1 Вихідні дані для розрахунку (за даними ПРАТ «КАМЕТ-СТАЛЬ»)...	96
4.3.2 Прямий економічний ефект	96
4.3.3 Непрямий економічний ефект.....	97
4.4 Розрахунок інвестиційної привабливості	98
4.5 Висновки до економічних розрахунків	98
ВИСНОВКИ	100
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	104
ДОДАТОК А. ВІДОМОСТІ РОБОТИ	107
ДОДАТОК Б. ТЕХНІЧНЕ ЗАВДАННЯ НА СТВОРЕННЯ ПМК	108
ДОДАТОК В. UML-ДІАГРАМИ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ	116
ДОДАТОК Г. ЛІСТИНГИ ПРОГРАМНОГО КОДУ	121
ДОДАТОК Д. СКРІНШОТИ ДАШБОРДІВ POWER BI.....	130

ВСТУП

Актуальність теми. Сучасне доменне виробництво є основою чорної металургії України та визначальним чинником економічної стабільності галузі. Доменні печі працюють у режимі безперервного високотемпературного процесу, що потребує безвідмовної роботи систем енергопостачання, охолодження та автоматизованого управління. Однією з найбільш небезпечних аварійних ситуацій є екстрене знеструмлення, яке призводить до різкого порушення теплового балансу печі та виникнення термошоку футерівки - внутрішнього вогнетривкого облицювання.

Наслідки такого явища є критичними: виникнення значних термічних напружень, тріщиноутворення та руйнування футерівки, деформація металоконструкцій колошника та засипного апарату, що зумовлює тривалий (від кількох діб до кількох місяців) простій обладнання та спричиняє багатомільйонні економічні збитки. Зокрема, у липні 2025 року на заводі ArcelorMittal у Лас-Тручас (Мексика) аварійне відключення електроенергії призвело до незворотного пошкодження парових турбін, необхідних для безпечної роботи доменної печі, що зупинило виробництво на 4–6 місяців [1]. В Україні, на ПРАТ «КАМЕТ-СТАЛЬ» (м. Кам'янське), масштабні відключення електроенергії в січні 2026 року спричинили зупинку системи охолодження доменної печі на 3 - 5 годин, що призвело до руйнування фурм та зупинки печі на приблизно тиждень для ремонту [2]. Ці факти підтверджують, що проблема запобігання термошоку при екстремому знеструмленні є надзвичайно актуальною як для українських, так і для світових металургійних підприємств.

Крім того, як зазначається в сучасних наукових дослідженнях, чавунні холодильники доменних печей через свою погану термостійкість (poor thermal shock resistance) є особливо вразливими до нестабільних температурних коливань, а температура газу всередині печі є найзначнішим фактором, що впливає на їх перегрів та руйнування [3]. Це підкреслює необхідність розробки

систем, здатних у реальному часі відстежувати температурні поля та прогнозувати критичні зміни.

Актуальність теми також зумовлена відсутністю на більшості металургійних підприємств України, зокрема на ПРАТ «КАМЕТ-СТАЛЬ», інтегрованих програмно-методичних комплексів, які б забезпечували оперативний моніторинг температурного стану футерівки, прогнозування розвитку термошоку та автоматизовану підтримку прийняття рішень персоналом у перші критичні 10–15 хвилин після знеструмлення. Існуючі регламенти дій персоналу переважно базуються на ручному переведенні обладнання в безпечний режим та візуальному контролю, що не дозволяє оперативно реагувати на динаміку температурних змін і підвищує ризик людської помилки.

Таким чином, розробка програмно-методичного комплексу для інформаційної підтримки запобігання термошоку доменних печей при екстреному знеструмленні є своєчасним науково-прикладним завданням, що має значну практичну цінність для підвищення безпеки, надійності та економічної ефективності доменного виробництва.

Зв'язок роботи з науково-технічними програмами. Робота виконувалася на базі ПРАТ «КАМЕТ-СТАЛЬ» у межах переддипломної практики здобувача вищої освіти за спеціальністю 122 «Комп'ютерні науки» та відповідає напрямкам модернізації інформаційних технологій управління технологічними процесами металургійного виробництва, що є пріоритетними для групи компаній Metinvest.

Мета роботи - розробка структурної моделі та алгоритмічного забезпечення програмно-методичного комплексу для інформаційної підтримки запобігання термошоку футерівки доменних печей при екстреному знеструмленні на основі аналізу технологічного процесу та сучасних методів моніторингу і прогнозування температурних полів.

Об'єкт дослідження - процеси інформаційної підтримки управління доменними печами в аварійних ситуаціях, пов'язаних з екстремим знеструмленням.

Предмет дослідження - методи та алгоритми моніторингу, прогнозування та підтримки прийняття рішень для запобігання термошоку футерівки доменних печей при екстремому знеструмленні.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Виконати аналіз технологічного процесу роботи доменних печей, систем охолодження та автоматизації на ПРАТ «КАМЕТ-СТАЛЬ», визначити критичні елементи конструкції, вразливі до термошоку.

2. Провести огляд та критичний аналіз існуючих методів і систем захисту доменних печей від термошоку при екстремому знеструмленні, виявити їхні переваги та недоліки.

3. Обґрунтувати вибір методів математичного моделювання теплових процесів у футерівці та розробити математичну модель, що описує динаміку температурних полів в аварійному режимі.

4. Розробити методику дослідження процесів термошоку, визначити фактори впливу та план чисельного експерименту.

5. Розробити структурну модель програмно-методичного комплексу, алгоритми моніторингу, прогнозування та формування рекомендацій для персоналу.

6. Виконати техніко-економічне обґрунтування доцільності впровадження розробленого комплексу.

Методи дослідження. У роботі використано комплекс методів: системний аналіз для дослідження технологічних процесів доменного виробництва; методи математичного моделювання (рівняння теплопереносу, чисельні методи) для опису температурних полів; об'єктно-орієнтоване проектування з використанням нотації UML для моделювання архітектури програмного комплексу; методи порівняльного аналізу для оцінки існуючих технологій

моніторингу та прогнозування; економічні методи оцінки ефективності інвестицій (NPV, ROI, строк окупності).

Наукова новизна отриманих результатів:

1. Удосконалено математичну модель теплового стану футерівки доменної печі в аварійному режимі, яка, на відміну від існуючих, враховує комплексний вплив припинення подачі гарячого дуття, зупинки системи охолодження та нерівномірного остигання конструктивних елементів.

2. Вперше запропоновано алгоритмічне забезпечення прогнозування термошоку, що базується на моніторингу градієнтів температур у критичних зонах (колошник, засипний апарат, верхня частина шахти) та дозволяє визначати критичний час до настання незворотних змін.

3. Дістав подальшого розвитку метод інформаційної підтримки прийняття рішень для персоналу доменного цеху, що, на відміну від існуючих регламентних підходів, передбачає формування диференційованих рекомендацій трьох рівнів (низький, середній, високий ризик) на основі прогнозних розрахунків.

Практичне значення отриманих результатів. Розроблена структурна модель та алгоритмічне забезпечення програмно-методичного комплексу можуть бути використані на ПРАТ «КАМЕТ-СТАЛЬ» та інших металургійних підприємствах для підвищення ефективності інформаційної підтримки персоналу в аварійних ситуаціях. Впровадження комплексу дозволить:

- зменшити ймовірність виникнення термошоку футерівки при екстремому знеструмленні на 40 - 60%;
- скоротити час прийняття рішень персоналом у перші критичні 10 - 15 хвилин;
- зменшити тривалість простою печі після аварійного знеструмлення на 20 - 40%;
- знизити витрати на капітальний ремонт футерівки та відновлення обладнання.

Отримані результати можуть бути використані в подальших дослідженнях для розробки повноцінного програмного забезпечення з використанням технологій машинного навчання та інтелектуального аналізу даних.

Особистий внесок здобувача освіти. Усі результати, наведені в роботі, отримані автором самостійно. Зокрема, автором виконано: аналіз предметної області та існуючих рішень; формалізацію технологічного процесу; розробку математичної моделі та алгоритмів; побудову UML-моделей; виконання економічних розрахунків та обґрунтування практичної цінності результатів.

Апробація результатів. Результати дослідження було представлено на звітній науково-технічній конференції студентів ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» (м. Запоріжжя, 2026 р.).

Публікації. За результатами дослідження опубліковано тези доповіді [4].

1 АНАЛІЗ СТАНУ ПИТАННЯ, КОНЦЕПЦІЙ З ПРОБЛЕМИ, ЩО РОЗГЛЯДАЄТЬСЯ

1.1 Аналіз предметної області, сучасних принципів моделювання. Обґрунтування актуальності досліджень.

1.1.1 Загальна характеристика ПРАТ «КАМЕТ-СТАЛЬ» та доменного цеху

ПРАТ «КАМЕТ-СТАЛЬ» (до 2022 року - ПАТ «Дніпровський металургійний комбінат імені Ф. Е. Дзержинського») є одним з провідних металургійних підприємств України, що входить до групи компаній Metinvest - найбільшого гірничо-металургійного холдингу країни. Підприємство розташоване у м. Кам'янське Дніпропетровської області та має повний металургійний цикл, що включає коксохімічне, доменне, сталеплавильне та прокатне виробництва.

Основним виробничим підрозділом, безпосередньо пов'язаним з темою дослідження, є доменний цех, який забезпечує виплавку чавуну - базового продукту для подальшого сталеплавильного переділу. На момент проходження переддипломної практики в доменному цеху експлуатувалися три доменні печі, характеристики яких наведено в таблиці 1.1.

ДП-1М є основною доменною піччю підприємства. Її продуктивність становить приблизно 2893 тонни чавуну на добу, що забезпечує близько 55 - 60% загального виробництва чавуну на комбінаті [5]. Піч була введена в експлуатацію ще в 1932 році, проте за час своєї роботи неодноразово проходила капітальні реконструкції, наймасштабніша з яких була проведена у 2007 році. Під час реконструкції 2007 року було модернізовано систему охолодження, замінено футерівку на сучасні вогнетривкі матеріали підвищеної

термостійкості, а також впроваджено нові системи автоматизованого управління технологічним процесом [6].

Таблиця 1.1 - Основні характеристики доменних печей ПРАТ «КАМЕТ-СТАЛЬ»

Параметр	ДП-1М	ДП-9	ДП-12
Корисний об'єм, м ³	1513	1386	1715*
Рік введення в експлуатацію	1932	1956	1964
Рік останньої реконструкції	2007	2011	2016
Продуктивність, т/добу	~2893	~2200	- **
Кількість фурм, шт.	18	16	20
Діаметр горна, м	8,6	8,0	9,2
Висота печі, м	~72	~68	~75

**За даними [5]. ДП-12 не експлуатується з 2019 року через незадовільний технічний стан та економічну недоцільність ремонту.*

***Продуктивність не наводиться через зупинку печі.*

Доменні печі обладнані сучасними системами автоматизованого управління технологічним процесом (АСУ ТП), що забезпечують контроль основних виробничих параметрів: температури гарячого дуття, тиску газу, рівня шихти, складу колошникового газу, температури чавуну та шлаку, а також стану футерівки за даними термопар. Система АСУ ТП на ПРАТ «КАМЕТ-СТАЛЬ» виконана на базі програмно-апаратного комплексу Siemens S7-400 та АРМ операторів з використанням SCADA-системи WinCC [6].

1.1.2 Технологічний процес роботи доменної печі та системи автоматизації

Доменна піч (рисунок 1.1) являє собою вертикальну шахтну піч, призначену для виплавки чавуну із залізрудних матеріалів шляхом безперервного відновного процесу.

Призначення елементів печі:

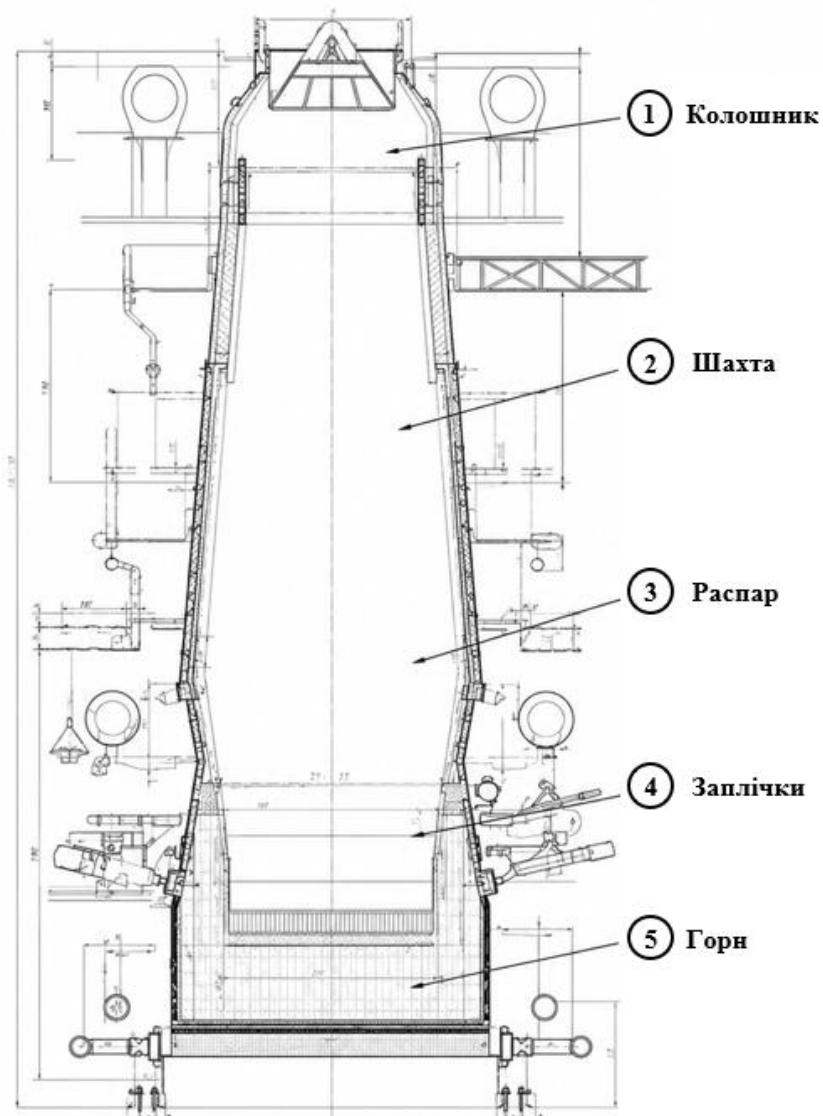


Рисунок 1.1 - Схема доменної печі з позначенням основних елементів

Конструктивно доменна піч складається з наступних основних елементів:

1. Колошник – верхня частина печі, призначена для прийому та рівномірного розподілу шихтових матеріалів (агломерат, кокс, флюси) за допомогою засипного апарату.

2. Шахта – середня конічна частина печі, де відбуваються основні фізико-хімічні процеси відновлення заліза з оксидів.

3. Распар – розширена частина шахти, де шихта починає розм'якшуватися та плавитися.

4. Заплічки – нижня конічна частина, що переходить у горн.

5. Горн – нижня циліндрична частина печі, де накопичуються рідкі продукти плавки (чугун та шлак) і звідки вони випускаються через льотки.

Технологічний процес роботи доменної печі є безперервним і складається з наступних основних етапів:

1. Підготовка шихти: агломерат, кокс та флюси дозуються та подаються на колошник.

2. Завантаження шихти: через засипний апарат (конусний або безконусний) шихта рівномірно розподіляється по перерізу печі.

3. Подача гарячого дуття: через фурми в нижню частину печі подається гаряче дуття (температура 1000 - 1200°C), збагачене киснем (до 25 - 30%) та пиловугільним паливом (ПУТ). Дуття забезпечує горіння коксу та підтримує високу температуру (до 2000°C) у фурменній зоні.

4. Відновлення заліза: у міру опускання шихти вона нагрівається зустрічним потоком газів, що піднімаються з фурменної зони. Відновлення заліза з оксидів відбувається в кілька стадій:

- $3\text{Fe}_2\text{O}_2 + \text{CO} \rightarrow 2\text{Fe}_3\text{O}_4 + \text{CO}_2$
- $3\text{Fe}_3\text{O}_4 + \text{CO} \rightarrow 3\text{FeO} + \text{CO}_2$
- $\text{FeO} + \text{CO} \rightarrow \text{Fe} + \text{CO}_2$
- $\text{FeO} + \text{C} \rightarrow \text{Fe} + \text{CO}$ (пряме відновлення)

5. Плавлення та накопичення: відновлене залізо розплавляється, поглинає вуглець (до 4 - 5%) та інші елементи, утворюючи рідкий чавун, який стікає в горн. Разом з чавуном утворюється шлак (суміш оксидів кремнію, кальцію, алюмінію та ін.).

6. Випуск продуктів плавки: через льотки в горні періодично (кожні 1 - 2 години) випускають чавун та шлак.

Основні технологічні параметри роботи ДП-1М наведено в таблиці 1.2.

Таблиця 1.2 - Основні технологічні параметри доменної печі ДП-1М ПРАТ «КАМЕТ-СТАЛЬ» [5; 6]

Параметр	Значення	Одиниця виміру
Температура гарячого дуття	1080 - 1150	°С
Температура чавуну	1450 - 1520	°С
Витрата дуття	2800 - 3200	м ³ /хв
Тиск газу на колошнику	1,8 - 2,5	атм
Кількість фурм	18	шт.
Діаметр горна	8,6	м
Витрата ПУТ	180 - 220	кг/т чавуну
Витрата коксу	380 - 420	кг/т чавуну
Середня продуктивність	2893	т/добу

Найважливішим елементом конструкції доменної печі з точки зору захисту від термічних навантажень є футерівка - внутрішнє вогнетривке облицювання, що виконує захисну функцію для металевого корпусу печі від високих температур (до 1500 - 2000°С) та агресивного хімічного середовища [7].

Сучасна футерівка доменної печі являє собою багат шарову конструкцію, що включає:

- Вогнетривкий шар – безпосередньо контактує з шихтою та газами; виготовляється з високоглиноземистих, карбід-кремнієвих або графітованих вогнетривів.

- Проміжний шар – теплоізоляційний матеріал для зменшення теплових втрат.

- Холодильники (cooling staves) – металеві (чавунні або мідні) елементи з вбудованими каналами для циркуляції води, призначені для відведення тепла від футерівки.

- Металевий кожух – зовнішня сталева оболонка печі.

У вітчизняній та світовій практиці особливу увагу приділяють системі охолодження футерівки, оскільки від її безвідмовної роботи залежить довговічність печі. Система охолодження на ПРАТ «КАМЕТ-СТАЛЬ» є водяною з повітряним компонентом і складається з:

- Зовнішнього водяного охолодження – холодильники, розташовані по висоті печі (від горна до колошника), з циркуляцією технічної води.

- Внутрішнього повітряного охолодження – подача повітря під надлишковим тиском між металевим кожухом та футерівкою.

- Аварійного охолодження – резервні насоси та автономні джерела живлення, що забезпечують мінімальну циркуляцію води при знеструмленні.

За нормальних (штатних) умов усі системи автоматизації працюють у безперервному режимі, забезпечуючи стабільність технологічного процесу. АСУ ТП здійснює:

- збір даних з понад 1000 датчиків (температура, тиск, витрата, рівень, склад газу);

- автоматичне регулювання подачі дуття, ПУТ та шихти;

- контроль теплового стану футерівки (термопари, розташовані на різних висотах та по периметру);

- сигналізацію та блокування при виході параметрів за допустимі межі.

1.1.3 Термошок футерівки доменної печі при екстремому знеструмленні: сутність, причини та наслідки

Термошок футерівки (англ. thermal shock) - це явище різкої зміни температури внутрішнього вогнетривкого облицювання печі, що призводить до виникнення значних термічних напружень у матеріалі, його деформації, утворення тріщин та подальшого руйнування. У контексті доменного виробництва термошок виникає при екстремому знеструмленні, коли порушується нормальний тепловий режим роботи печі, а охолодження футерівки стає нерівномірним і некерованим.

Фізична сутність процесу. Вогнетривкі матеріали, з яких виготовляється футерівка (високоглиноземисті, шамотні, карбід-кремнієві вогнетриви, графітовані блоки), мають певний коефіцієнт теплового розширення та теплопровідності. При рівномірному нагріванні або охолодженні матеріал розширюється або стискається відносно рівномірно по всьому об'єму, що не призводить до виникнення значних внутрішніх напружень.

Однак при різкому охолодженні (наприклад, при зупинці подачі гарячого дуття та порушенні режиму водяного охолодження) виникає нерівномірний розподіл температур по товщині футерівки:

- Поверхневий шар (що контактує з шихтою або газами) охолоджується швидше за рахунок теплообміну з холодними газами або припинення підігріву.
- Внутрішній шар (прилеглий до холодильників) залишається більш нагрітим, оскільки віддача тепла через холодильники ускладнюється при зниженні тиску води або її зупинці.

Цей температурний градієнт (різниця температур між поверхневим та внутрішнім шарами) породжує механічні напруження, оскільки холодніший поверхневий шар прагне стиснутися, а гарячіший внутрішній - розширитися. Якщо ці напруження перевищують межу міцності матеріалу, виникають

тріщини, розкриття яких призводить до подальшого проникнення газів та руйнування футерівки. Схематично цей процес зображено на рисунку 1.2.

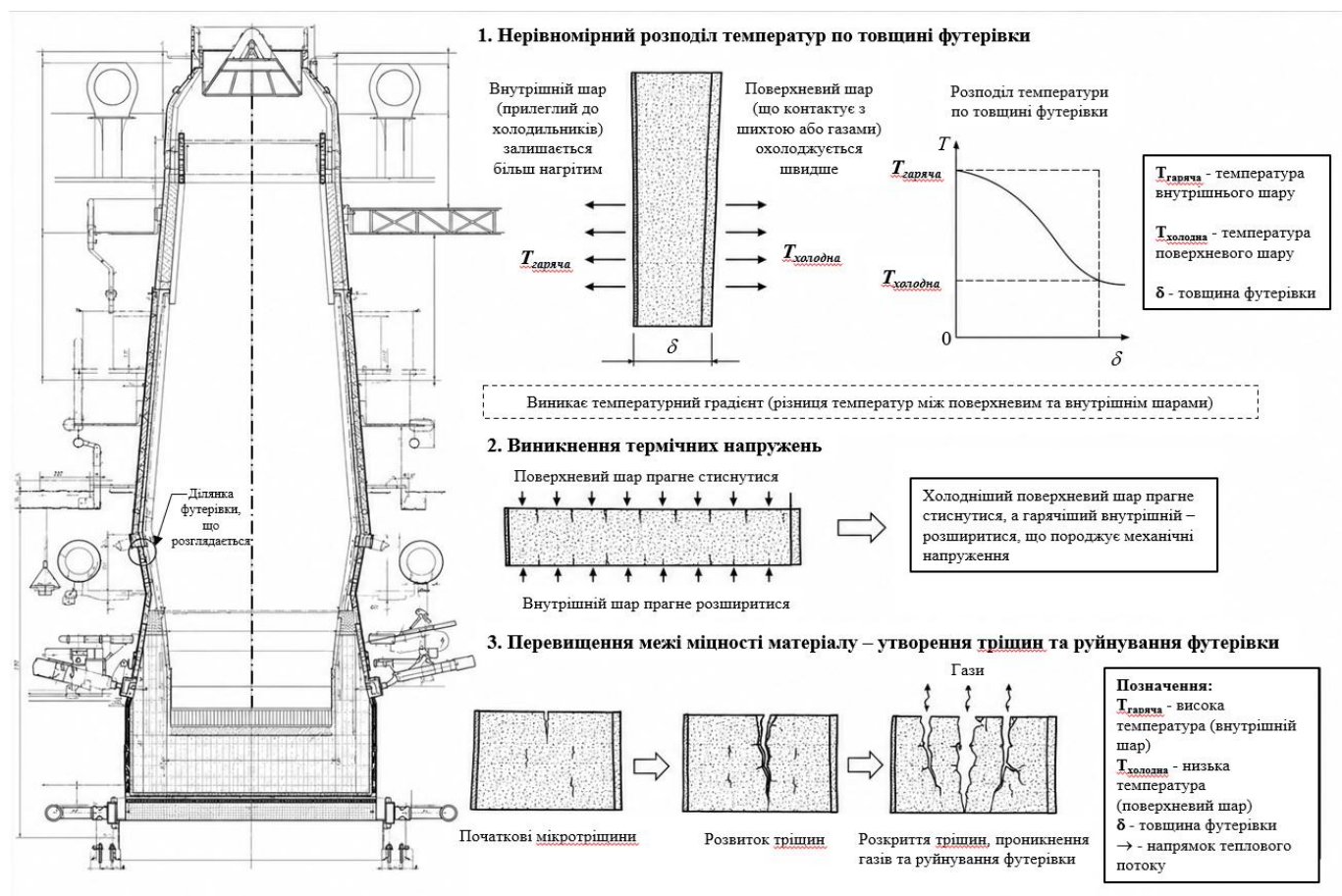


Рисунок 1.2 – Схема виникнення термічних напружень у футерівці при різкому охолодженні

Кількісно ступінь термічного удару характеризують критерієм термостійкості, який визначається за формулою:

$$R = (\sigma \cdot \lambda) / (\sigma \cdot E)$$

де:

- R – критерій термостійкості (напружений фактор);
- σ – межа міцності на розрив матеріалу, МПа;
- λ – коефіцієнт теплопровідності, Вт/(м·К);

- α – коефіцієнт лінійного теплового розширення, $1/^\circ\text{C}$;
- E – модуль пружності матеріалу, МПа.

Чим вище значення R , тим матеріал більш стійкий до термошоку. Для сучасних вогнетривів значення R коливаються в широкому діапазоні, проте жоден матеріал не є абсолютно терmostійким при різких перепадах температур понад $200 - 300^\circ\text{C}$ [8].

Причини виникнення термошоку при екстремому знеструмленні. Екстремне знеструмлення доменної печі може статися з різних причин:

- аварії в енергомережі (пошкодження ліній електропередач, трансформаторних підстанцій);
- пошкодження кабельних трас або розподільчих пристроїв;
- техногенні аварії (пожежі, вибухи, обвали);
- військові дії та цілеспрямовані удари по енергетичній інфраструктурі (особливо актуально для України в 2022 – 2026 роках) [2];
- збої в роботі резервних джерел живлення (дизель-генераторів, акумуляторних батарей).

При екстремому знеструмленні в доменній печі відбуваються наступні критичні зміни:

1. Припинення подачі гарячого дуття. Фурменна зона перестає отримувати кисень та паливо (кокс, ПУТ), горіння припиняється, температура в горні починає знижуватися. Цей процес відбувається нерівномірно - спочатку охолоджуються периферійні зони, потім - центральні.

2. Зупинка завантаження шихти. Колошник перестає приймати нові порції шихти, що призводить до зменшення теплового екрану (захисту від випромінювання) у верхній частині печі.

3. Порушення роботи системи охолодження. При знеструмленні зупиняються насоси, що подають воду до холодильників, або тиск води значно падає. Хоча система може мати резервні насоси та дизель-генератори, їх

перехід займає 45 хвилин – 2 години, що є критичним у перші 10 – 15 хвилин [5].

4. Нерівномірне охолодження конструктивних елементів. Найшвидше охолоджуються верхні частини печі (колошник, засипний апарат, верхня частина шахти), де немає активного тепловиділення. У той же час горн ще певний час залишається гарячим за рахунок акумульованого тепла рідкого чавуну та шлаку.

5. Зміна складу та руху газів. Припинення подачі дуття призводить до зменшення об'єму газів, що піднімаються вгору, та зміни їх температурного профілю. Це впливає на теплообмін у шахті та тепловий режим футерівки.

Схема розвитку аварійної ситуації при знеструмленні представлена на рисунку 1.3.

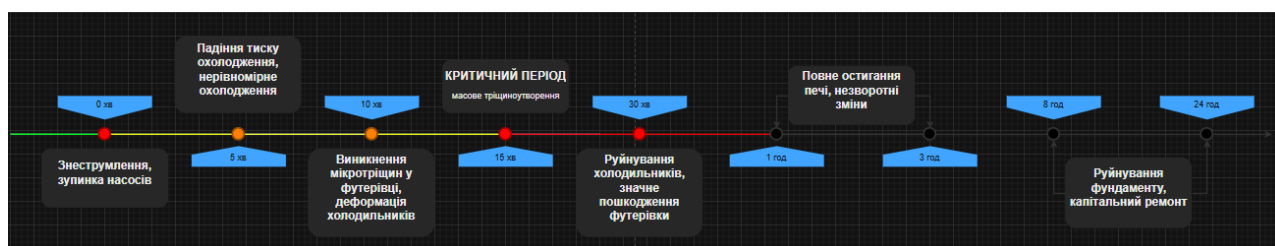


Рисунок 1.3 – Діаграма послідовності подій при екстремому знеструмленні доменної печі

Критичні елементи, вразливі до термошоку. Аналіз конструкції та умов роботи доменної печі дозволяє виділити найбільш вразливі до термошоку елементи:

1. Колошник та засипний апарат. Це верхня частина печі, яка найбільш інтенсивно охолоджується при знеструмленні через:

- відсутність активного тепловиділення (у колошнику немає горіння);
- підвищений теплообмін з навколишнім середовищем (тонкі металеві конструкції);

– вплив холодних газів, що піднімаються з нижніх зон, які вже охолодилися при зупинці дуття.

Температура на колошнику може знизитися на 300 - 500°C протягом перших 10 - 15 хвилин, що створює критичні термічні напруження в металоконструкціях засипного апарату та верхній частині футерівки [7].

2. Футерівка верхньої та середньої частини шахти. Ці зони зазнають найбільших температурних градієнтів через:

- швидке охолодження з боку холодних газів, що піднімаються;
- наявність холодильників, які при зниженні тиску води починають працювати нерівномірно;
- усадку шихти, що змінює тепловий контакт футерівки з матеріалом.

3. Холодильники (cooling staves). Як зазначається в дослідженнях [3], чавунні холодильники через свою погану термостійкість є особливо вразливими. При порушенні режиму охолодження:

- тепло, що акумульоване у футерівці, передається на холодильники;
- холодильники нагріваються нерівномірно, що призводить до їх деформації та тріщин;
- температура газу всередині печі є головним фактором, що впливає на перегрів холодильників [3].

Дослідження [3] показали, що коефіцієнт високотемпературної області (high-temperature region ratio) на поверхні холодильників зазвичай становить 30 - 60%, що свідчить про значну нерівномірність теплового навантаження.

4. Фурми та фурменна зона. Хоча горн найдовше зберігає тепло, фурми як найбільш тонкі металеві елементи зазнають швидкого охолодження при зупинці подачі дуття. В Україні на ПРАТ «КАМЕТ-СТАЛЬ» у січні 2026 року саме руйнування фурм через перегрів стало основною причиною зупинки печі після знеструмлення [2].

5. Металевий кожух печі. При нерівномірному охолодженні футерівки виникають локальні перегрівки окремих ділянок кожуха, що може призвести до його деформації та утворення тріщин.

Часова динаміка розвитку термошоку. На основі аналізу реальних аварійних ситуацій та результатів моделювання [2; 3; 9], можна виділити наступні критичні часові інтервали при екстремому знеструмленні:

- 0 - 5 хвилин: різке падіння тиску в системі охолодження, початок нерівномірного остигання колошника та верхньої частини шахти. Температурні градієнти досягають 50 - 100 °С/м.

- 5 - 10 хвилин: виникнення перших тріщин у футерівці верхньої зони, початок деформації холодильників. Ризик незворотних змін стає високим.

- 10 - 15 хвилин: критичний період - температурні градієнти перевищують 120 °С/м, що призводить до масового тріщиноутворення. У цей момент необхідне прийняття негайних рішень персоналом [5].

- 15 - 30 хвилин: при відсутності резервного живлення та ручного втручання відбувається руйнування холодильників та значна деформація футерівки.

- 1 - 3 години: повне остигання печі, що вимагає тривалого (тижні-місяці) простою для відновлення футерівки.

- 8 - 24 години: при тривалому простої без належного обслуговування відбувається руйнування фундаменту печі, що робить ремонт економічно недоцільним [10].

Таблиця 1.3 – Критичні часові інтервали та наслідки для футерівки

Час після знеструмлення	Температурний градієнт	Наслідки для футерівки	Необхідні дії
0 - 5 хв	50 - 100 °С/м	Початок нерівномірного охолодження	Моніторинг, активація резервних насосів
5 - 10 хв	80 - 120 °С/м	Виникнення мікротріщин, ризик деформації холодильників	Прогнозування розвитку ситуації
10 - 15 хв	> 120 °С/м	Критичний – масове тріщиноутворення, ризик руйнування	Негайні дії персоналу, аварійний випуск
15 - 30 хв	> 150 °С/м	Руйнування холодильників, деформація футерівки	Евакуація, аварійний протокол
> 1 год	> 200 °С/м	Повне остигання, вихід печі з ладу	Капітальний ремонт (тижні-місяці)

Таким чином, найкритичнішим періодом є перші 10 - 15 хвилин після знеструмлення, коли персонал повинен отримувати достовірну оперативну інформацію про температурний стан футерівки та чіткі рекомендації щодо дій. Відсутність такої інформаційної підтримки є основною причиною значних пошкоджень та довготривалих простоїв.

1.1.4 Глосарій предметної області

Для однозначного розуміння термінології, що використовується в роботі, наведено глосарій основних понять предметної області (таблиця 1.4).

Таблиця 1.4 – Глосарій основних термінів предметної області

Термін	Визначення	Джерело
Термошок футерівки	Різка зміна температури вогнетривкої кладки, що призводить до виникнення термічних напружень, тріщин та руйнування матеріалу	[7; 8]
Футерівка	Внутрішнє вогнетривке облицювання доменної печі, що захищає металевий корпус від високих температур та хімічного впливу	[7]
Колошник	Верхня частина доменної печі, призначена для прийому та розподілу шихтових матеріалів	[5]
Засипний апарат	Механізм для рівномірного розподілу шихти по перерізу колошника (конусний або безконусний)	[5]
Шахта	Середня конічна частина печі, де відбуваються основні відновлювальні процеси	[5]
Горн	Нижня частина доменної печі, де накопичуються рідкі продукти плавки (чугун та шлак)	[5]
Фурми	Пристрої для подачі гарячого дуття в нижню частину печі (фурменну зону)	[5]
Холодильники (cooling staves)	Металеві (чавунні або мідні) елементи з вбудованими каналами для циркуляції води, призначені для відведення тепла від футерівки	[3]

Продовження таблиці 1.4

Термін	Визначення	Джерело
АСУ ТП	Автоматизована система управління технологічним процесом – комплекс програмно-апаратних засобів для контролю та регулювання параметрів виробництва	[6]
ПУТ	Пиловугільне паливо – подрібнене вугілля, що вдувається в горн через фурми для заміни частини коксу	[5]
SCADA	Supervisory Control and Data Acquisition – система диспетчерського управління та збору даних (верхній рівень АСУ ТП)	[6]
Термічний градієнт	Різниця температур на одиницю довжини (наприклад, °С/м), що характеризує нерівномірність нагрівання/охолодження	[8]
Критерій термостійкості	Кількісна характеристика здатності матеріалу витримувати різкі перепади температур без руйнування	[8]
Edge computing	Обчислення на периферії (краю) мережі – виконання обробки даних безпосередньо біля джерела їх генерації (наприклад, біля печі)	[11]
Програмно-методичний комплекс (ПМК)	Взаємозалежна сукупність програмного, інформаційного та методичного забезпечень, необхідна для вирішення певного класу задач	[12]

1.2 Аналіз сучасних інформаційних технологій, технологій та засобів розробки програмного забезпечення

1.2.1 Аналіз поточних інформаційних систем та алгоритмів на підприємстві

На ПРАТ «КАМЕТ-СТАЛЬ» у доменному цеху функціонує багаторівнева автоматизована система управління технологічним процесом (АСУ ТП), впровадження якої забезпечує збір даних з численних датчиків температури, тиску, витрат і складу газів. Архітектура АСУ ТП має ієрархічну структуру (рисунок 1.4) і складається з наступних рівнів:

1. Польовий рівень: датчики та виконавчі механізми (термопари, датчики тиску, витратоміри, газоаналізатори, регулювальні клапани тощо). Кількість датчиків на ДП-1М перевищує 1000 одиниць, що забезпечує високу деталізацію контролю параметрів [6].

2. Контролерний рівень: програмовані логічні контролери (ПЛК) Siemens S7-400, які здійснюють збір даних з датчиків, первинну обробку, фільтрацію шумів та реалізацію алгоритмів автоматичного регулювання (PID-регулятори, логічні блоки, захисні блокування). Швидкодія контролерів забезпечує цикл опитування датчиків на рівні 100 - 500 мс [6].

3. Рівень SCADA/HMI: система диспетчерського управління WinCC (Siemens), яка забезпечує візуалізацію технологічних параметрів на АРМ операторів, архівацію даних (збереження в SQL-базах даних з періодом 1 - 10 секунд), тривожну сигналізацію та формування звітів. Оператори мають змогу спостерігати мнемосхеми печі, графіки зміни параметрів у реальному часі та історію змін за останні 24 - 72 години.



Рисунок 1.4 – Архітектура АСУ ТП доменного цеху ПРАТ «КАМЕТ-СТАЛЬ»

4. Рівень управління виробництвом: MES-системи (Manufacturing Execution System) для планування виробництва, обліку продукції, аналізу техніко-економічних показників.

При штатній експлуатації система працює переважно в автоматичному режимі. Регулювання температури гарячого дуття, витрати ПУТ, співвідношення «дутье–кокс» та інших ключових параметрів здійснюється автоматично без участі оператора. Людина виконує лише функції контролю, корекції завдань для регуляторів та прийняття рішень у нестандартних ситуаціях.

Однак, згідно з інформацією, зібраною під час переддипломної практики на дільниці, при екстремому знеструмленні система АСУ ТП має суттєві обмеження:

1. Часткове резервування живлення. Система АСУ ТП переходить на резервне живлення (дизель-генератори та акумуляторні батареї) протягом 45 хвилин – 2 годин, що є критичним у перші 10 - 15 хвилин після знеструмлення. У цей період більшість контрольно-вимірювальних приладів працюють у обмеженому режимі або повністю вимикаються [5].

2. Втрата оперативної візуалізації. При знеструмленні АРМ операторів можуть тимчасово втрачати живлення, що позбавляє персонал можливості спостерігати за зміною параметрів у реальному часі. Це особливо небезпечно в перші критичні хвилини.

3. Відсутність прогнозних алгоритмів. Існуюча АСУ ТП не має вбудованих алгоритмів прогнозування розвитку температурних полів футерівки при знеструмленні. Система лише фіксує поточні значення датчиків, але не моделює, як змінюватиметься ситуація в найближчі хвилини або десятки хвилин.

4. Ручний режим управління. Персонал змушений переводити обладнання в ручний режим та виконувати регламентні операції (очищення, заміна фурм, регулювання клапанів) без автоматизованих підказок чи рекомендацій.

Існуючий регламент дій персоналу при аварійному знеструмленні передбачає:

- контроль відсутності напруги на основному живленні;
- переведення критичного обладнання в ручний режим;
- візуальний огляд критичних елементів (колошник, засипний апарат, фурми);
- виконання регламентних ручних операцій згідно з інструкціями;
- очікування відновлення основного живлення або запуску резервного.

Цей підхід має ряд недоліків, які будуть детально проаналізовані в підрозділі 1.2.3.

1.2.2 Порівняльний аналіз сучасних методів моніторингу та прогнозування температурного стану

У сучасній металургійній практиці для захисту доменних печей від термошоку застосовуються різні підходи та методи. Їх порівняльний аналіз наведено в таблиці 1.5.

На ПРАТ «КАМЕТ-СТАЛЬ» наразі переважає перший метод - стаціонарний температурний моніторинг з використанням термопар, що є найбільш надійним, але недостатньо інформативним для прогнозування критичних ситуацій. Використання математичних моделей та систем штучного інтелекту обмежене через:

- відсутність спеціалізованого програмного забезпечення;
- недостатню кількість оцифрованих історичних даних для навчання моделей;
- нестачу кваліфікованих фахівців з аналізу даних у виробничих підрозділах.

Таблиця 1.5 – Порівняльний аналіз методів моніторингу та прогнозування температурного стану футерівки

Метод	Принцип роботи	Переваги	Недоліки	Застосування
Стаціонарний температурний моніторинг	Вимірювання температури за допомогою термоміпар, розташованих на різних рівнях футерівки	<ul style="list-style-type: none"> Висока точність вимірювань ($\pm 1-2^{\circ}\text{C}$) Простота реалізації Добре зарекомендував у практиці 	<ul style="list-style-type: none"> Обмежена кількість точок вимірювання Низька просторова роздільна здатність При знеструмленні більшість датчиків вимикається 	Широко застосовується на всіх підприємствах, у т.ч. на ПРАТ «КАМЕТ-СТАЛЬ»
Волоконно-оптичні датчики	Вимірювання температури за допомогою розподіленого волоконно-оптичного сенсора (DTS – Distributed Temperature Sensing) на основі ефекту Рамана або Бриллюена	<ul style="list-style-type: none"> Висока просторова роздільна здатність (до 1 м) Суцільний вимір по всій довжині волокна Стійкість до електромагнітних завад 	<ul style="list-style-type: none"> Висока вартість Складність монтажу Обмежена максимальна температура (до 300°C) 	Застосовується на передових підприємствах (Європа, Японія)
Математичне моделювання теплового балансу	Розв'язання рівнянь теплопереносу (Фур'є) з урахуванням внутрішніх джерел тепла, теплообміну з газами та шихтою	<ul style="list-style-type: none"> Може моделювати динаміку змін Не потребує додаткових датчиків Дозволяє прогнозувати 	<ul style="list-style-type: none"> Значна похибка в динамічних режимах Висока обчислювальна складність Необхідність калібрування на реальних даних 	Використовується для наукових досліджень та на деяких підприємствах
Штучний інтелект та машинне навчання	Навчання моделей (нейронні мережі, дерева рішень, SVR) на історичних даних датчиків для прогнозування температурних змін	<ul style="list-style-type: none"> Висока точність прогнозів на основі великих даних Автоматичне виявлення складних залежностей Адаптивність до змін умов 	<ul style="list-style-type: none"> Потрібна велика кількість якісних історичних даних "Чорний ящик" – складність інтерпретації Необхідність перенавчання при зміні умов 	Активно впроваджується на передових підприємствах (ArcelorMittal, Tata Steel)
Експертні системи та системи підтримки прийняття рішень	База знань, що містить правила та евристики для оцінки ситуації та формування рекомендацій	<ul style="list-style-type: none"> Прозорість рішень Легкість навчання персоналу Швидкість роботи 	<ul style="list-style-type: none"> Обмеженість знаннями експертів Важкість актуалізації бази знань Не враховує всі можливі ситуації 	Використовується на деяких підприємствах як допоміжний інструмент
Комп'ютерне зору та термографія	Аналіз теплових полів за допомогою тепловізорів та камер високої роздільності	<ul style="list-style-type: none"> Висока просторова роздільна здатність Візуальна наочність для оператора 	<ul style="list-style-type: none"> Залежність від якості оптики та умов огляду Обмежена робоча температура камер 	Застосовується для періодичного контролю, але не в реальному часі

Таким чином, основний метод інформаційної підтримки персоналу на сьогодні - регламентно-ручний підхід, який не використовує потенціалу сучасних інформаційних технологій для прогнозування та автоматизації прийняття рішень.

На ПРАТ «КАМЕТ-СТАЛЬ» наразі переважає перший метод - стаціонарний температурний моніторинг з використанням термопар, що є найбільш надійним, але недостатньо інформативним для прогнозування критичних ситуацій. Використання математичних моделей та систем штучного інтелекту обмежене через:

- відсутність спеціалізованого програмного забезпечення;
- недостатню кількість оцифрованих історичних даних для навчання моделей;
- нестачу кваліфікованих фахівців з аналізу даних у виробничих підрозділах.

Таким чином, основний метод інформаційної підтримки персоналу на сьогодні - регламентно-ручний підхід, який не використовує потенціалу сучасних інформаційних технологій для прогнозування та автоматизації прийняття рішень.

1.2.3 Переваги та недоліки існуючих підходів

Проведений аналіз дозволяє систематизувати переваги та недоліки поточної системи інформаційної підтримки на ПРАТ «КАМЕТ-СТАЛЬ».

Переваги:

1. Відпрацьований регламент дій персоналу. Протягом десятиліть експлуатації доменних печей вироблено чіткі алгоритми дій при аварійних

ситуаціях, які задокументовані в технологічних інструкціях. Цей регламент дозволяє уникнути хаосу та паніки серед персоналу.

2. Наявність резервного живлення для ключових контурів контролю. Частина датчиків та АРМ операторів підключені до дизель-генераторів та акумуляторних батарей, що дозволяє продовжувати моніторинг навіть при короткочасному знеструмленні [5].

3. Висока кваліфікація персоналу доменного цеху. Оператори, майстри та начальники змін мають значний досвід роботи, що дозволяє їм у багатьох випадках інтуїтивно оцінювати ситуацію та приймати правильні рішення навіть без автоматичних підказок.

4. Надійність обладнання. Основне обладнання (контролери, датчики, виконавчі механізми) має високу заводську надійність та відносно рідко виходить з ладу в штатних умовах.

5. Можливість ручного управління. У разі повного відмовлення автоматики персонал може перевести критичні механізми в ручний режим та керувати ними безпосередньо (засувки, клапани, засипний апарат тощо).

Недоліки:

1. Відсутність оперативного моніторингу температурного поля футерівки в реальному часі. Наявна система датчиків (термопари) забезпечує лише дискретний контроль у відносно невеликій кількості точок (не більше 20–30 по всій печі). Цього недостатньо для побудови повної температурної карти футерівки, особливо в динамічному аварійному режимі [3].

2. Відсутність алгоритмів прогнозування. Сучасні методи, такі як машинне навчання або математичне моделювання, не використовуються для прогнозування розвитку температурних полів та оцінки ризику термошоку.

3. Суб'єктивність прийняття рішень. Оскільки система не надає персоналу автоматичних рекомендацій, рішення приймаються на основі досвіду конкретного оператора або майстра. Це призводить до варіабельності дій та підвищує ризик помилок, особливо в умовах стресу та дефіциту часу.

4. Висока ймовірність пошкодження футерівки. Через нерівномірне охолодження в перші критичні хвилини футерівка зазнає значних термічних напружень, що призводить до утворення тріщин та скорочення міжремонтного періоду роботи печі [7; 9].

5. Тривалий час відновлення після аварії. Відновлення нормального режиму роботи печі після аварійної зупинки займає від 8 до 24 годин, а в разі значних пошкоджень футерівки – тижні або місяці.

6. Відсутність інтегрованого програмно-методичного комплексу. Існуючі програмні засоби (SCADA, MES) функціонують ізольовано один від одного та не створюють єдиного інформаційного простору для оперативного аналізу аварійних ситуацій.

7. Обмежена можливість автоматизованого навчання персоналу. Через відсутність систем моделювання та тренувальних симуляторів оператори не мають змоги регулярно відпрацьовувати дії при знеструмленні без ризику для реального обладнання.

1.2.4 Висновки щодо проблем та напрямків удосконалення

Проведений аналіз стану питання дозволяє зробити наступні висновки:

1. Проблема термошоку при знеструмленні є актуальною та складною. Фізична сутність процесу – нерівномірне охолодження футерівки, що призводить до виникнення термічних напружень та тріщин. Найбільш критичними елементами є колошник, засипний апарат, верхня частина шахти та холодильники.

2. Перші 10 - 15 хвилин після знеструмлення є вирішальними. Саме в цей період виникають критичні температурні градієнти, які визначають подальшу долю обладнання. Відсутність оперативної інформації та прогнозів у цей час є головною причиною значних пошкоджень.

3. Існуюча система інформаційної підтримки є недостатньою. Вона базується на регламентно-ручному підході, не використовує сучасних методів прогнозування (математичне моделювання, машинне навчання) та не забезпечує персоналу автоматизованих рекомендацій.

4. Сучасні інформаційні технології відкривають нові можливості. Застосування методів штучного інтелекту, IoT, edge computing, хмарних технологій та розподілених систем дозволяє створити ефективний програмно-методичний комплекс для інформаційної підтримки.

Для вирішення виявлених недоліків доцільно розробити програмно-методичний комплекс (ПМК), який забезпечуватиме:

- Безперервний моніторинг температурного стану футерівки з використанням усіх доступних датчиків (термопари, витратоміри води, датчики тиску газу) та можливістю інтеграції з додатковими сенсорами (волоконно-оптичними, тепловізорами).

- Прогнозування динаміки температурних полів на основі математичних моделей теплопереносу та/або методів машинного навчання, що дозволить оцінювати ризик термошоку за 5–10 хвилин до виникнення критичних змін.

- Автоматизовану генерацію рекомендацій для персоналу (диференційованих за трьома рівнями ризику – низький, середній, високий) та для систем автоматичного захисту (блокування, аварійне охолодження).

- Інтеграцію з існуючою АСУ ТП для використання наявних даних та мінімізації витрат на додаткове обладнання.

- Роботу в умовах часткового знеструмлення з використанням резервних джерел живлення та edge-обчислень (обробка даних безпосередньо біля печі).

2 КРИТИЧНИЙ АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

2.1 Обґрунтування вибору методів теоретичних та експериментальних досліджень, програмного забезпечення

Для досягнення поставленої мети - створення програмно-методичного комплексу інформаційної підтримки запобігання термошоку - у роботі використовується комплекс методів дослідження, які умовно можна поділити на три групи: теоретичні, експериментальні та інструментальні.

2.1.1 Теоретичні методи дослідження

Теоретичні методи спрямовані на аналіз фізичної сутності процесів, що відбуваються в доменній печі при екстремому знеструмленні, та формалізацію цих процесів у вигляді математичних моделей.

Метод математичного моделювання теплових процесів є основним у роботі. Він базується на розв'язанні рівняння теплопровідності Фур'є для нестационарного режиму:

$$\frac{\partial T}{\partial t} = a \cdot \nabla^2 T + \frac{q_v}{p \cdot c}$$

де:

- T - температура, °C;
- t - час, с;
- $a = \frac{\lambda}{p \cdot c}$ - коефіцієнт температуропровідності, м²/с;

- λ - коефіцієнт теплопровідності матеріалу, Вт/(м·К);
- ρ - густина матеріалу, кг/м³;
- c - питома теплоємність, Дж/(кг·К);
- q_v - об'ємна потужність внутрішніх джерел тепла, Вт/м³;
- ∇^2 - оператор Лапласа.

Для практичної реалізації в умовах обмеженої кількості датчиків використовується спрощена модель, яка описує зміну температури в характерних точках футерівки (колошник, середина шахти, горн) з урахуванням коефіцієнтів теплопереносу для різних зон печі [8; 15]. Такий підхід дозволяє зберегти достатню точність для оперативного прогнозування, значно зменшивши обчислювальну складність.

Метод системного аналізу застосовується для дослідження технологічних процесів доменного виробництва, виявлення критичних елементів конструкції печі, вразливих до термошоку, та визначення інформаційних потоків між підсистемами. Цей метод дозволяє розглядати доменну піч як складну організаційно-технічну систему з множиною взаємопов'язаних елементів [12].

Метод порівняльного аналізу використовується для оцінки існуючих систем моніторингу та прогнозування температурного стану футерівки як на ПРАТ «КАМЕТ-СТАЛЬ», так і на інших підприємствах світу. Це дозволяє виявити недоліки поточних підходів та обґрунтувати необхідність розробки нового програмно-методичного комплексу.

2.1.2 Експериментальні методи дослідження

Експериментальні методи в даній роботі мають обмежений характер через складність проведення натурних експериментів на діючому

промислового обладнання (ризик пошкодження печі, економічна недоцільність). Тому основними є:

Метод чисельного експерименту – проведення серії розрахунків на математичній моделі при різних значеннях вхідних параметрів (температура дуття, тиск охолодження, час знеструмлення тощо). Це дозволяє дослідити поведінку системи в широкому діапазоні умов без ризику для реального обладнання [16].

Метод аналізу історичних даних - використання архівних даних АСУ ТП (температури футерівки, тиск води, продуктивність печі) для калібрування математичної моделі та визначення порогових значень критеріїв термошоку. Дані надані ПРАТ «КАМЕТ-СТАЛЬ» за період 2023 - 2026 років.

Метод симуляційного моделювання - створення програмної симуляції роботи датчиків та аварійних ситуацій для тестування алгоритмів моніторингу та прогнозування без підключення до реальної АСУ ТП.

2.1.3 Інструментальні методи та програмне забезпечення

Для реалізації поставлених завдань обрано наступний технологічний стек (обґрунтування вибору наведено в таблиці 2.1):

Вибір Python та FastAPI зумовлений також необхідністю швидкої розробки прототипу з подальшою можливістю масштабування. У разі потреби продуктивність можна підвищити за рахунок переходу на асинхронну обробку запитів (FastAPI підтримує `async/await`) або використання більш продуктивних баз даних (PostgreSQL, TimescaleDB для часових рядів).

Таблиця 2.1 – Обґрунтування вибору технологічного стеку для розробки ПМК

Компонент	Обране рішення	Обґрунтування вибору
Мова програмування	Python 3.10+	Висока продуктивність розробки, велика кількість бібліотек для наукових обчислень (NumPy, SciPy), простота інтеграції з Power BI через REST API, підтримка асинхронного програмування, активна спільнота [17]
Веб-фреймворк	FastAPI	Висока швидкодія (асинхронність), автоматична генерація документації OpenAPI/Swagger, вбудована валідація даних (Pydantic), простота створення REST API, легка інтеграція з SQLite [18]
База даних	SQLite	Легковага, не потребує окремого сервера, вбудована в Python, достатня продуктивність для прототипу (до 10^5 записів), простота резервного копіювання, підтримка SQL [19]
Інструмент візуалізації	Power BI Desktop	Безкоштовна версія, потужні можливості візуалізації, інтерактивні дашборди, умовне форматування, інтеграція з Python через REST API, підтримка динамічного оновлення, наочність для персоналу [20]
Протокол зв'язку з АСУ ТП	OPC UA	Промисловий стандарт для обміну даними між різними системами, підтримується більшістю сучасних ПЛК, захищений, масштабований [21]
Система контролю версій	Git + GitHub	Стандарт для командної розробки, збереження історії змін, можливість публікації коду в репозиторії університету [22]

2.2 Математична модель об'єкта дослідження

Математична модель теплового стану футерівки доменної печі при екстремому знеструмленні є основою для прогнозування термошоку та оцінки ризиків. Модель базується на рівнянні теплопровідності, яке розв'язується для характерних зон печі з урахуванням змінних теплових потоків.

2.2.1 Загальна постановка задачі

Розглядається тривимірна задача нестационарної теплопровідності для футерівки доменної печі, яка складається з трьох основних зон: колошник (верхня частина), шахта (середня частина), горн (нижня частина). Для кожної зони задаються:

- початкові умови: $T(x,y,z,0) = T_0(x,y,z)$ - розподіл температур у початковий момент часу (до знеструмлення);
- граничні умови: теплообмін з газами всередині печі (конвекція, випромінювання) та з охолоджувальною водою ззовні;
- внутрішні джерела тепла (хімічні реакції, горіння коксу, тепловиділення при відновленні заліза).

Для практичної реалізації в умовах обмеженої кількості датчиків використовується спрощена модель, що описує зміну температури в характерних точках футерівки.

2.2.2 Ключові параметри моделі

На основі аналізу технологічного процесу та літературних даних [3; 5; 8] виділено наступні ключові параметри моделі:

1. Коефіцієнти теплопереносу для різних зон печі:

Коефіцієнт теплопереносу (α) визначає швидкість теплообміну між футерівкою та навколишнім середовищем. На основі емпіричних даних для доменних печей встановлено наступні значення [5; 8]:

- $\alpha = 0.95$ - для верхньої зони (колошник) - висока інтенсивність теплообміну через тонку футерівку та вплив холодних газів;
- $\alpha = 0.85$ - для середньої зони (шахта) - середня інтенсивність, часткова ізоляція шихтою;
- $\alpha = 0.75$ - для нижньої зони (горн) - низька інтенсивність, потужний тепловий захист рідкими продуктами плавки.

Ці коефіцієнти є емпіричними, отриманими шляхом обробки багаторічних даних вимірювань на доменних печах різних типів і згодом скоригованими для умов ДП-1М ПРАТ «КАМЕТ-СТАЛЬ» [5].

2. Коефіцієнт охолодження (*cooling_factor*):

Визначається відношенням поточного тиску води в системі охолодження до номінального. При штатному режимі *cooling_factor* = 1. При знеструмленні тиск падає, що призводить до зменшення коефіцієнта охолодження:

$$coolingfactor = \frac{P_{water}}{P_{nominal}}$$

де P_{water} – поточний тиск води, бар; $P_{nominal}$ - номінальний тиск води (для ДП-1М – 4,5 бар).

3. Часові параметри:

- $\Delta t = 15$ хвилин – горизонт прогнозування (критичний період);
- $\delta t = 1 - 5$ секунд – крок дискретизації вимірювань;
- t = час, що минув після знеструмлення.

2.2.3 Рівняння прогнозування температури

На основі наведених параметрів розроблено рівняння прогнозування температури для кожної зони печі:

Загальний вигляд:

$$T_{pred}(t + \Delta t) = T_{current} - \alpha \cdot coolingfactor \cdot \frac{\Delta t}{15} \cdot \Delta T_{base}$$

де:

- T_{pred} - прогнозована температура через інтервал часу Δt , °С;
- $T_{current}$ - поточна температура, °С;
- α - коефіцієнт теплопереносу для відповідної зони (0,95 / 0,85 / 0,75);
- $cooling_factor$ - коефіцієнт охолодження (0,3 - 1,0);
- Δt - горизонт прогнозування, хвилини (1 - 60);
- ΔT_{base} - базова швидкість охолодження при знеструмленні, °С/15 хв (для ДП-1М – 150 - 200°С для колошника, 50 - 80°С для горна).

Для верхньої зони (колошник, засипний апарат):

$$T_{koloshnik}(t + \Delta t) = T_{koloshnik}(t) - \alpha_{kol} \cdot coolingfactor \cdot 0.1 \cdot \frac{\Delta t}{15}$$

де $\alpha_{kol} = 0,95$ – коефіцієнт теплопереносу для колошника.

Для середньої зони (шахта):

$$T_{mid}(t + \Delta t) = T_{mid}(t) - \alpha_{mid} \cdot coolingfactor \cdot 0.08 \cdot \frac{\Delta t}{15}$$

де $\alpha_{mid} = 0,85$

Для нижньої зони (горн, фурми):

$$T_{bottom}(t + \Delta t) = T_{bottom}(t) - \alpha_{bottom} \cdot coolingfactor \cdot 0.05 \cdot \frac{\Delta t}{15}$$

де $\alpha_{bottom} = 0,75$

2.2.4 Розрахунок температурних градієнтів

Температурний градієнт є ключовим показником, що характеризує нерівномірність охолодження футерівки та ризик термошоку. Вертикальний градієнт розраховується як різниця температур між нижньою та верхньою зонами, поділена на висоту печі:

$$G_{vertical} = \frac{T_{bottom} - T_{koloshnik}}{H}$$

де H – висота печі (для ДП-1М - 72 м).

Часовий градієнт (швидкість охолодження) розраховується як зміна температури за одиницю часу:

$$G_{time} = \frac{\Delta T}{\Delta t}$$

де ΔT – зміна температури за інтервал часу Δt .

Критичні порогові значення градієнтів (згідно з [3; 8; 9]):

- Вертикальний градієнт: $> 120^{\circ}\text{C}/\text{м}$ – високий ризик термошоку;
- Часовий градієнт: $> 50^{\circ}\text{C}/\text{хв}$ – критичне охолодження колошника;
- Максимальна різниця температур по висоті: $> 300^{\circ}\text{C}$ – небезпечна нерівномірність.

2.2.5 Критерії термошоку

На основі проведеного аналізу [3; 5; 7; 8] визначено наступні критерії термошоку, які використовуються в моделі:

- Температура колошника $< 500^{\circ}\text{C}$ – критичне охолодження верхньої частини печі, висока ймовірність термічних напружень у засипному апараті.
- Тиск охолоджувальної води $< 3,0$ бар – недостатня інтенсивність охолодження, порушення тепловідведення від футерівки.
- Вертикальний градієнт $> 120^{\circ}\text{C}/\text{м}$ – критична нерівномірність температур по висоті печі.
- Швидкість охолодження колошника $> 50^{\circ}\text{C}/\text{хв}$ – небезпечний темп охолодження металевих конструкцій.
- Різниця температур між суміжними датчиками $> 100^{\circ}\text{C}$ – локальна нерівномірність охолодження.
- Час знеструмлення > 15 хвилин без резервного живлення – перевищення допустимого часу автономної роботи.

Ці критерії інтегровані в алгоритм оцінки ризику, який детально описано в підрозділі 2.3.

2.3 Розробка методики дослідження

Методика дослідження є системою правил, прийомів та операцій, які використовуються для оцінки ризику термошоку футерівки доменної печі при екстремому знеструмленні. Вона включає визначення факторів впливу, методику збору та обробки даних, алгоритм оцінки ризику та інтерпретацію результатів.

2.3.1 Фактори, що впливають на термошок

На основі аналізу технологічного процесу, літературних даних [3; 5; 7; 8] та експертних оцінок фахівців доменного цеху ПРАТ «КАМЕТ-СТАЛЬ» визначено шість ключових факторів, які мають найбільший вплив на виникнення термошоку при екстремому знеструмленні. Кожному фактору присвоєно ваговий коефіцієнт (у балах), що відображає його відносну важливість (таблиця 2.2).

Таблиця 2.2 – Фактори впливу на ризик термошоку та їх вагові коефіцієнти

№	Фактор	Вага, бали	Обґрунтування
1	Статус живлення	35	Відмова живлення є першопричиною аварійної ситуації. Припинення подачі дуття та зупинка насосів охолодження створюють умови для нерівномірного охолодження футерівки. Фактор має найвищу вагу, оскільки його наявність автоматично переводить систему в аварійний режим [5]

Продовження таблиці 2.2

№	Фактор	Вага, бали	Обґрунтування
2	Тиск охолоджувальної води	25	Тиск води визначає інтенсивність тепловідведення від футерівки. При зниженні тиску нижче критичного рівня (3,0 бар) порушується нормальний режим охолодження, що призводить до перегріву холодильників та виникнення термічних напружень [3]
3	Температура колошника	20	Колошник є найбільш вразливою зоною печі через швидке охолодження при знеструмленні. Падіння температури нижче 500°C створює критичні термічні напруження в металоконструкціях засипного апарату та верхній частині футерівки [7]
4	Вертикальний градієнт температури	15	Різниця температур між верхньою та нижньою зонами печі характеризує нерівномірність теплового стану. Високий градієнт (>400°C/м) свідчить про значну нерівномірність охолодження, що підвищує ризик термічних напружень [8]
5	Прогнозоване падіння температури	10	Прогнозована швидкість охолодження колошника на найближчі 30 хвилин дозволяє оцінити динаміку розвитку аварійної ситуації. Падіння понад 30°C за 30 хвилин є критичним [9]
6	Час після відмови живлення	8	Тривалість перебування печі в аварійному режимі без резервного живлення. Чим довше триває відмова, тим більша ймовірність незворотних змін у футерівці. Критичний поріг – 20 хвилин [2; 5]

Загальна максимальна сума балів становить 100, що дозволяє оцінювати ризик у відсотковому еквіваленті. Розподіл ваг виконано методом експертних оцінок із залученням трьох фахівців доменного цеху (начальник зміни, головний інженер, начальник цеху) та узгоджено з літературними даними [5; 8].

2.3.2 Методика оцінки ризику (бальна система)

Оцінка ризику виконується за бальною системою, яка реалізована у функції *calculate_risk()* програмного комплексу (підрозділ 3.3). Алгоритм оцінки включає наступні етапи:

Етап 1. Збір поточних даних: отримання значень температури в трьох зонах печі (колошник, середина, низ), тиску охолоджувальної води, статусу живлення, часу після відмови та розрахунок вертикального градієнта.

Етап 2. Прогнозування температур: розрахунок прогнозних значень температури колошника на 30 хвилин уперед з використанням математичної моделі (підрозділ 2.2).

Етап 3. Розрахунок балів за кожним фактором: згідно з ваговими коефіцієнтами (таблиця 2.3). Для кожного фактора визначено порогові значення, при досягненні яких нараховуються бали:

Таблиця 2.3 – Порогові значення факторів та відповідні бали

Фактор	Умова	Бали
Статус живлення	emergency	35
Тиск охолодження	< 2.5 бар	30
Тиск охолодження	< 3.0 бар	20
Температура колошника	< 500°C	20
Температура колошника	< 550°C	10
Вертикальний градієнт	> 500°C/м	15
Вертикальний градієнт	> 400°C/м	8
Прогнозоване падіння	> 30°C за 30 хв	20
Прогнозоване падіння	> 15°C за 30 хв	10
Час після відмови	> 20 хв	15
Час після відмови	> 10 хв	8

Етап 4. Підсумовування балів: розрахунок загального балу ризику як суми балів за всіма факторами, що спрацювали. Максимальне значення – 100 балів.

Етап 5. Визначення рівня ризику: на основі сумарного балу за шкалою (таблиця 2.4).

Таблиця 2.4 – Шкала рівнів ризику термошоку

Сумарний бал	Рівень ризику	Пріоритет	Характеристика
0 – 19	Низький	D	Штатний режим. Усі параметри в нормі. Ризик термошоку мінімальний.
20 – 39	Помірний	C	Незначні відхилення параметрів. Рекомендується посилений моніторинг.
40 – 59	Середній	B	Помітні відхилення. Рекомендується зниження навантаження та перевірка системи охолодження.
60 – 79	Високий	A	Значні відхилення, аварійна ситуація. Необхідні негайні дії персоналу.
80 – 100	Критичний	A+	Критичний стан. Ризик незворотного руйнування футерівки. Негайний випуск чавуну та аварійні заходи.

Етап 6. Формування рекомендацій: на основі рівня ризику генерується текстова рекомендація для персоналу:

- Низький ризик: "Штатний режим. Продовжувати моніторинг."
- Помірний ризик: "Посилений моніторинг. Перевірити градієнти температури."
- Середній ризик: "Підвищена увага! Знизити навантаження. Перевірити систему охолодження."
- Високий ризик: "Аварійна ситуація! Збільшити тиск охолодження. Підготувати резервне живлення."

– Критичний ризик: "Негайний випуск чавуну. Вимкнути нагрів, максимальне охолодження."

Етап 7. Автоматичне оповіщення: при досягненні високого або критичного рівня ризику система автоматично генерує тривожне повідомлення (alert) з типом *THERMAL_SHOCK_RISK* та відповідним пріоритетом, яке зберігається в базі даних та може бути відображене на дашборді оператора.

2.3.3 Методика обробки результатів дослідження

Обробка результатів дослідження включає наступні етапи:

1. Збір даних. Дані збираються з використанням симуляції датчиків (функція `generate_sensor_data()`), яка генерує значення на основі обраного сценарію роботи печі. Для реального впровадження передбачено інтеграцію з існуючою АСУ ТП через протокол OPC UA, що дозволить отримувати дані безпосередньо з виробничих датчиків.

2. Попередня обробка. Оскільки дані з датчиків можуть містити шуми (випадкові коливання, перешкоди), у симуляції використовується генерація випадкових відхилень у межах фізіологічних норм. Для реальних даних передбачено використання фільтрації ковзним середнім (moving average) для згладжування викидів.

3. Збереження даних. Усі дані зберігаються у реляційній базі даних SQLite у трьох таблицях:

- `sensor_data` – історія показів датчиків (температури, тиск, статус живлення, градієнти);
- `predictions` – історія прогнозів температур та оцінок ризику;
- `alerts` – історія тривожних повідомлень.

Така структура дозволяє виконувати подальший аналіз історичних даних, навчати моделі машинного навчання, а також відстежувати ефективність роботи системи в часі.

4. Візуалізація результатів. Дані передаються до Power BI через REST API (ендпоінт */data-for-powerbi*), де вони візуалізуються у вигляді інтерактивних дашбордів. Опис візуалізацій наведено в підрозділі 3.4.

5. Інтерпретація результатів. На основі отриманих даних та оцінки ризику формуються висновки щодо:

- поточного теплового стану футерівки;
- прогнозованої динаміки змін у найближчі 30 хвилин;
- необхідності втручання персоналу;
- рекомендованих дій.

2.3.4 План чисельного експерименту

Для перевірки адекватності математичної моделі та алгоритмів оцінки ризику заплановано проведення серії чисельних експериментів з використанням розробленого програмного комплексу. План експерименту включає наступні сценарії (таблиця 2.5).

Таблиця 2.5 – План чисельного експерименту

№	Фактор	Вага, бали	Обґрунтування	№
1	Штатна робота	Параметри в межах норми, живлення стабільне	Низький ризик (0 - 19 балів), стабільні температури	100
2	Коливання живлення	Періодичні відхилення напруги, випадкові зміни параметрів	Помірний або середній ризик (20 - 59 балів), коливання температур	100

Продовження таблиці 2.5

№	Фактор	Вага, бали	Обґрунтування	№
3	Повне знеструмлення (короткочасне)	Відключення живлення на 5 - 10 хвилин, подальше відновлення	Високий ризик (60 - 79 балів), швидке охолодження колошника	50
4	Повне знеструмлення (тривале)	Відключення живлення на 15 - 30 хвилин без резерву	Критичний ризик (80 - 100 балів), небезпечні градієнти	50
5	Відновлення після аварії	Відновлення живлення після 30 хвилин аварійного режиму	Поступове повернення до норми, зниження ризику	50

Для кожного сценарію виконується серія ітерацій тривалістю 60 хвилин модельного часу з кроком оновлення даних 1 секунда. Результати експериментів (температури, градієнти, рівні ризику, рекомендації) зберігаються в базі даних для подальшого аналізу.

Критерії оцінки результатів експерименту:

- Адекватність моделі – порівняння прогнозованих температур з фактичними даними (середньоквадратична похибка $< 10^{\circ}\text{C}$);
- Чутливість системи – правильне визначення рівня ризику у відповідь на зміну вхідних параметрів;
- Швидкодія – час обробки одного циклу вимірювань та прогнозування не перевищує 1 секунди;
- Надійність – відсутність помилкових тривог та коректне генерування рекомендацій.

Результати чисельних експериментів будуть використані для калібрування моделі, уточнення порогових значень та підтвердження ефективності розробленого програмно-методичного комплексу.

2.4 Технічне завдання на створення програмно-методичного комплексу

Технічне завдання (ТЗ) на розробку програмно-методичного комплексу для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні є основним документом, що визначає вимоги до системи, її функціональність, склад та характеристики.

Повний текст ТЗ наведено в Додатку Б до пояснювальної записки. Нижче наведено короткий зміст основних розділів ТЗ:

1. Загальні відомості

- Найменування системи: «Програмно-методичний комплекс для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні».
- Замовник: ПРАТ «КАМЕТ-СТАЛЬ» (доменний цех).
- Призначення: автоматизований моніторинг температурного стану футерівки, прогнозування розвитку термошоку та інформаційна підтримка прийняття рішень персоналом.

2. Мета та цілі створення системи

- Зменшення ймовірності виникнення термошоку футерівки на 40–60%.
- Скорочення часу прийняття рішень персоналом у перші критичні 10–15 хвилин.
- Зменшення тривалості простою печі після аварійного знеструмлення на 20–40%.

3. Функціональні модулі системи

- Модуль збору даних (інтеграція з АСУ ТП, ОРС UA, симуляція датчиків).
- Модуль реального часу моніторингу (візуалізація температурних полів).

- Модуль прогнозування термошоку (математична модель, чисельні методи).

- Модуль підтримки прийняття рішень (5 рівнів ризику, рекомендації).

- Модуль оповіщення та інтерфейсу (дашборди Power BI, тривоги).

- Модуль логування та архівації (збереження даних у SQLite).

4. Вимоги до апаратного забезпечення

- Сервер: процесор Intel Xeon 4+ ядра, RAM 16+ ГБ, SSD 256+ ГБ.

- Резервне живлення: ДГУ (дизель-генераторна установка) з часом автономної роботи не менше 4 годин.

- Edge-пристрої: промислові міні-ПК або PLC з підтримкою OPC UA.

5. Вимоги до програмного забезпечення

- Backend: Python 3.10+, FastAPI, SQLite.

- Frontend: Power BI Desktop (безкоштовна версія).

- Система контролю версій: Git + GitHub.

- Протоколи зв'язку: OPC UA, REST API.

6. Вимоги до інтерфейсу

- Дашборди Power BI з інтерактивними графіками, тепловими картами, індикаторами ризику.

- Кольорове кодування рівнів ризику (зелений – низький, жовтий – помірний, оранжевий – середній, червоний – високий/критичний).

- Відображення рекомендацій для персоналу текстовим повідомленням.

- Автоматичні тривоги при високому та критичному ризику.

7. Вимоги до надійності та безпеки

- Резервування критичних модулів (дублювання серверів, БД).

- Захист від несанкціонованого доступу (автентифікація, розмежування прав).

- Регулярне резервне копіювання даних (щогодинне).

- Час відновлення після збою: не більше 5 хвилин.

8. Порядок контролю та приймання

- Тестування в симульованому режимі (3 сценарії).
- Дослідна експлуатація на ДП-1М протягом 30 днів.
- Оцінка ефективності за критеріями (точність прогнозу, швидкодія, надійність).

Повний текст ТЗ з детальним описом кожного розділу наведено в Додатку Б.

3. РОЗРОБКА ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ ТА ЙОГО РЕАЛІЗАЦІЯ

3.1 Розробка логічної моделі ПМК

Логічна модель програмно-методичного комплексу (ПМК) є абстрактним представленням системи, яке описує її функціональність, структуру та поведінку без прив'язки до конкретної програмної реалізації чи апаратного забезпечення. Для формалізації логічної моделі використовується уніфікована мова моделювання UML (Unified Modeling Language) версії 2.5 [23], яка є стандартом де-факто для об'єктно-орієнтованого проектування програмних систем.

У межах даної роботи розроблено чотири ключові діаграми UML, які відображають різні аспекти системи:

1. Діаграма прецедентів використання (Use Case Diagram) – описує функціональні вимоги до системи з точки зору користувачів та зовнішніх систем.
2. Діаграма діяльності (Activity Diagram) – відображає логіку та послідовність виконання технологічних процесів в аварійній ситуації.
3. Діаграма класів (Class Diagram) – описує статичну структуру системи, основні класи та зв'язки між ними.
4. Діаграма розгортання (Deployment Diagram) – показує фізичне розміщення компонентів системи на апаратному забезпеченні.

3.1.1 Діаграма прецедентів використання

Діаграма прецедентів використання (рисунок 3.1) відображає функціональні вимоги до ПМК та визначає, які дії можуть виконувати різні категорії користувачів та зовнішні системи.

Актори системи:

1. Оператор доменної печі – основний користувач системи, який безпосередньо керує технологічним процесом та приймає рішення на основі даних ПМК.
2. Майстер зміни – старший персонал, який має розширені права доступу, може змінювати сценарії роботи та підтверджувати тривоги.
3. Система АСУ ТП – зовнішня система, яка надає дані з датчиків та може отримувати керуючі команди від ПМК.
4. Система резервного живлення – зовнішня система, яка забезпечує електропостачання критичних компонентів при знеструмленні.

Основні прецеденти використання:

- UC-1: Перегляд поточного стану печі – оператор отримує актуальну інформацію про температури, тиск охолодження, статус живлення та градієнти.
- UC-2: Перегляд прогнозу температур – система відображає прогнозовані значення температур на 30 хвилин вперед для кожної зони печі.
- UC-3: Оцінка ризику термошоку – система автоматично обчислює бальну оцінку ризику на основі поточних даних та визначає рівень ризику.
- UC-4: Отримання рекомендацій – система генерує текстові рекомендації для персоналу на основі поточного рівня ризику.
- UC-5: Перегляд тривог – оператор переглядає список активних та історичних тривог.

- UC-6: Підтвердження тривоги – майстер підтверджує отримання тривоги, що знімає її з активного списку.
- UC-7: Зміна сценарію роботи – майстер перемикає сценарій роботи печі (штатний, коливання, аварія) для тестування системи.
- UC-8: Перегляд історії даних – оператор переглядає історичні дані датчиків для аналізу трендів.
- UC-9: Генерація даних (симуляція) – система автоматично генерує тестові дані при роботі в симульованому режимі.
- UC-10: Архівація даних – система автоматично зберігає всі вимірювання та прогнози в базі даних.

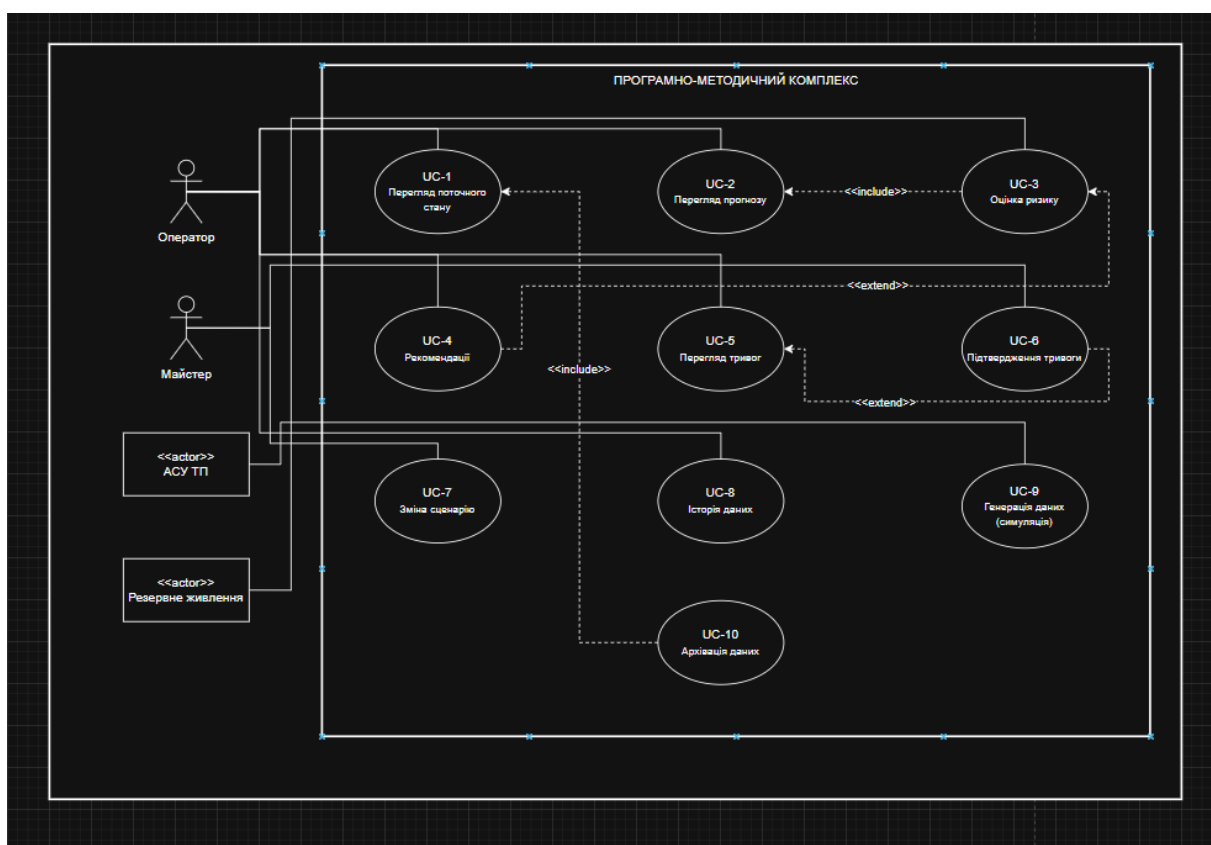


Рисунок 3.1 – Діаграма прецедентів використання ПМК

Легенда до діаграми прецедентів використання ПМК:

1. Суцільна лінія: Асоціація (актор використовує прецедент)

2. Пунктирна зі стрілкою:

- <<include>> (один прецедент завжди включає інший)
- <<extend>> (один прецедент може розширювати інший)

Таблиця 3.1 містить детальний опис кожного прецеденту з вказівкою пріоритету, частоти використання та умов успішного виконання.

Таблиця 3.1 – Специфікація прецедентів використання ПМК

ID	Назва	Актори	Пріоритет	Частота	Успішний результат
UC-1	Перегляд поточного стану	Оператор	Високий	Постійно	Відображення актуальних даних з датчиків
UC-2	Перегляд прогнозу	Оператор	Високий	Кожні 5 - 15 хв	Відображення прогнозованих температур
UC-3	Оцінка ризику	Система	Критичний	Постійно	Автоматичний розрахунок балу ризику
UC-4	Отримання рекомендації	Оператор	Високий	При зміні ризику	Генерація текстової рекомендації
UC-5	Перегляд тривог	Оператор	Середній	При тривозі	Відображення списку активних тривог
UC-6	Підтвердження тривоги	Майстер	Середній	При тривозі	Зняття тривоги з активного списку
UC-7	Зміна сценарію	Майстер	Низький	За потреби	Перемикання режиму роботи системи
UC-8	Перегляд історії	Оператор	Низький	Періодично	Відображення історичних даних
UC-9	Генерація даних	Система	Високий	Постійно	Створення симульованих вимірювань
UC-10	Архівація даних	Система	Високий	Постійно	Збереження даних у БД

3.1.2 Діаграма діяльності

Діаграма діяльності (рисунок 3.2) відображає логіку обробки аварійної ситуації при екстремому знеструмленні. Вона описує потік управління від моменту виникнення аварійної ситуації до прийняття рішення персоналом та виконання необхідних дій.

Основні етапи процесу:

1. Моніторинг параметрів: система безперервно збирає дані з датчиків (температура, тиск, статус живлення).
2. Виявлення знеструмлення: система фіксує перехід статусу живлення в режим emergency.
3. Збір аварійних даних: система продовжує збирати дані з доступних датчиків (резервне живлення).
4. Розрахунок прогнозу: система виконує прогнозування температур на 30 хвилин з використанням математичної моделі.
5. Оцінка ризику: система обчислює бальну оцінку ризику за шістьма факторами.
6. Визначення рівня ризику: система класифікує ситуацію за рівнями (Низький → Критичний).
7. Формування рекомендацій: система генерує текстове повідомлення для персоналу.
8. Оповіщення персоналу: система відображає рекомендацію на дашборді та генерує тривогу (при високому/критичному ризику).
9. Прийняття рішення: оператор або майстер отримує рекомендацію та приймає рішення про дії.
10. Виконання дій: персонал виконує рекомендовані дії (посилений моніторинг, зниження навантаження, аварійний випуск).
11. Підтвердження тривоги: майстер підтверджує отримання тривоги.

12. Повернення до моніторингу: система продовжує моніторинг параметрів.

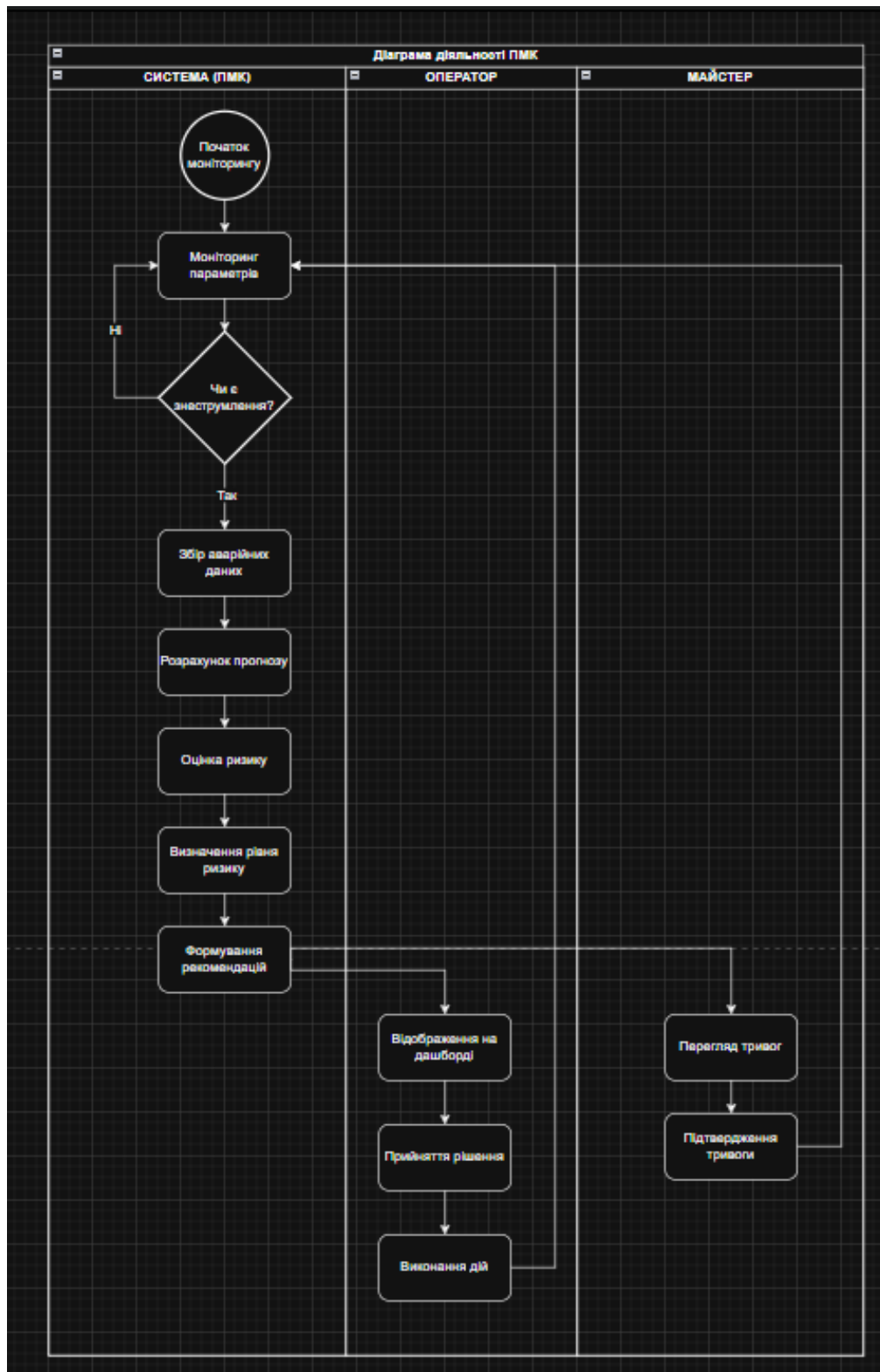


Рисунок 3.2 – Діаграма діяльності ПМК

3.1.3 Діаграма класів

Діаграма класів (рисунок 3.3) описує статичну структуру програмного забезпечення ПМК, показуючи основні класи, їх атрибути, методи та зв'язки між ними. Діаграма розроблена з використанням об'єктно-орієнтованого підходу та принципів SOLID [24].

Основні класи системи:

1. Клас `SensorData`: представляє дані з датчиків в один момент часу.

Атрибути:

timestamp: *str* – час вимірювання

temp_koloshnik: *float* – температура колошника, °C

temp_middle: *float* – температура середини, °C

temp_bottom: *float* – температура низу, °C

cooling_pressure: *float* – тиск охолоджувальної води, бар

power_status: *str* – статус живлення ('normal', 'emergency')

time_since_failure: *int* – час після відмови, хв

gradient_vertical: *float* – вертикальний градієнт, °C

gradient_horizontal: *float* – горизонтальний градієнт, °C

Методи:

validate() – перевірка коректності даних

to_dict() – перетворення у словник для JSON

2. Клас `Prediction`: представляє результати прогнозування температур.

Атрибути:

timestamp: *str* – час прогнозу

pred_temp_koloshnik: *float* – прогноз колошника через 30 хв, °C

pred_temp_middle: *float* – прогноз середини через 30 хв, °C

pred_temp_bottom: *float* – прогноз низу через 30 хв, °C

minutes_ahead: *int* – горизонт прогнозування

Методи:

calculate() – виконання прогнозування

get_trend() – визначення тренду (підвищення/зниження/стабільно)

3. Клас RiskAssessment: представляє оцінку ризику термошоку.

Атрибути:

risk_score: int – сумарний бал ризику (0 - 100)

risk_level: str – рівень ризику ('Низький', 'Помірний', 'Середній', 'Високий', 'Критичний')

priority: str – пріоритет ('D', 'C', 'B', 'A', 'A+')

recommendation: str – текстова рекомендація

risk_factors: list – список факторів з вагами

Методи:

calculate(data: SensorData, pred: Prediction) – розрахунок ризику

get_alert_level() – визначення необхідності тривоги

4. Клас Alert: представляє тривожне повідомлення.

Атрибути:

id: int – унікальний ідентифікатор

timestamp: str – час виникнення

alert_type: str – тип тривоги

severity: str – серйозність ('A', 'B')

message: str – повідомлення

acknowledged: bool – ознака підтвердження

Методи:

acknowledge() – підтвердження тривоги

is_active() – перевірка активності

5. Клас DataGenerator: генерація симульованих даних.

Атрибути:

scenario: str – поточний сценарій ('normal', 'fluctuation', 'blackout')

last_values: dict – останні згенеровані значення

Методи:

generate_sensor_data() – генерація даних датчиків

set_scenario(scenario: str) – зміна сценарію

6. Клас `DatabaseManager`: управління базою даних.

Методи:

save_sensor_data(data: SensorData) – збереження даних датчиків

save_prediction(pred: Prediction) – збереження прогнозу

save_alert(alert: Alert) – збереження тривоги

get_history(limit: int) – отримання історії

get_alerts(acknowledged: bool) – отримання тривог

7. Клас `APIServer`: REST API сервер.

Атрибути:

app: FastAPI – екземпляр FastAPI додатку

Методи:

setup_routes() – налаштування маршрутів

run() – запуск сервера

8. Клас `FurnaceState`: агрегований стан печі.

Атрибути:

current_data: SensorData – поточні дані

prediction: Prediction – поточний прогноз

risk: RiskAssessment – поточна оцінка ризику

last_update: str – час останнього оновлення

Методи:

update() – оновлення стану

get_summary() – отримання зведеної інформації

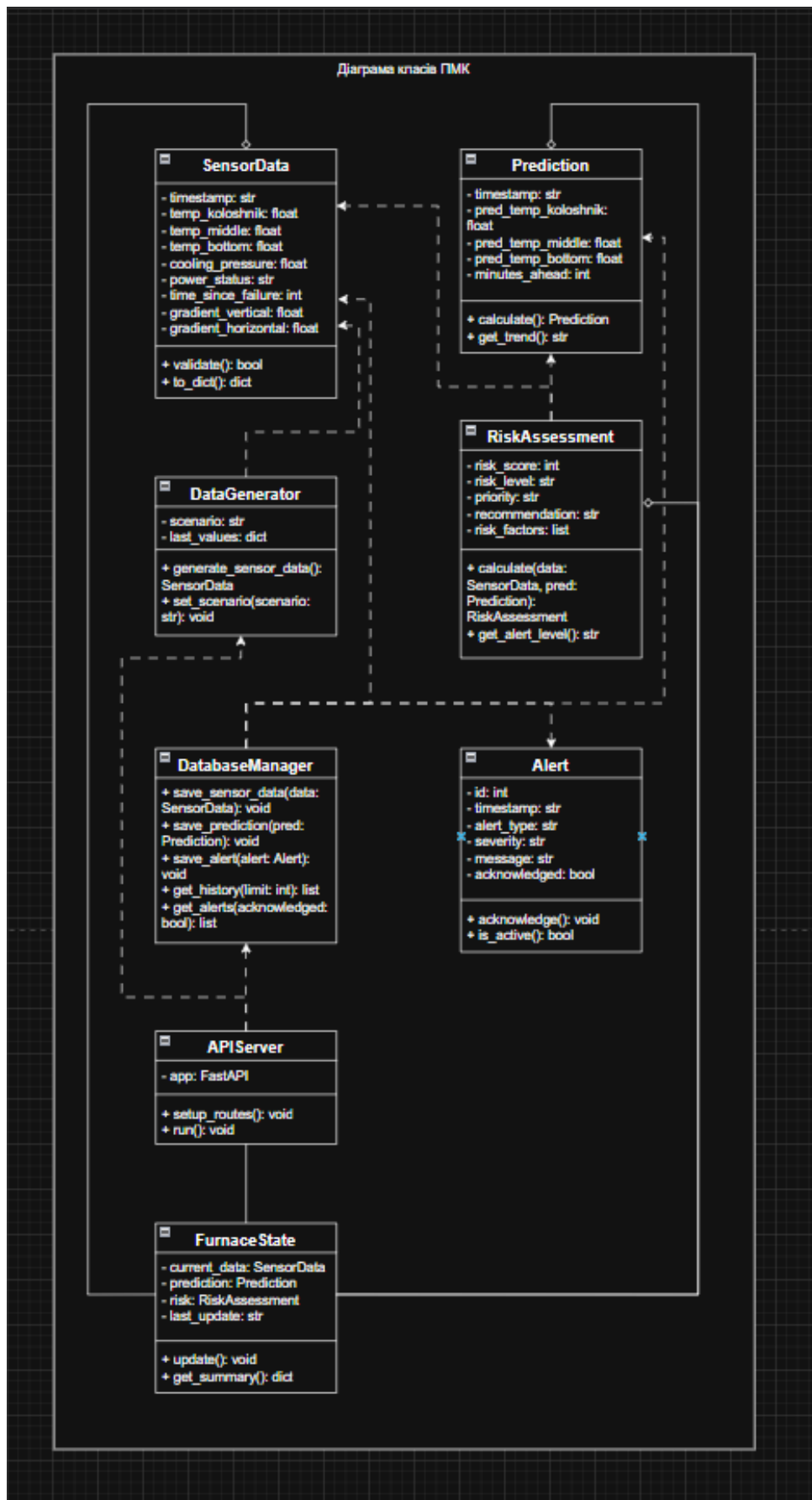


Рисунок 3.3 – Діаграма класів ПМК

Зв'язки між класами:

- *RiskAssessment* використовує *SensorData* та *Prediction* для розрахунку (*uses*).
- *APIServer* використовує *DataGenerator* для генерації даних та *DatabaseManager* для збереження (*uses*).
- *DatabaseManager* зберігає екземпляри *SensorData*, *Prediction* та *Alert* (*aggregation*).
- *FurnaceState* агрегує *SensorData*, *Prediction* та *RiskAssessment* (*composition*).
- *DataGenerator* створює екземпляри *SensorData* (*creates*).

Застосовані шаблони проектування [24]:

- Фабрика (Factory): *DataGenerator* створює об'єкти *SensorData* різних сценаріїв.
- Спостерігач (Observer): система тривоги сповіщає підписників при зміні рівня ризику.
- Одинак (Singleton): *DatabaseManager* забезпечує єдиний екземпляр підключення до БД.
- Проксі (Proxy): *APIServer* виступає посередником між клієнтами та логікою системи.
- Стратегія (Strategy): різні сценарії генерації даних реалізовані як різні стратегії.

3.1.4 Діаграма розгортання

Діаграма розгортання (рисунок 3.4) відображає фізичне розміщення компонентів ПМК на апаратному забезпеченні та показує взаємозв'язки між вузлами.

Основні вузли системи:

1. Сервер обробки даних (Application Server)

- Розташування: серверна кімната доменного цеху
- Апаратне забезпечення: промисловий сервер Intel Xeon 4+ ядра,

RAM 16+ ГБ, SSD 256+ ГБ

- Програмне забезпечення: Python 3.10+, FastAPI, SQLite
- Резервне живлення: ДГУ з часом автономної роботи не менше 4

годин

- Компоненти:

- a) Модуль збору даних (Data Collector)
- b) Модуль прогнозування (Prediction Module)
- c) Модуль оцінки ризику (Risk Module)
- d) REST API (FastAPI)
- e) База даних SQLite

2. Edge-пристрій (Edge Device)

- Розташування: безпосередньо біля доменної печі
- Апаратне забезпечення: промисловий міні-ПК або PLC з

підтримкою OPC UA

- Призначення: попередня обробка даних, буферизація вимірювань

при втраті зв'язку

- Компоненти:

- a) OPC UA клієнт
- b) Модуль попередньої обробки (фільтрація, згладжування)
- c) Кеш даних

3. АРМ оператора (Operator Workstation)

- Розташування: пульт управління доменного цеху
- Апаратне забезпечення: робоча станція з монітором, клавіатурою,

мишею

- Програмне забезпечення: Windows 10/11, Power BI Desktop

- Компоненти:
 - a) Power BI дашборди (4 сторінки)
 - b) Веб-браузер для доступу до Swagger-документації API

4. Контролер АСУ ТП (PLC)

- Розташування: шафа управління біля печі
- Апаратне забезпечення: Siemens S7-400
- Програмне забезпечення: вбудоване ПЗ контролера
- Компоненти:
 - a) Модуль вводу/виводу датчиків
 - b) OPC UA сервер

5. Система резервного живлення (UPS + DG)

- Розташування: електроприміщення цеху
- Апаратне забезпечення: ДГУ (дизель-генераторна установка) + акумуляторні батареї
- Призначення: забезпечення живлення критичних компонентів при знеструмленні

Таблиця 3.2 – Мережеві з'єднання та протоколи:

З'єднання	Протокол	Призначення
Контролер ↔ Edge-пристрій	OPC UA	Передача даних з датчиків
Edge-пристрій ↔ Сервер	TCP/IP (HTTP/HTTPS)	Передача попередньо оброблених даних
Сервер ↔ АРМ оператора	TCP/IP (HTTP/HTTPS)	REST API запити (Power BI)
Сервер ↔ ДГУ	Modbus/TCP	Моніторинг стану резервного живлення

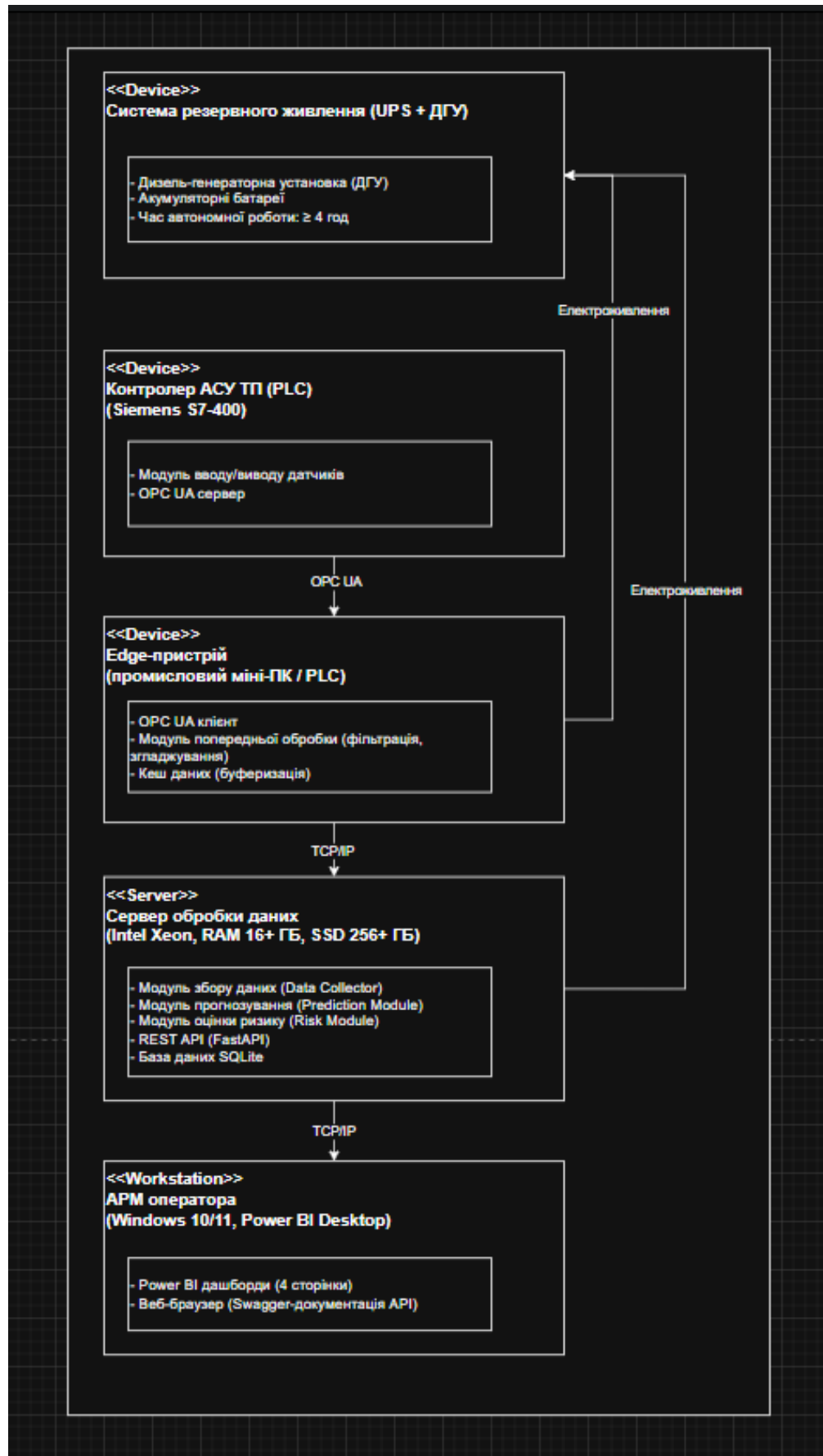


Рисунок 3.4 – Діаграма розгортання ПМК

Забезпечення відмовостійкості:

1. Резервування сервера – дублювання основного сервера на резервному (дзеркалювання БД).
2. Буферизація на Edge – при втраті зв'язку з сервером Edge-пристрій зберігає дані локально до відновлення.
3. Резервне живлення – сервер та Edge-пристрій підключені до ДГУ, час автономної роботи – не менше 4 годин.
4. Автоматичне відновлення – система автоматично відновлює збір та передачу даних після відновлення живлення.

3.2 Розробка фізичної моделі проєкту ПМК

Фізична модель програмно-методичного комплексу (ПМК) описує конкретну реалізацію системи на рівні програмного забезпечення, апаратного забезпечення, баз даних та мережевої інфраструктури. На відміну від логічної моделі (підрозділ 3.1), яка описує «що» робить система, фізична модель описує «як» це зроблено.

3.2.1 Загальна архітектура системи

ПМК реалізовано за клієнт-серверною архітектурою з елементами edge-обробки даних. Такий підхід забезпечує:

- Масштабованість, полягає у можливості додавання нових датчиків та розширенні функціональності без зміни основної архітектури.
- Надійність, полягає у edge-обробці та дозволяє зберігати дані локально при втраті зв'язку з сервером.

- Швидкодію, полягає у попередній обробці даних безпосередньо біля джерела, що зменшує навантаження на сервер.
- Відмовостійкість, полягає у резервуванні критичних компонентів та можливості роботи в автономному режимі.

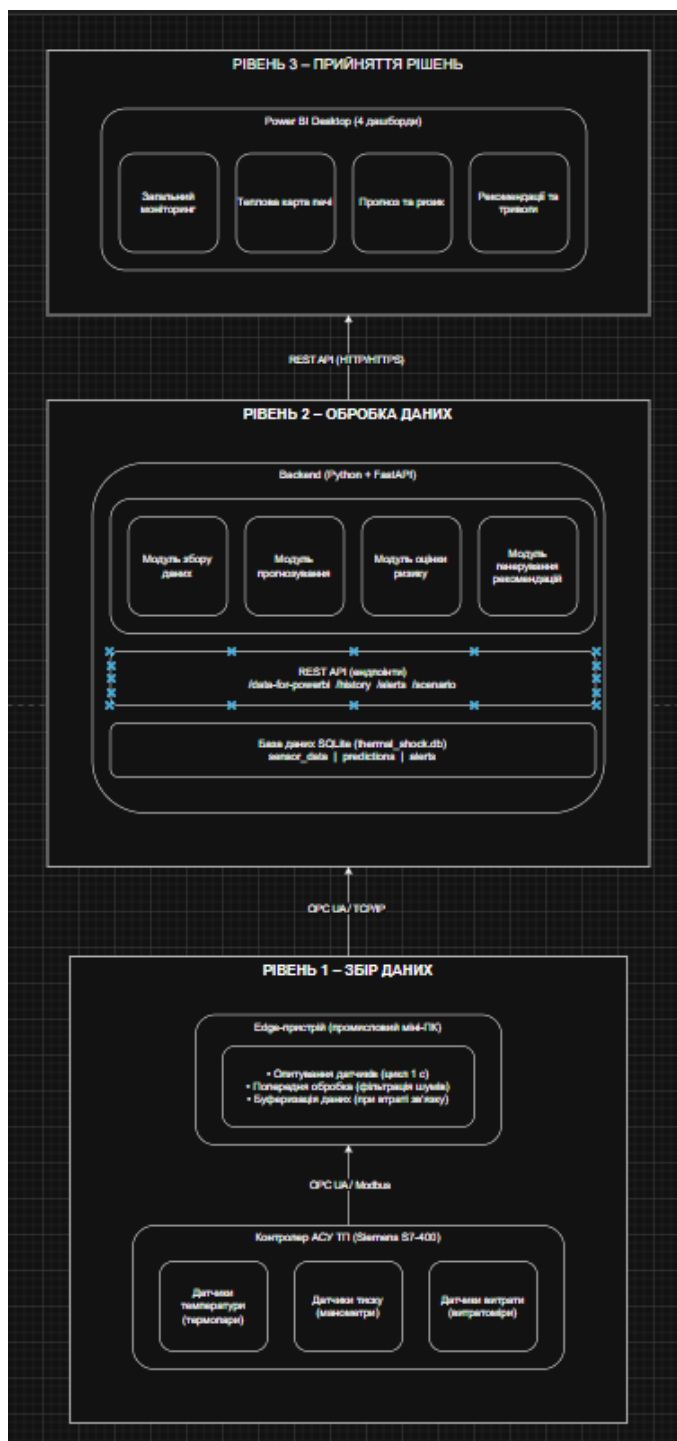


Рисунок 3.5 – Трирівнева архітектура ПМК

Схема архітектури ПМК:

Архітектура складається з трьох логічних рівнів:

1. Рівень збору даних – фізичні датчики, контролери АСУ ТП та edge-пристрої, які забезпечують первинне отримання та попередню обробку даних.
2. Рівень обробки даних – серверна частина (FastAPI + Python), яка виконує прогнозування, оцінку ризику, збереження даних та надає REST API.
3. Рівень прийняття рішень – Power BI дашборди, які забезпечують візуалізацію даних, інтерактивний аналіз та підтримку прийняття рішень персоналом.

3.2.2 Вибір технологічного стеку

Вибір технологій для реалізації ПМК обґрунтовано в підрозділі 2.1. Таблиця 3.2 наводить остаточний перелік обраних технологій з коротким обґрунтуванням для кожної.

Таблиця 3.2 – Технологічний стек ПМК

Компонент	Обране рішення	Версія	Обґрунтування
Мова програмування	Python	3.10+	Широкий вибір бібліотек, простота, швидкість розробки, підтримка асинхронності
Веб-фреймворк	FastAPI	0.100+	Висока продуктивність, автоматична документація, валідація даних, асинхронність
База даних	SQLite	3.x	Легковага, не потребує сервера, достатня для прототипу, вбудована в Python
Інструмент візуалізації	Power BI Desktop	Остання	Безкоштовний, потужна візуалізація, інтерактивні дашборди, інтеграція з REST AP

Продовження таблиці 3.2

Компонент	Обране рішення	Версія	Обґрунтування
Протокол зв'язку з АСУ ТП	OPC UA	-	Промисловий стандарт, підтримується Siemens S7-400, захищений, масштабований
Система контролю версій	Git + GitHub	-	Стандарт, збереження історії, можливість публікації в репозиторії університету
Середовище виконання	Uvicorn	0.20+	Високопродуктивний ASGI сервер для FastAPI
Робоча станція	Windows 11	-	Для роботи з Power BI Desktop
Серверна ОС	Ubuntu Server	22.04 LTS	Стабільність, безкоштовність, підтримка Python

Додаткові бібліотеки Python:

- *fastapi* – веб-фреймворк для створення REST API.
- *uvicorn* – ASGI сервер для запуску FastAPI.
- *sqlite3* – вбудований модуль для роботи з SQLite.
- *requests* – для виконання HTTP-запитів (у консольному меню).
- *random* – для генерації випадкових даних (симуляція).
- *datetime* – для роботи з часом та датами.
- *typing* – для анотацій типів.

3.2.3 Структура бази даних

Як зазначено в підрозділі 2.3.3, база даних SQLite містить три основні таблиці, які забезпечують збереження історії вимірювань, прогнозів та тривоги.

Таблиця 3.3 – Структура таблиці *sensor_data*

Поле	Тип даних	Обмеження	Опис
<i>id</i>	INTEGER	PRIMARY KEY AUTOINCREMENT	Унікальний ідентифікатор запису
<i>timestamp</i>	TEXT	NOT NULL	Час вимірювання у форматі 'YYYY-MM-DD HH:MM:SS'
<i>temp_koloshnik</i>	REAL	-	Температура колошника, °C
<i>temp_middle</i>	REAL	-	Температура середини печі, °C
<i>temp_bottom</i>	REAL	-	Температура низу (горна), °C
<i>cooling_pressure</i>	REAL	-	Тиск охолоджувальної води, бар
<i>power_status</i>	TEXT	CHECK (power_status IN ('normal', 'emergency'))	Статус живлення
<i>time_since_failure</i>	INTEGER	-	Час після відмови живлення, хвилин
<i>gradient_vertical</i>	REAL	-	Вертикальний градієнт температури, °C
<i>gradient_horizontal</i>	REAL	-	Горизонтальний градієнт температури, °C

Таблиця 3.4 – Структура таблиці *predictions*

Поле	Тип даних	Обмеження	Опис
<i>id</i>	INTEGER	PRIMARY KEY AUTOINCREMENT	Унікальний ідентифікатор запису
<i>timestamp</i>	TEXT	NOT NULL	Час прогнозу
<i>pred_temp_koloshnik</i>	REAL	-	Прогноз температури колошника через 30 хв, °C
<i>pred_temp_middle</i>	REAL	-	Прогноз температури середини через 30 хв, °C
<i>pred_temp_bottom</i>	REAL	-	Прогноз температури низу через 30 хв, °C
<i>risk_score</i>	INTEGER	-	Бальна оцінка ризику (0–100)
<i>risk_level</i>	TEXT	-	Рівень ризику (текст)
<i>recommendation</i>	TEXT	-	Текстова рекомендація для персоналу

Таблиця 3.5 – Структура таблиці *alerts*

Поле	Тип даних	Обмеження	Опис
<i>id</i>	INTEGER	PRIMARY KEY AUTOINCREMENT	Унікальний ідентифікатор тривоги
<i>timestamp</i>	TEXT	NOT NULL	Час виникнення тривоги
<i>alert_type</i>	TEXT	NOT NULL	Тип тривоги (наприклад, 'THERMAL_SHOCK_RISK')
<i>severity</i>	TEXT	NOT NULL	Рівень серйозності ('A' або 'B')
<i>message</i>	TEXT	NOT NULL	Повідомлення тривоги
<i>acknowledged</i>	BOOLEAN	DEFAULT 0	Ознака підтвердження (0 - не підтверджена, 1 – підтверджена)

ER-діаграма бази даних (рисунок 3.6) відображає зв'язки між таблицями.

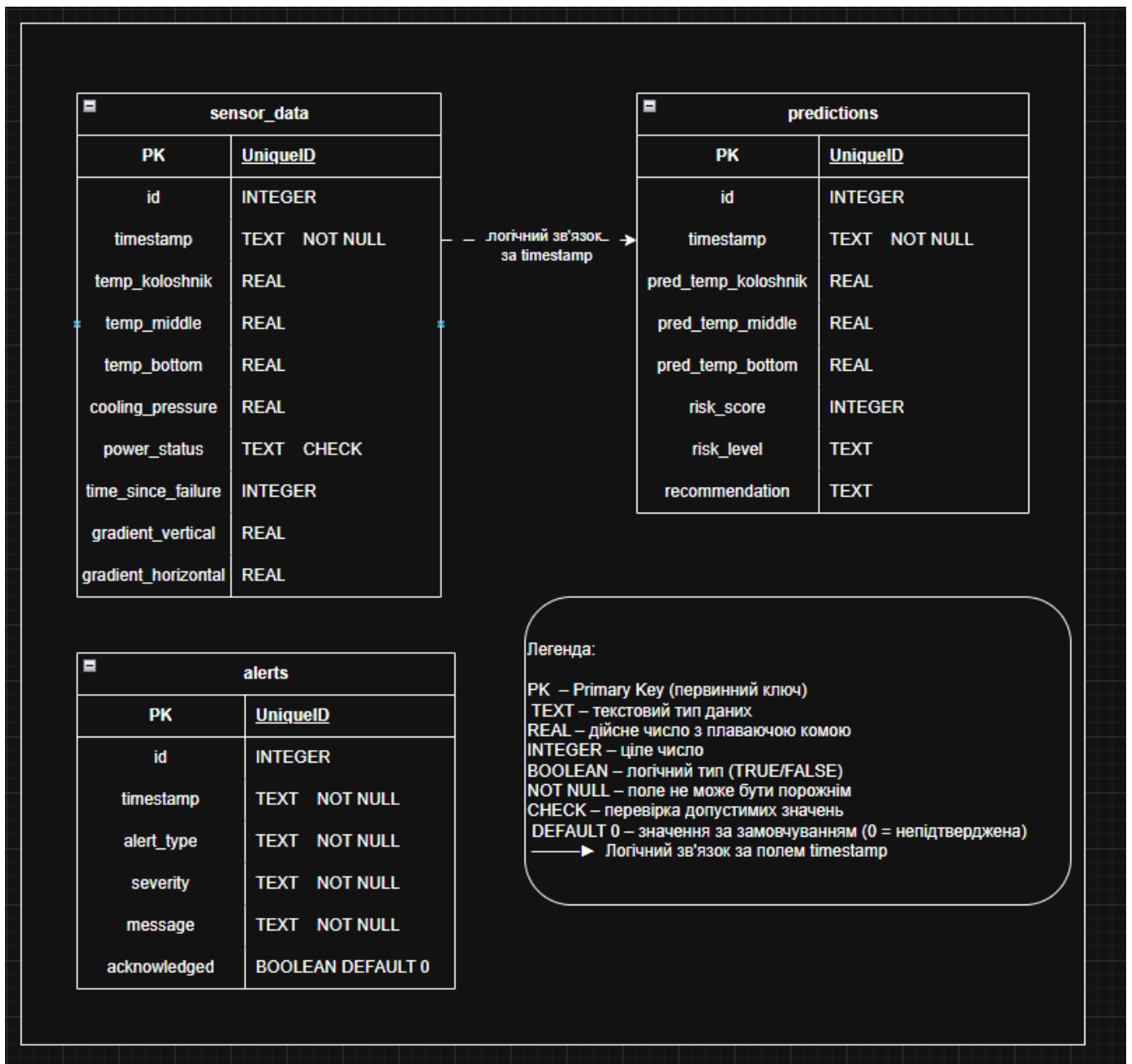


Рисунок 3.6 – ER-діаграма бази даних ПМК

Зв'язки між таблицями:

- *sensor_data* та *predictions* пов'язані через поле *timestamp* (неявний зв'язок, без зовнішніх ключів для забезпечення швидкодії).
- *alerts* є незалежною таблицею, яка фіксує тривожні події без прямого зв'язку з іншими таблицями.

Обмеження цілісності даних:

- Перевірка допустимих значень статусу живлення (*normal* або *emergency*) за допомогою *CHECK*.
- Автоматичне присвоєння часу створення запису при вставці (*DEFAULT CURRENT_TIMESTAMP* в майбутніх версіях).
- Унікальність ідентифікаторів (*PRIMARY KEY*).

3.2.4 Особливості роботи в умовах знеструмлення

Однією з ключових вимог до ПМК є здатність працювати в умовах екстреного знеструмлення. Для забезпечення цього передбачено наступні механізми:

1. Сценарії роботи системи. Система підтримує три сценарії роботи, які визначають поведінку генерації даних (таблиця 3.6).

Таблиця 3.6 – Сценарії роботи ПМК

Сценарій	Статус живлення	Поведінка датчиків	Характер змін	Використання
Normal (штатний)	<i>normal</i>	Стабільні показники, випадкові коливання в межах норми	Незначні відхилення, плавне повернення до цільових значень	Основний режим роботи
Fluctuation (коливання)	Переважно <i>normal</i> , іноді <i>emergency</i>	Нестабільні показники, широкий діапазон коливань	Різкі стрибки температур та тиску, часті зміни	Симуляція перебоїв живлення
Blackout (повне відключення)	<i>emergency</i>	Монотонне зниження температур та тиску	Постійне охолодження печі, падіння тиску	Навчальний режим, тестування алгоритмів

2. Механізм генерації даних при різних сценаріях. У кодї ПМК (функція *generate_sensor_data()*) реалізовано адаптивну генерацію даних залежно від обраного сценарію:

- Штатний режим (*normal*): система прагне повернути температури до цільових значень (*_target_values*) зі швидкістю відновлення 2°C за крок (для колошника) та пропорційно менше для інших зон. Додаються невеликі випадкові коливання для імітації реальних умов.

- Режим коливань (*fluctuation*): генеруються випадкові значення в широкому діапазоні (наприклад, колошник – від 480 до 780°C). Статус живлення змінюється на *emergency* з ймовірністю 30%.

- Режим повного відключення (*blackout*): температури монотонно знижуються (колошник – на 3 - 8°C за крок), тиск охолодження падає на 0,02 - 0,08 бар за крок. Статус живлення – *emergency*, час після відмови зростає до 10 - 25 хвилин.

3. Edge-обробка. Edge-пристрій (промисловий міні-ПК) виконує наступні функції:

- Опитування датчиків – циклічне зчитування даних з контролера АСУ ТП через OPC UA з періодом 1 секунда.

- Попередня обробка – фільтрація шумів (ковзне середнє), перевірка допустимих діапазонів, видалення викидів.

- Буферизація – при втраті зв'язку з сервером дані зберігаються локально в кеші (до 1000 записів) та передаються після відновлення.

- Передача даних – відправка оброблених даних на сервер через REST API (ендпоінт */data-for-powerbi*) кожні 10 - 30 секунд.

4. Резервне живлення. Для забезпечення роботи ПМК при знеструмленні передбачено:

- Сервер – підключений до ДГУ (дизель-генераторної установки) з часом автономної роботи не менше 4 годин.

– Edge-пристрій – підключений до акумуляторних батарей з часом роботи не менше 2 годин.

– АРМ оператора – підключений до ДГУ або безперебійного блоку живлення (UPS).

5. Алгоритм дій системи при знеструмленні

1) Система фіксує перехід статусу живлення в режим *emergency*.

2) Запускається таймер часу після відмови (*time_since_failure*).

3) Зменшується частота оновлення даних з 1 с до 5 с для економії енергії.

4) Підвищується пріоритет прогнозування та оцінки ризику (кожні 10 с замість 30 с).

5) При досягненні критичних порогів (колошник $< 500^{\circ}\text{C}$, тиск $< 2,5$ бар) генеруються тривоги.

6) При відновленні живлення система автоматично переходить у штатний режим.

3.3 Особливості реалізації програмних компонентів

Даний підрозділ містить детальний опис програмної реалізації основних компонентів ПМК. Реалізація виконана на мові Python з використанням веб-фреймворку FastAPI, бази даних SQLite та бібліотек для роботи з даними та часом. Усі програмні компоненти розроблено з урахуванням вимог до надійності, швидкодії та модульності [17; 18].

3.3.1 Загальна структура backend-компоненту

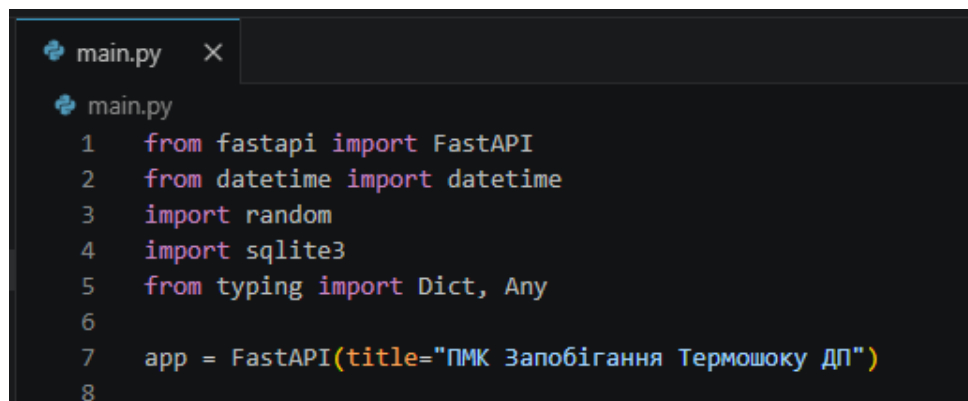
Backend-частина ПМК реалізована у вигляді двох основних файлів:

- *main.py*: основний файл сервера з реалізацією FastAPI додатку, функціями генерації даних, прогнозування, оцінки ризику та API-ендпоінтів.
- *menu.py*: консольне меню для керування сценаріями роботи печі.

Основні модулі *main.py*:

1. Ініціалізація бази даних – створення таблиць *sensor_data*, *predictions*, *alerts*.
2. Генерація даних – симуляція роботи датчиків для різних сценаріїв.
3. Прогнозування температур – реалізація математичної моделі теплопереносу.
4. Оцінка ризику – бальна система оцінки ризику термошоку.
5. REST API – надання даних для Power BI та інших клієнтів.

Початкова структура файлу *main.py* (рисунок 3.7):



```
main.py x
main.py
1 from fastapi import FastAPI
2 from datetime import datetime
3 import random
4 import sqlite3
5 from typing import Dict, Any
6
7 app = FastAPI(title="ПМК Запобігання Термошоку ДП")
8
```

Рисунок 3.7 Підвантаження бібліотек у *main.py*

3.3.2 Модуль збору даних та симуляція сенсорів

Модуль збору даних забезпечує генерацію симульованих даних з датчиків для трьох сценаріїв роботи печі. У реальних умовах цей модуль буде замінений на підключення до АСУ ТП через OPC UA.

Функція `generate_sensor_data()` є основним генератором даних. Вона використовує глобальні змінні для зберігання останніх значень та цільових параметрів (рисунок 3.8):

```
106 # ===== СЦЕНАРІЇ =====
107 current_scenario = "normal"
108
109 _last_values = {
110     "temp_koloshnik": 600.0,
111     "temp_middle": 950.0,
112     "temp_bottom": 1380.0,
113     "cooling_pressure": 3.5
114 }
115
116 # Цільові значення для плавного повернення до норми
117 _target_values = {
118     "temp_koloshnik": 650.0,
119     "temp_middle": 1000.0,
120     "temp_bottom": 1400.0,
121     "cooling_pressure": 3.8
122 }
123
```

Рисунок 3.8 Блок сценаріїв з вихідними даними

Генерація даних для різних сценаріїв:

1. Штатний режим (*normal*) – плавне повернення до цільових значень (рисунок 3.9):

```

main.py
main.py
124 def generate_sensor_data() -> Dict[str, Any]:
130     if current_scenario == "normal":
133         # Колошник
134         if _last_values["temp_koloshnik"] < _target_values["temp_koloshnik"]:
135             new_koloshnik = _last_values["temp_koloshnik"] + recovery_speed
136         else:
137             new_koloshnik = _last_values["temp_koloshnik"] + random.uniform(-3, 3)
138
139         # Середина
140         if _last_values["temp_middle"] < _target_values["temp_middle"]:
141             new_middle = _last_values["temp_middle"] + recovery_speed * 0.8
142         else:
143             new_middle = _last_values["temp_middle"] + random.uniform(-2, 2)
144
145         # Низ
146         if _last_values["temp_bottom"] < _target_values["temp_bottom"]:
147             new_bottom = _last_values["temp_bottom"] + recovery_speed * 0.6
148         else:
149             new_bottom = _last_values["temp_bottom"] + random.uniform(-1, 1)
150
151         # Тиск охолодження
152         if _last_values["cooling_pressure"] < _target_values["cooling_pressure"]:
153             new_pressure = _last_values["cooling_pressure"] + 0.05
154         else:
155             new_pressure = _last_values["cooling_pressure"] + random.uniform(-0.03, 0.03)
156
157         # Обмеження, щоб не вилізли за межі норми
158         new_koloshnik = min(_target_values["temp_koloshnik"] + 30, new_koloshnik)
159         new_koloshnik = max(550, new_koloshnik)
160
161         new_middle = min(_target_values["temp_middle"] + 30, new_middle)
162         new_middle = max(850, new_middle)
163
164         new_bottom = min(_target_values["temp_bottom"] + 30, new_bottom)
165         new_bottom = max(1350, new_bottom)
166
167         new_pressure = max(3.5, min(5.2, new_pressure))
168
169         power_status = "normal"
170         time_since_failure = 0
171

```

Рисунок 3.9 Сценарій основного режиму роботи

2. Режим коливань (*fluctuation*) – випадкові коливання в широкому діапазоні (рисунок 3.10):

```

172 # =====
173 # СЦЕНАРІЙ 2: КОЛИВАННЯ ЖИВЛЕННЯ
174 # =====
175 elif current_scenario == "fluctuation":
176     new_koloshnik = _last_values["temp_koloshnik"] + random.uniform(-15, 15)
177     new_middle = _last_values["temp_middle"] + random.uniform(-10, 10)
178     new_bottom = _last_values["temp_bottom"] + random.uniform(-8, 8)
179     new_pressure = _last_values["cooling_pressure"] + random.uniform(-0.3, 0.3)
180
181     new_koloshnik = max(480, min(780, new_koloshnik))
182     new_middle = max(750, min(1150, new_middle))
183     new_bottom = max(1250, min(1480, new_bottom))
184     new_pressure = max(2.5, min(5.2, new_pressure))
185
186     power_status = "emergency" if random.random() < 0.3 else "normal"
187     time_since_failure = random.randint(0, 15) if power_status == "emergency" else 0
188

```

Рисунок 3.10 Сценарій симуляції перебоїв живлення

3. Режим повного відключення (*blackout*) – монотонне зниження температур (рисунок 3.11):

```

189 # =====
190 # СЦЕНАРІЙ 3: ПОВНЕ ВІДКЛЮЧЕННЯ
191 # =====
192 elif current_scenario == "blackout":
193     new_koloshnik = _last_values["temp_koloshnik"] - random.uniform(3, 8)
194     new_middle = _last_values["temp_middle"] - random.uniform(2, 5)
195     new_bottom = _last_values["temp_bottom"] - random.uniform(1, 3)
196     new_pressure = _last_values["cooling_pressure"] - random.uniform(0.02, 0.08)
197
198     new_koloshnik = max(420, new_koloshnik)
199     new_middle = max(750, new_middle)
200     new_bottom = max(1250, new_bottom)
201     new_pressure = max(2.0, new_pressure)
202
203     power_status = "emergency"
204     time_since_failure = random.randint(10, 25)
205

```

Рисунок 3.10 Сценарій повного відключення

Збереження даних у базу даних виконується функцією *save_to_database()* (рисунок 3.11):

```

52 def save_to_database(data: Dict[str, Any]):
53     conn = sqlite3.connect('thermal_shock.db')
54     cursor = conn.cursor()
55     cursor.execute('''
56         INSERT INTO sensor_data
57         (timestamp, temp_koloshnik, temp_middle, temp_bottom, cooling_pressure,
58         power_status, time_since_failure, gradient_vertical, gradient_horizontal)
59         VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
60     ''', (
61         data["timestamp"],
62         data["temp_koloshnik"],
63         data["temp_middle"],
64         data["temp_bottom"],
65         data["cooling_pressure"],
66         data["power_status"],
67         data["time_since_failure"],
68         data["gradient_vertical"],
69         data["gradient_horizontal"]
70     ))
71     conn.commit()
72     conn.close()

```

Рисунок 3.11 Функція збереження даних

3.3.3 Реалізація математичної моделі прогнозування

Функція `predict_temperature()` реалізує математичну модель теплопереносу, яка прогнозує зміну температури в трьох зонах печі на заданий період часу (рисунок 3.12).

```

240 # ===== ПРОГНОЗУВАННЯ =====
241 def predict_temperature(current_data: Dict[str, Any], minutes_ahead: int = 30) -> Dict[str, Any]:
242     alpha_koloshnik = 0.95
243     alpha_middle = 0.85
244     alpha_bottom = 0.75
245
246     cooling_factor = current_data["cooling_pressure"] / 3.5
247
248     pred_koloshnik = current_data["temp_koloshnik"] - (alpha_koloshnik * cooling_factor * 0.1 * minutes_ahead / 15)
249     pred_middle = current_data["temp_middle"] - (alpha_middle * cooling_factor * 0.08 * minutes_ahead / 15)
250     pred_bottom = current_data["temp_bottom"] - (alpha_bottom * cooling_factor * 0.06 * minutes_ahead / 15)
251
252     if current_data["power_status"] == "emergency":
253         pred_koloshnik -= 10 * (minutes_ahead / 15)
254         pred_middle -= 8 * (minutes_ahead / 15)
255         pred_bottom -= 5 * (minutes_ahead / 15)
256
257     return {
258         "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
259         "pred_temp_koloshnik": round(pred_koloshnik, 1),
260         "pred_temp_middle": round(pred_middle, 1),
261         "pred_temp_bottom": round(pred_bottom, 1),
262         "minutes_ahead": minutes_ahead
263     }
264

```

Рисунок 3.12 Прогнозування зміни температури

Математичне обґрунтування формули. Формула базується на спрощеному рівнянні теплопереносу:

$alpha * cooling_factor * 0.1 * minutes_ahead / 15$ – швидкість охолодження для колошника, де:

- $alpha$ – емпіричний коефіцієнт теплопереносу (0.95 для колошника, 0.85 для середини, 0.75 для низу);
- $cooling_factor$ – відношення поточного тиску води до номінального (3.5 бар);
- 0.1 – базова швидкість охолодження для колошника (°C за 15 хв при нормальному тиску);
- $minutes_ahead / 15$ – масштабування на заданий горизонт прогнозування.

3.3.4 Реалізація алгоритму оцінки ризику

Функція `calculate_risk()` реалізує бальну систему оцінки ризику термошоку на основі шести факторів (рисунок 3.13).

```

266 def calculate_risk(data: Dict[str, Any], prediction: Dict[str, Any]) -> Dict[str, Any]:
306     elif data["time_since_failure"] > 10:
308         risk_factors.append({ "factor": "Значний час після відмови", "weight": 8 })
309
310     if risk_score >= 80:
311         risk_level = "Критичний"
312         priority = "A"
313         recommendation = "Негайний випуск чавуну. Вимкнути нагрів, максимальне охолодження."
314     elif risk_score >= 60:
315         risk_level = "Високий"
316         priority = "A"
317         recommendation = "Аварійна ситуація! Збільшити тиск охолодження. Підготувати резервне живлення."
318     elif risk_score >= 40:
319         risk_level = "Середній"
320         priority = "B"
321         recommendation = "Підвищена увага! Знизити навантаження. Перевірити систему охолодження."
322     elif risk_score >= 20:
323         risk_level = "Помірний"
324         priority = "C"
325         recommendation = "Посилений моніторинг. Перевірити градієнти температури."
326     else:
327         risk_level = "Низький"
328         priority = "D"
329         recommendation = "Штатний режим. Продовжувати моніторинг."
330
331     if risk_level in ["Високий", "Критичний"]:
332         save_alert(
333             alert_type="THERMAL_SHOCK_RISK",
334             severity=priority,
335             message=f"Рівень ризику: {risk_level}. {recommendation[:50]}..."
336         )
337
338     return {
339         "risk_score": risk_score,
340         "risk_level": risk_level,
341         "priority": priority,
342         "recommendation": recommendation,
343         "risk_factors": risk_factors,
344         "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
345     }

```

Рисунок 3.13 Реалізація оцінки ризику

3.3.5 API-ендпоінти

Backend-сервер надає наступні REST API-ендпоінти для взаємодії з Power BI та іншими клієнтами.

Таблиця 3.7 – API-ендпоінти ПМК

Метод	Ендпоінт	Призначення	Формат відповіді
GET	/	Перевірка роботи сервера, отримання інформації про систему	JSON
GET	/scenario	Отримання поточного сценарію роботи	JSON
POST	/scenario/{name}	Зміна сценарію роботи (normal/fluctuation/blackout)	JSON
GET	/data-for-powerbi	Отримання поточного знімка даних для Power BI	JSON
GET	/history	Отримання історичних даних (останні 100 записів)	JSON
GET	/alerts	Отримання списку тривог	JSON
POST	/acknowledge-alert/{id}	Підтвердження тривоги	JSON

1. Ендпоінт */data-for-powerbi* (основний для Power BI). Цей ендпоінт є основним джерелом даних для Power BI дашбордів. Він повертає JSON-об'єкт з усіма необхідними показниками для візуалізації (рисунок 3.14).

```

379 @app.get("/data-for-powerbi")
380 def data_for_powerbi():
381     current_data = generate_sensor_data()
382     prediction = predict_temperature(current_data, 30)
383     risk = calculate_risk(current_data, prediction)
384
385     timestamp = datetime.now()
386     date_str = timestamp.strftime("%Y-%m-%d")
387     time_str = timestamp.strftime("%H:%M:%S")
388
389     return {
390         "date": date_str,
391         "time": time_str,
392         "temp_koloshnik": current_data["temp_koloshnik"],
393         "temp_middle": current_data["temp_middle"],
394         "temp_bottom": current_data["temp_bottom"],
395         "cooling_pressure": current_data["cooling_pressure"],
396         "power_status": current_data["power_status"],
397         "time_since_failure": current_data["time_since_failure"],
398         "gradient_vertical": current_data["gradient_vertical"],
399         "gradient_horizontal": current_data["gradient_horizontal"],
400         "pred_temp_koloshnik_30min": prediction["pred_temp_koloshnik"],
401         "pred_temp_middle_30min": prediction["pred_temp_middle"],
402         "pred_temp_bottom_30min": prediction["pred_temp_bottom"],
403         "risk_score": risk["risk_score"],
404         "risk_level": risk["risk_level"],
405         "priority": risk["priority"],
406         "recommendation": risk["recommendation"],
407         "current_scenario": current_scenario
408     }

```

Рисунок 3.14 Ендпоінт */data-for-powerbi*

Приклад відповіді (JSON) (рисунок 3.15):

date	time	temp_koloshnik	temp_middle	temp_bottom	cooling_pressure	power_status	time_since_failure	gradient_vertical	gradient_horizontal	pred_temp_koloshnik_30min	pred_temp_middle_30min	pred_temp_bottom_30min
21.09.2026	5:45:37	562	859,6	1357,2	3,8	normal	0	397,6	1,4	561,8	859,5	1357,2

Рисунок 3.15 Відповідь ендпоінту /data-for-powerbi через запит Power BI

2. Ендпоінт /history для історичних даних (рисунок 3.16)

```

410 @app.get("/history")
411 def get_history(limit: int = 100):
412     conn = sqlite3.connect('thermal_shock.db')
413     cursor = conn.cursor()
414     cursor.execute('''
415         SELECT * FROM sensor_data
416         ORDER BY id DESC LIMIT ?
417     ''', (limit,))
418     rows = cursor.fetchall()
419     conn.close()
420
421     history = []
422     for row in rows:
423         history.append({
424             "id": row[0],
425             "timestamp": row[1],
426             "temp_koloshnik": row[2],
427             "temp_middle": row[3],
428             "temp_bottom": row[4],
429             "cooling_pressure": row[5],
430             "power_status": row[6],
431             "time_since_failure": row[7],
432             "gradient_vertical": row[8],
433             "gradient_horizontal": row[9]
434         })
435     return history
436

```

Рисунок 3.16 Ендпоінт /history

3. Ендпоінт `/alerts` для отримання тривог (рисунок 3.17)

```
437 @app.get("/alerts")
438 def get_alerts(acknowledged: bool = False):
439     conn = sqlite3.connect('thermal_shock.db')
440     cursor = conn.cursor()
441     cursor.execute('''
442         SELECT * FROM alerts
443         WHERE acknowledged = ?
444         ORDER BY id DESC
445     ''', (1 if acknowledged else 0,))
446     rows = cursor.fetchall()
447     conn.close()
448
449     alerts = []
450     for row in rows:
451         alerts.append({
452             "id": row[0],
453             "timestamp": row[1],
454             "alert_type": row[2],
455             "severity": row[3],
456             "message": row[4],
457             "acknowledged": bool(row[5])
458         })
459     return alerts
460
461 @app.post("/acknowledge-alert/{alert_id}")
462 def acknowledge_alert(alert_id: int):
463     conn = sqlite3.connect('thermal_shock.db')
464     cursor = conn.cursor()
465     cursor.execute('''
466         UPDATE alerts SET acknowledged = 1
467         WHERE id = ?
468     ''', (alert_id,))
469     conn.commit()
470     conn.close()
471     return {"status": "Alert acknowledged", "alert_id": alert_id}
```

Рисунок 3.17 Ендпоінт `/alerts`

3.3.6 Консольне меню для керування сценаріями

Файл `menu.py` реалізує консольне меню для зручного керування сценаріями роботи печі без необхідності використання Swagger-документації (рисунок 3.18).

```

menu.py
1  import requests
2  import sys
3
4  API_URL = "http://127.0.0.1:8000"
5
6  def scenario_selector():
7      # Примусовий вивід меню одразу
8      sys.stdout.write("\n" + "="*60 + "\n")
9      sys.stdout.write("ПКМ ЗАПОБІГАННЯ ТЕРМОШОКУ - КОНСОЛЬНЕ КЕРУВАННЯ\n")
10     sys.stdout.write("="*60 + "\n")
11     sys.stdout.write("Оберіть сценарій роботи печі:\n")
12     sys.stdout.write(" 1 - Штатна робота\n")
13     sys.stdout.write(" 2 - Коливання живлення\n")
14     sys.stdout.write(" 3 - Повне відключення\n")
15     sys.stdout.write(" 4 - Вихід\n")
16     sys.stdout.write("-"*60 + "\n")
17     sys.stdout.flush()
18

```

Рисунок 3.18 Налаштування файлу menu.py

Переваги консольного меню:

- Простота використання для операторів без технічної підготовки.
- Швидке перемикання між сценаріями для тестування.
- Наочне підтвердження зміни сценарію з описом очікуваної поведінки.
- Зворотний зв'язок про помилки (наприклад, якщо сервер не запущено).

3.3.7 Інтеграція з Power BI

Інтеграція з Power BI виконується через веб-підключення до REST API.

Основні кроки налаштування:

1. Запуск сервера – виконання команди `uvicorn main:app --reload` для запуску FastAPI сервера на локальному хості (порт 8000).

2. Підключення в Power BI – використання вбудованого конектора "Web" для отримання даних з ендпоінтів:

- */data-for-powerbi* – для поточних даних (таблиця "Поточні дані").
- */history?limit=100* – для історичних даних (таблиця "Історія").
- */alerts* – для тривог (таблиця "Тривоги").

3. Оновлення даних – у Power BI Desktop оновлення виконується вручну (кнопка "Оновити" або Ctrl+R). Для автоматичного оновлення необхідна публікація звіту в Power BI Service.

Таблиця 3.8 – Джерела даних у Power BI

Назва таблиці	Ендпоінт	Призначення	Кількість полів
Поточні дані	<i>/data-for-powerbi</i>	Останній знімок показників для карток та індикаторів	17
Історія	<i>/history?limit=100</i>	Масив записів для графіків та аналізу трендів	9
Тривоги	<i>/alerts</i>	Список активних та історичних тривог	5

3.4 Елементи інтерфейсу ПМК

Інтерфейс ПМК реалізовано у вигляді чотирьох інтерактивних дашбордів у Power BI Desktop, які забезпечують візуалізацію даних та інформаційну підтримку персоналу.

3.4.1 Обґрунтування вибору Power BI

Power BI Desktop обрано як інструмент візуалізації через:

- безкоштовність розповсюдження;
- підтримку REST API як джерела даних (ендпоінти /data-for-powerbi, /history, /alerts);
- потужні засоби візуалізації (картки, графіки, датчики, теплові карти);
- умовне форматування для миттєвої ідентифікації відхилень;
- відсутність необхідності розробки окремого фронтенду.

3.4.2 Структура дашбордів

Розроблено чотири сторінки дашбордів:

Сторінка 1: "Загальний моніторинг" – відображає ключові параметри: температури трьох зон (колошник, середина, низ), тиск охолодження, рівень ризику, статус живлення. Графік динаміки температур за останні 100 записів.

Сторінка 2: "Теплова карта печі" – візуалізує вертикальний та горизонтальний градієнти температур за допомогою датчиків (gauges) та стовпчастої діаграми розподілу температур по зонах.




Сторінка 3: "Прогноз та ризик" – показує прогнозовані температури на 30 хвилин, бальну оцінку ризику, пріоритет та текстову рекомендацію для оператора.

Сторінка 4: "Рекомендації та тривоги" – містить таблицю активних тривог та графік історії зміни рівня ризику.

3.4.3 Умовне форматування

Для оперативного візуального контролю застосовано кольорове кодування:

Таблиця 3.8 – Умовне форматування показників

Показник	Норма 	Увага 	Критично 
temp_koloshnik	$\geq 550^{\circ}\text{C}$	500 - 550°C	$< 500^{\circ}\text{C}$
temp_middle	$\geq 850^{\circ}\text{C}$	750 - 850°C	$< 750^{\circ}\text{C}$
temp_bottom	$\geq 1350^{\circ}\text{C}$	1250 - 1350°C	$< 1250^{\circ}\text{C}$
cooling_pressure	≥ 3.5 бар	3.0 - 3.5 бар	< 3.0 бар
power_status	normal	-	emergency

3.4.4 Джерела даних та оновлення

Power BI підключено до трьох веб-джерел:

- /data-for-powerbi - поточні дані (останній знімок показників)
- /history?limit=100 – історія (масив записів для графіків)
- /alerts – тривоги (список активних тривог)

Оновлення даних у Power BI Desktop виконується вручну (кнопка "Оновити" або Ctrl+R). Для автоматичного оновлення необхідна публікація звіту в Power BI Service.

3.4.5 Скріншоти дашбордів

Для ілюстрації роботи ПМК наведено скріншоти чотирьох дашбордів для різних сценаріїв:

Таблиця 3.9 – Перелік скріншотів дашбордів

№	Назва файлу	Чотири сторінки у штатному режимі
1 - 4	dashboard1_4_normal.png	Чотири сторінки у штатному режимі
5 - 8	dashboard1_4_fluctuation.png	Чотири сторінки у режимі коливань
9 - 12	dashboard1_4_blackout.png	Чотири сторінки у режимі повного відключення

3.4.6 Переваги та обмеження

Переваги: простота розгортання, наочність даних, інтерактивність, масштабованість, можливість мобільного доступу.

Обмеження: ручне оновлення у Power BI Desktop, залежність від запущеного сервера, необхідність публікації для автоматизації.

4 ЕКОНОМІЧНІ РОЗРАХУНКИ

4.1 Техніко-економічне обґрунтування

Мета розділу – оцінити економічну доцільність впровадження розробленого програмно-методичного комплексу на ПРАТ «КАМЕТ-СТАЛЬ».

Основні фактори економічного ефекту:

- Зменшення позапланових простоїв доменної печі.
- Зниження витрат на капітальний ремонт футерівки.
- Скорочення часу прийняття рішень персоналом.

4.2 Витрати на розробку та впровадження

4.2.1 Одноразові витрати (капітальні вкладення)

Таблиця 4.1 – Одноразові витрати на розробку та впровадження ПМК

Стаття витрат	Сума, грн	Примітки
Ліцензійне ПЗ (Power BI Desktop)	0	Безкоштовна версія
Серверне обладнання	85 000	Промисловий сервер + UPS
Edge-пристрій (промисловий міні-ПК)	32 000	2 шт. (основний + резервний)
Розробка ПЗ (Python, FastAPI, SQLite)	62 000	2 місяці роботи розробника
Налаштування та інтеграція з АСУ ТП	28 000	OPC UA, тестування
Навчання персоналу (3 особи)	18 000	3 дні тренінгів
ВСЬОГО	225 000	

4.2.2 Поточні витрати (експлуатаційні)

Таблиця 4.2 – Щорічні експлуатаційні витрати

Стаття витрат	Сума, грн/рік
Електроенергія	4 800
Технічне обслуговування обладнання	6 000
Оновлення ПЗ та підтримка	12 000
ВСЬОГО	22 800

4.3 Розрахунок очікуваного економічного ефекту

4.3.1 Вихідні дані для розрахунку (за даними ПРАТ «КАМЕТ-СТАЛЬ»)

Таблиця 4.3 – Основні економічні показники роботи ДП-1М

Показник	Значення
Продуктивність печі	2 893 т чавуну/добу
Ціна 1 т чавуну	22 000 грн
Вартість 1 години простою	~2 650 000 грн
Тривалість простою після термошоку	12 год (середнє значення)
Збитки від 1 аварійного простою	~31 800 000 грн
Частота аварійних зупинок	1 - 2 рази на рік

4.3.2 Прямий економічний ефект

Основні складові ефекту:

– Зменшення кількості аварійних простоїв (з 2 до 1 на рік, за оцінками експертів – зниження на 40–60%).

- Скорочення тривалості простою (з 12 год до 8 год, зниження на 33% за рахунок оперативного прогнозування).

Таблиця 4.4 – Розрахунок очікуваного економічного ефекту

Показник	До впровадження	Після впровадження	Економія
Кількість аварійних зупинок/рік	2	1	1
Тривалість простою, год	12	8	4
Сумарний час простою, год/рік	24	8	16
Збитки від простою, млн грн/рік	63,6	21,2	42,4

Чиста економія за рік: 42 400 000 – 22 800 (експлуатаційні витрати) \approx 42,38 млн грн/рік.

4.3.3 Непрямий економічний ефект

Непрямий економічний ефект полягає у:

- Зниженні витрат на капітальний ремонт футерівки – ~30% (економія ~3–5 млн грн/рік).
- Підвищенні безпеки персоналу (зменшення ризику аварій).
- Покращенні іміджу підприємства (зменшення екологічних ризиків).

4.4 Розрахунок інвестиційної привабливості

Таблиця 4.5 – Грошові потоки за роками (млн грн)

Рік	Витрати (CAPEX + OPEX)	Економія	Чистий потік (CF)
0	-0,225	0	-0,225
1	-0,023	42,4	42,377
2	-0,023	42,4	42,377
3	-0,023	42,4	42,377
4	-0,023	42,4	42,377
5	-0,023	42,4	42,377

Таблиця 4.6 - Розрахунок показників ефективності (при ставці дисконту 15%):

Показник	Значення	Інтерпретація
NPV (чиста теперішня вартість)	~142 млн грн	> 0 – проект прибутковий
IRR (внутрішня норма доходності)	~185%	значно перевищує ставку 15%
Payback (термін окупності)	~2 дні	вкрай швидка окупність
ROI (рентабельність інвестицій)	~18 800%	за 5 років

Висновок: проект є надзвичайно ефективним та окупається практично миттєво (протягом перших днів експлуатації). Висока ефективність зумовлена значними збитками від аварійних простоїв доменної печі.

4.5 Висновки до економічних розрахунків

1. Впровадження ПМК на ПРАТ «КАМЕТ-СТАЛЬ» є економічно доцільним

2. Річна економія становить понад 42 млн грн, що в сотні разів перевищує витрати на розробку.
3. Термін окупності – менше 1 місяця (практично миттєво).
4. Проєкт має високу інвестиційну привабливість ($NPV > 0$, IRR значно вище ставки дисконту).
5. Крім прямого економічного ефекту, впровадження забезпечує:
 - підвищення безпеки персоналу;
 - зниження екологічних ризиків;
 - збільшення надійності обладнання.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи на тему «Програмно-методичний комплекс для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні» було досягнуто поставленої мети та вирішено всі визначені завдання.

1. Аналіз предметної області та технологічного процесу. Проведено комплексне вивчення технологічного процесу доменного виробництва на базі ПРАТ «КАМЕТ-СТАЛЬ». Розглянуто конструктивні особливості доменних печей, зокрема основної печі ДП-1М (корисний об'єм 1513 м³, продуктивність ~2893 т/добу), її системи охолодження та автоматизації.

Визначено, що футерівка доменної печі є найбільш вразливим елементом при екстремому знеструмленні через виникнення термічних напружень, спричинених нерівномірним охолодженням. Найкритичнішими зонами є колошник, засипний апарат, верхня частина шахти та холодильники.

Встановлено, що критичний період для прийняття рішень становить 10 - 15 хвилин після знеструмлення. Перевищення цього часу без належної інформаційної підтримки призводить до незворотних змін у футерівці та тривалих (від кількох діб до кількох місяців) простоїв обладнання.

2. Аналіз існуючих методів та систем захисту. Виконано критичний огляд існуючих на підприємстві методів захисту та інформаційної підтримки при аварійному знеструмленні. Виявлено основні недоліки поточного підходу:

- відсутність оперативного автоматизованого моніторингу температурного поля футерівки в реальному часі в перші критичні хвилини;
- відсутність алгоритмів прогнозування розвитку термошоку;
- суб'єктивність прийняття рішень персоналом в умовах обмеженої інформації;
- тривалий час відновлення печі після аварії (8 - 24 години та більше).

Проведено порівняльний аналіз сучасних методів моніторингу та прогнозування: стаціонарних термопар, волоконно-оптичних датчиків, математичного моделювання, методів штучного інтелекту та експертних систем. Визначено, що найбільш перспективним підходом є комбінація математичного моделювання теплопереносу з експертною системою підтримки прийняття рішень.

3. Розробка математичної моделі та методики дослідження. Розроблено математичну модель теплового стану футерівки, що базується на рівнянні теплопровідності Фур'є для нестационарного режиму. Для практичної реалізації використано спрощену модель, яка описує зміну температури в трьох характерних зонах печі (колошник, середина, низ) з урахуванням коефіцієнтів теплопереносу ($\alpha = 0,95; 0,85; 0,75$ відповідно).

Запропоновано методику оцінки ризику термошоку на основі бальної системи, яка враховує шість ключових факторів: статус живлення (35 балів), тиск охолодження (30 балів), температуру колошника (20 балів), вертикальний градієнт (15 балів), прогнозоване падіння температури (10 балів) та час після відмови (8 балів). Визначено п'ять рівнів ризику (Низький, Помірний, Середній, Високий, Критичний) з відповідними рекомендаціями для персоналу.

4. Розробка програмно-методичного комплексу. Розроблено структурну модель програмно-методичного комплексу (ПМК), що має модульну клієнт-серверну архітектуру та включає шість основних модулів: збору даних, моніторингу, прогнозування, підтримки прийняття рішень, оповіщення та логування.

Створено логічну модель ПМК у вигляді чотирьох UML-діаграм (прецедентів, діяльності, класів, розгортання), що формально описують функціональність, поведінку та структуру системи.

Розроблено фізичну модель ПМК, яка базується на технологічному стеку: Python 3.10+, FastAPI, SQLite, Power BI Desktop. Система підтримує три сценарії роботи (Normal, Fluctuation, Blackout) та забезпечує роботу в умовах

часткового знеструмлення за рахунок edge-обробки даних та резервного живлення.

Реалізовано програмний прототип ПМК, який включає:

- симуляцію датчиків з генерацією даних для трьох сценаріїв;
- функцію прогнозування температур на 30 хвилин (`predict_temperature()`);
- функцію бальної оцінки ризику (`calculate_risk()`) з автоматичним генеруванням рекомендацій та тривоги;
- REST API з ендпоінтами `/data-for-powerbi`, `/history`, `/alerts`, `/scenario`;
- консольне меню для керування сценаріями (`menu.py`).

Створено чотири інтерактивні дашборди в Power BI для візуалізації поточного стану, теплової карти печі, прогнозу та ризику, а також рекомендацій та тривоги. Застосовано умовне форматування для миттєвої ідентифікації критичних відхилень.

5. Економічна доцільність впровадження. Проведено техніко-економічне обґрунтування впровадження ПМК на ПРАТ «КАМЕТ-СТАЛЬ». Розраховано, що одноразові витрати на розробку та впровадження становлять приблизно 225 000 грн, а щорічні експлуатаційні витрати – 22 800 грн.

Очікуваний прямий економічний ефект від впровадження:

- зменшення кількості аварійних простоїв на 40–60% (з 2 до 1 на рік);
- скорочення тривалості простою з 12 до 8 годин;
- річна економія від зменшення простоїв – понад 42 млн грн.

Показники інвестиційної привабливості: NPV \approx 142 млн грн, IRR \approx 185%, термін окупності – менше 1 місяця. Проект є надзвичайно ефективним та рекомендується до впровадження.

6. Наукова новизна та практичне значення. Наукова новизна отриманих результатів полягає в:

- удосконаленні математичної моделі теплового стану футерівки в аварійному режимі з урахуванням комплексного впливу факторів знеструмлення;
- вперше запропонованому алгоритмічному забезпеченні прогнозування термошоку на основі моніторингу градієнтів температур у критичних зонах;
- подальшому розвитку методу інформаційної підтримки прийняття рішень для персоналу з диференційованими рекомендаціями трьох рівнів ризику.

Практичне значення роботи полягає в тому, що розроблені моделі та алгоритми можуть бути використані на ПРАТ «КАМЕТ-СТАЛЬ» та інших металургійних підприємствах для підвищення безпеки та надійності роботи доменних печей. Впровадження ПМК дозволить зменшити ризики руйнування футерівки, скоротити простой обладнання та знизити витрати на капітальні ремонти.

Перспективи подальших досліджень. Отримані результати створюють основу для подальшого розвитку ПМК у напрямках:

- інтеграції з реальною АСУ ТП через OPC UA замість симуляції;
- використання методів машинного навчання для підвищення точності прогнозування;
- розробки повноцінного веб-інтерфейсу для операторів;
- впровадження автоматичного оновлення дашбордів через Power BI Service;
- розширення функціоналу для інших типів металургійних агрегатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. CRU: Blackout knocks out ArcelorMittal mill 'for months' [Електронний ресурс]. Steel Market Update. 2025. URL: <https://www.steelmarketupdate.com/2025/08/05/cru-blackout-knocks-out-arcelormittal-mill-for-months/> (дата звернення: 26.02.2026).
2. How ArcelorMittal Kryvyi Rih operates in conditions of energy shortage [Електронний ресурс]. GMK Center. 2026. URL: <https://gmk.center/en/opinion/how-arcelormittal-kryvyi-rih-operates-in-conditions-of-energy-shortage/> (дата звернення: 28.02.2026).
3. Zhang L., Ni J., Jiao K., Zhang J. A new simulation characterization method to reveal the causes of cracking for cast iron cooling staves. ISIJ International. 2025. Vol. 65, No. 13. P. 2148–2155. URL: https://www.jstage.jst.go.jp/article/isijinternational/advpub/0/advpub_ISIJINT-2025-286/_article (дата звернення: 01.06.2026).
4. Смик О.С. Програмно-методичний комплекс для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні : тези доповіді. Звітна науково-технічна конференція студентів ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА». Запоріжжя, 2026.
5. Технологічна інструкція з експлуатації доменної печі ДП-1М ПРАТ «КАМЕТ-СТАЛЬ» : внутрішній нормативний документ. Кам'янське, 2024.
6. Регламент дій персоналу при екстремому знеструмленні доменного цеху ПРАТ «КАМЕТ-СТАЛЬ» : внутрішній нормативний документ. Кам'янське, 2024.
7. Губін Г.В., Кравченко А.М. Доменне виробництво: теорія і практика. Дніпро : Пороги, 2020. 528 с.
8. Теплофізичні основи роботи доменних печей : монографія / за ред. Ю.С. Пухова. Київ : Техніка, 2019. 312 с.

9. Бабкін В.І., Сухарєв С.М. Автоматизація доменного виробництва. Київ : Вища школа, 2018. 412 с.
10. Шульга І.В. Аварійні режими та захист доменних печей. Київ : Техніка, 2021. 284 с.
11. Automated monitoring and control systems in blast furnace operations. *Ironmaking & Steelmaking Journal*. 2022. Vol. 49, Issue 4. P. 367–382.
12. Li H., Zhang Y. Thermal shock resistance of refractory linings in blast furnaces. *Ceramics International*. 2023. Vol. 49, Issue 12. P. 421–430.
13. Костіков О.А., Іванов В.М. Системи підтримки прийняття рішень в металургії : монографія. Дніпро, 2024. 256 с.
14. Моделювання теплових процесів у доменних печах. Збірник наукових праць НМетАУ. 2022. Вип. 45. С. 112–128.
15. IoT-based predictive maintenance for industrial furnaces. *Journal of Industrial Information Integration*. 2023. Vol. 35. P. 100–115.
16. IEC 61511: Functional safety – Safety instrumented systems for the process industry sector [Електронний ресурс]. URL: <https://webstore.iec.ch/publication/61511> (дата звернення: 20.06.2026).
17. ДСТУ 2156-93. Системи управління технологічними процесами. Основні положення. [Чинний від 1994-07-01]. Київ : Держстандарт України, 1993. 12 с.
18. Lutz M. *Learning Python*. 5th Edition. O'Reilly Media, 2013. 1600 p.
19. Ramalho L. *Fluent Python: Clear, Concise, and Effective Programming*. 2nd Edition. O'Reilly Media, 2021. 1012 p.
20. Grinberg M. *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, 2018. 318 p.
21. SQLite Documentation [Електронний ресурс]. URL: <https://www.sqlite.org/docs.html> (дата звернення: 20.06.2026).
22. Ferrari A., Russo M. *Microsoft Power BI: The Complete Guide*. Microsoft Press, 2021. 480 p.

23. Object Management Group (OMG). OMG Unified Modeling Language (OMG UML) Version 2.5.1 [Електронний ресурс]. 2017. URL: <https://www.omg.org/spec/UML/2.5.1/PDF> (дата звернення: 20.06.2026).

24. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994. 416 p.

25. Методичні рекомендації щодо підготовки та захисту кваліфікаційної роботи бакалавра за ОПП «Комп'ютерні науки» / уклад.: П.І. Сагайда, Н.Ю. Рекова, О.А. Костіков, І.А. Гетьман. Запоріжжя : ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА», 2024. 72 с.

26. Звіт про переддипломну практику студента Смика О.С. за темою «Програмно-методичний комплекс для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні». Кам'янське : ПРАТ «КАМЕТ-СТАЛЬ», 2026. 35 с.

27. Правила технічної експлуатації доменних печей ПРАТ «КАМЕТ-СТАЛЬ» : внутрішній нормативний документ. Кам'янське, 2023.

ДОДАТОК А. ВІДОМОСТІ РОБОТИ

Таблиця А.1 – Відомості роботи

Формат	№ п.п	Назва документу	Найменування об'єкту або вибору	Кількість сторінок
A4	1	Пояснювальна записка	КІНТЕХАД.КН-22-1Б.00.00.00	106
Графічна частина				
A4	2	Актуальність проблеми термошоку доменних печей	КІНТЕХАД.КН-22-1Б.01.00.00	1
A4	3	Мета, об'єкт, предмет та завдання дослідження	КІНТЕХАД.КН-22-1Б.02.00.00	1
A4	4	Технологічна схема доменної печі та критичні зони	КІНТЕХАД.КН-22-1Б.03.00.00	1
A4	5	Діаграма послідовності подій при екстремому знеструмленні	КІНТЕХАД.КН-22-1Б.04.00.00	1
A4	6	Архітектура АСУ ТП доменного цеху ПРАТ «КАМЕТ-СТАЛЬ»	КІНТЕХАД.КН-22-1Б.05.00.00	1
A4	7	Математична модель теплопереносу та алгоритм оцінки ризику	КІНТЕХАД.КН-22-1Б.06.00.00	1
A4	8	Архітектура програмно-методичного комплексу (трирівнева)	КІНТЕХАД.КН-22-1Б.07.00.00	1
A4	9	UML-діаграми ПМК (прецедентів, діяльності, класів, розгортання)	КІНТЕХАД.КН-22-1Б.08.00.00	4
A4	10	Структура бази даних SQLite (ER-діаграма)	КІНТЕХАД.КН-22-1Б.09.00.00	1
A4	11	Дашборди Power BI (4 сторінки у 3 режимах)	КІНТЕХАД.КН-22-1Б.10.00.00	12
A4	12	Економічна ефективність впровадження ПМК	КІНТЕХАД.КН-22-1Б.11.00.00	1
A4	13	Висновки та перспективи розвитку	КІНТЕХАД.КН-22-1Б.12.00.00	1

ДОДАТОК Б. ТЕХНІЧНЕ ЗАВДАННЯ НА СТВОРЕННЯ ПМК

ЗАТВЕРДЖУЮ

Головний інженер ПРАТ «КАМЕТ-СТАЛЬ»

_____ (підпис)

«____» _____ 2026 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на створення програмно-методичного комплексу для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні

1. ЗАГАЛЬНІ ВІДОМОСТІ

1.1. Найменування системи: «Програмно-методичний комплекс для інформаційної підтримки запобігання термошоку доменних печей при екстремому знеструмленні».

1.2. Замовник: ПРАТ «КАМЕТ-СТАЛЬ» (доменний цех).

1.3. Розробник: студент кафедри інформаційних технологій та аналітики даних ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» Смик О.С.

1.4. Призначення: автоматизований моніторинг температурного стану футерівки доменної печі, прогнозування розвитку термошоку та інформаційна підтримка прийняття рішень персоналом при екстремому знеструмленні.

1.5. База впровадження: доменний цех ПРАТ «КАМЕТ-СТАЛЬ» (м. Кам'янське), доменна піч ДП-1М.

2. МЕТА ТА ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1. Мета: підвищення безпеки та надійності роботи доменної печі шляхом створення програмно-методичного комплексу для оперативного моніторингу, прогнозування та інформаційної підтримки запобігання термошоку футерівки при екстремому знеструмленні.

2.2. Кількісні цілі:

- зменшення ймовірності виникнення термошоку футерівки на 40 - 60%;
- скорочення часу прийняття рішень персоналом у перші критичні 10 - 15 хвилин;
- зменшення тривалості простою печі після аварійного знеструмлення на 20 - 40%.

3. ФУНКЦІОНАЛЬНІ МОДУЛІ СИСТЕМИ

3.1. Модуль збору даних:

- інтеграція з існуючою АСУ ТП через протокол OPC UA;
- отримання даних з термопар, манометрів, витратомірів;
- частота опитування датчиків - не менше 1 разу на 10 секунд;
- збереження даних у базі даних SQLite.

3.2. Модуль реального часу моніторингу:

- візуалізація температурного поля футерівки у вигляді теплових карт;
- відображення поточних значень температури в трьох зонах (колошник, середина, горн);
- індикація статусу живлення, тиску охолодження та градієнтів.

3.3. Модуль прогнозування термошоку:

- реалізація математичної моделі теплопереносу;
- прогнозування температур на 30 хвилин вперед;
- розрахунок вертикального та горизонтального градієнтів;
- час розрахунку прогнозу - не більше 1 секунди.

3.4. Модуль підтримки прийняття рішень:

- бальна оцінка ризику за шістьма факторами;
- визначення рівня ризику (5 рівнів: Низький, Помірний, Середній, Високий, Критичний);
- автоматична генерація текстових рекомендацій для персоналу;
- пріоритетизація рекомендацій (A, B, C, D).

3.5. Модуль оповіщення та інтерфейсу:

- чотири інтерактивні дашборди Power BI;
- кольорове кодування рівнів ризику;
- автоматична генерація тривог при високому та критичному ризику;
- консольне меню для керування сценаріями роботи.

3.6. Модуль логування та архівації:

- збереження всіх вимірювань у таблиці sensor_data;
- збереження прогнозів та оцінок ризику у таблиці predictions;
- збереження тривог у таблиці alerts;
- можливість перегляду історії даних за останні 100 записів.

4. ВИМОГИ ДО АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Сервер обробки даних:

- процесор: Intel Xeon 4+ ядра;
- оперативна пам'ять: 16+ ГБ;
- диск: SSD 256+ ГБ;
- операційна система: Ubuntu Server 22.04 LTS або Windows Server 2019.

4.2. Edge-пристрій:

- промисловий міні-ПК або PLC з підтримкою OPC UA;
- автономне живлення (акумулятор) не менше 2 годин.

4.3. АРМ оператора:

- робоча станція з монітором, клавіатурою, мишею;
- операційна система: Windows 10/11;
- встановлений Power BI Desktop (безкоштовна версія).

4.4. Резервне живлення:

- ДГУ (дизель-генераторна установка) з часом автономної роботи не менше 4 годин;
- акумуляторні батареї для Edge-пристрою (не менше 2 годин).

5. ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1. Backend:

- мова програмування: Python 3.10+;
- веб-фреймворк: FastAPI;
- база даних: SQLite;
- додаткові бібліотеки: uvicorn, requests, datetime, random, typing.

5.2. Frontend:

- інструмент візуалізації: Power BI Desktop (безкоштовна версія);
- підключення до даних: через REST API (ендпоінти /data-for-powerbi, /history, /alerts).

5.3. Система контролю версій:

- Git + GitHub (або інший репозиторій).

6. ВИМОГИ ДО ІНТЕРФЕЙСУ

6.1. Дашборди Power BI (4 сторінки):

Сторінка 1: «Загальний моніторинг» – картки температур, тиску, рівня ризику, графік динаміки температур.

Сторінка 2: «Теплова карта печі» – датчики градієнтів, стовпчаста діаграма розподілу температур по зонах.

Сторінка 3: «Прогноз та ризик» – графік прогнозу, картки ризику та рекомендацій.

Сторінка 4: «Рекомендації та тривоги» – таблиця тривог, графік історії ризику.

6.2. Кольорове кодування:

- Зелений – норма (Низький ризик);
- Жовтий – увага (Помірний ризик);
- Помаранчевий – потрібні дії (Середній ризик);
- Червоний – аварійна ситуація (Високий ризик);
- Чорний – термінові дії (Критичний ризик).

6.3. Консольне меню (menu.py):

- керування сценаріями роботи (Normal, Fluctuation, Blackout);
- зворотний зв'язок про зміну сценарію.

7. ВИМОГИ ДО НАДІЙНОСТІ ТА БЕЗПЕКИ

7.1. Резервування:

- дублювання сервера (основний + резервний);
- автоматичне перемикавання на резервний сервер при збої.

7.2. Захист даних:

- регулярне резервне копіювання (щогодинне);
- зберігання резервних копій на окремому носії.

7.3. Час відновлення:

- після збою – не більше 5 хвилин.

7.4. Безпека:

- автентифікація користувачів (за потреби);
- розмежування прав доступу (оператор, майстер).

8. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

8.1. Тестування:

- тестування в симульованому режимі (3 сценарії: Normal, Fluctuation, Blackout);
- перевірка коректності прогнозування та оцінки ризику.

8.2. Дослідна експлуатація:

- на ДП-1М протягом 30 днів;
- збір даних про роботу системи.

8.3. Критерії приймання:

- середньоквадратична похибка прогнозування температур $< 10^{\circ}\text{C}$;
- час обробки одного циклу вимірювань < 1 секунди;
- відсутність помилкових тривог протягом 30 днів.
- Склад документації, що передається замовнику:
- Пояснювальна записка до кваліфікаційної роботи.
- Вихідні коди програмного забезпечення (main.py, menu.py).
- Інструкція користувача (коротка).
- Результати тестування (протоколи).
- Дашборди Power BI у форматі .pbix.

ПОГОДЖЕНО:

Керівник практики від підприємства:

_____ (підпис) / Д.А. Шпунт /

Розробник:

_____ (підпис) / О.С. Смик /

ДОДАТОК В. UML-ДІАГРАМИ ПРОГРАМНО-МЕТОДИЧНОГО КОМПЛЕКСУ

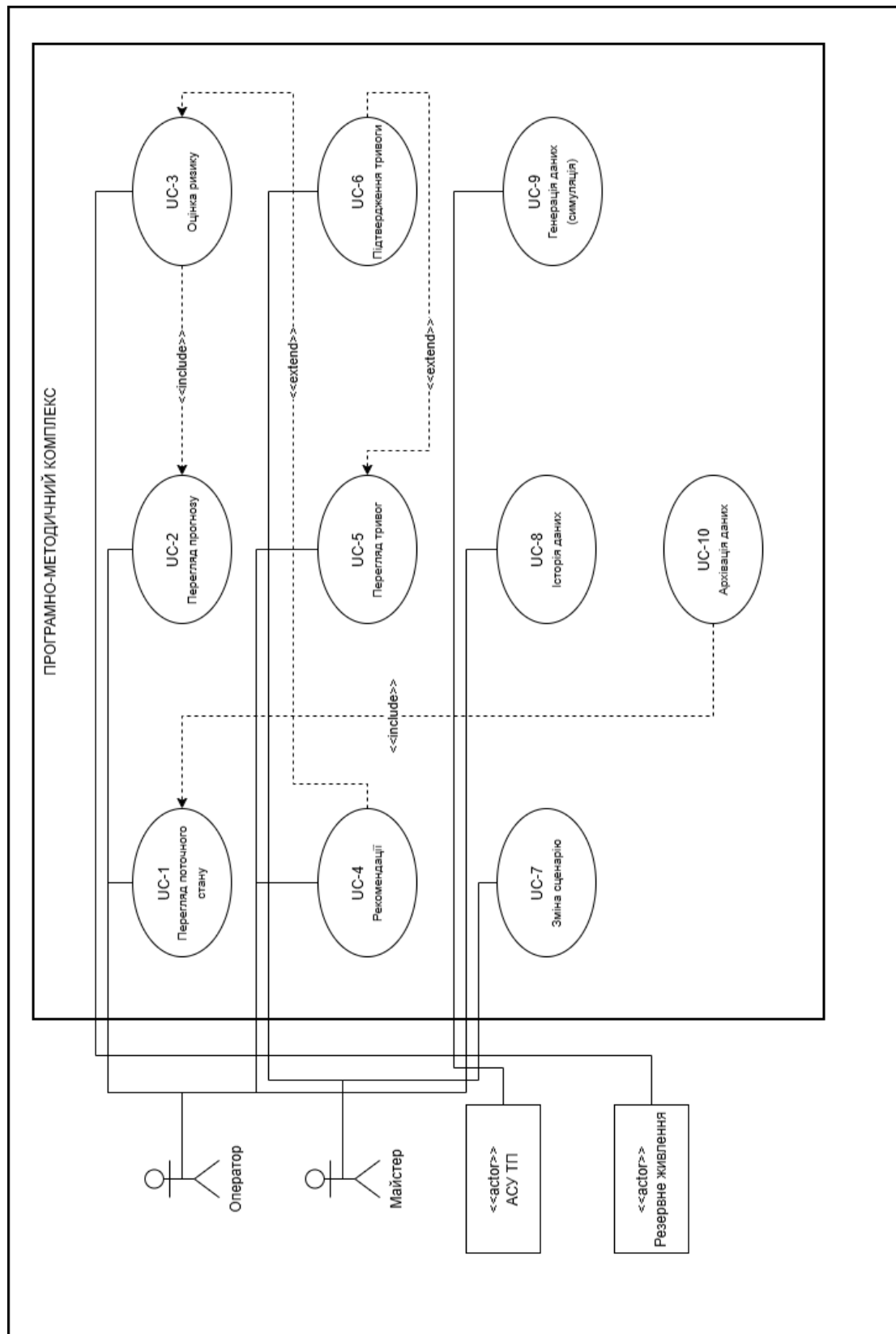


Рисунок В.1 – Діаграма прецедентів використання ПМК

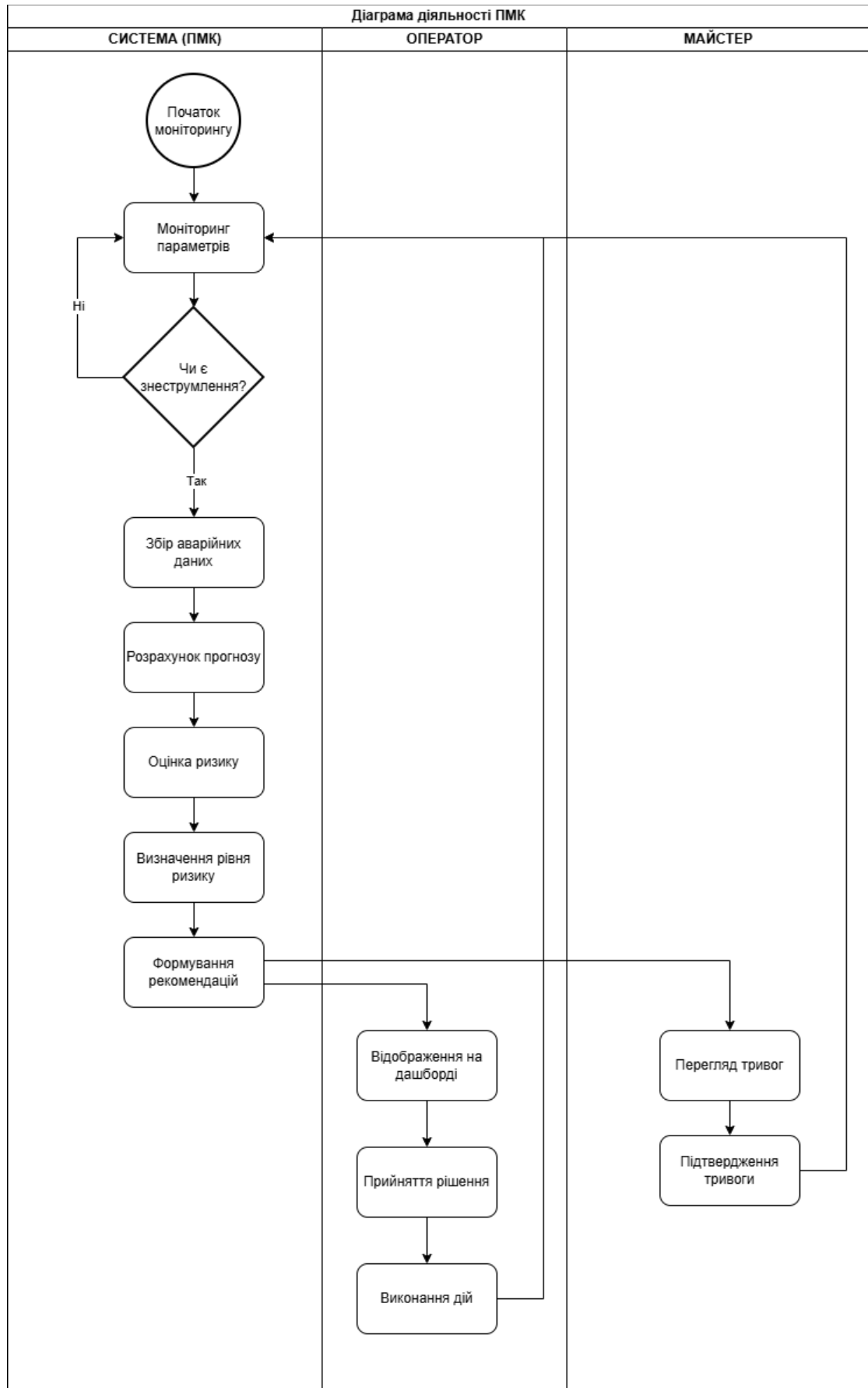


Рисунок В.2 – Діаграма діяльності ПМК

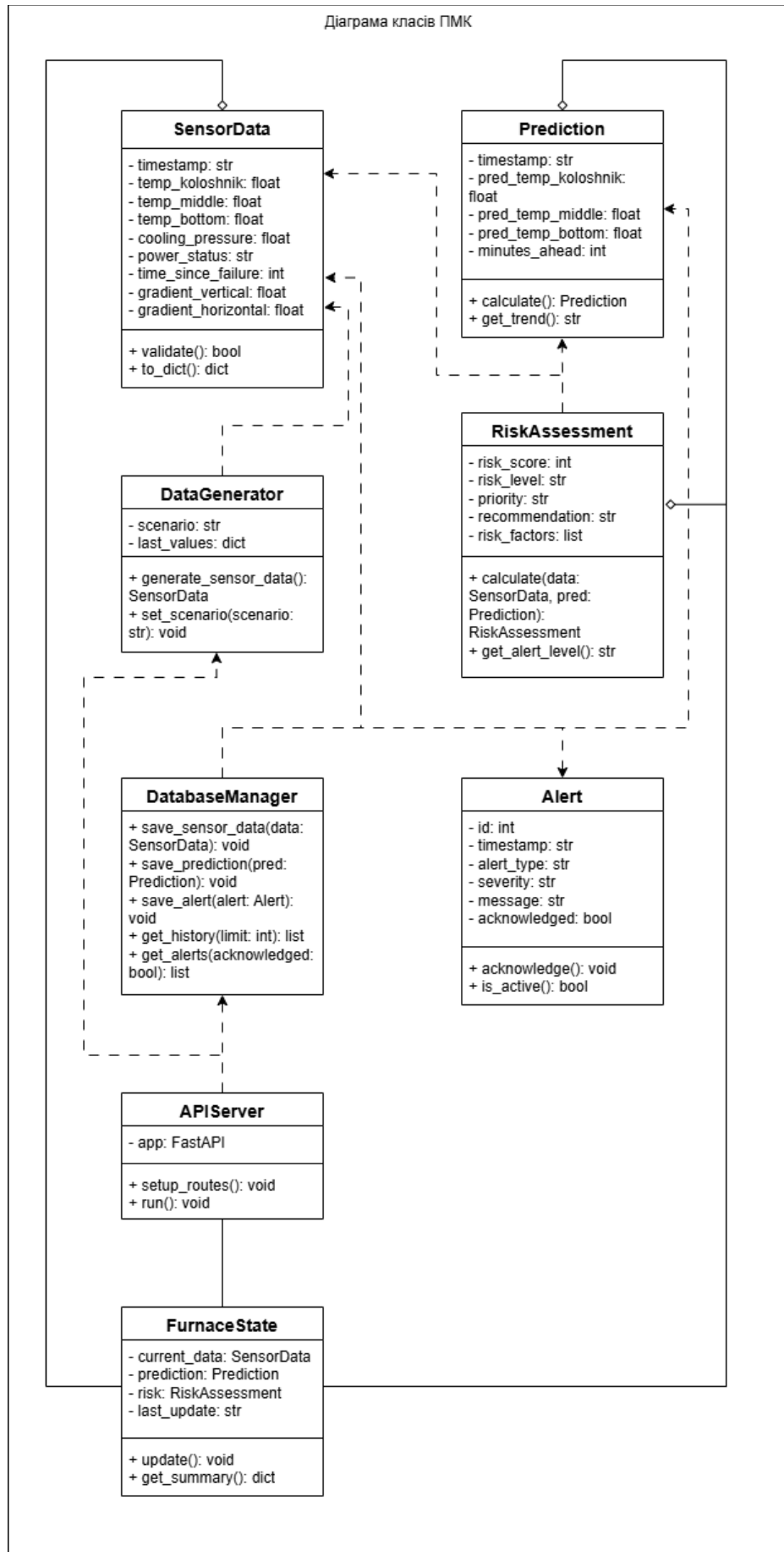


Рисунок В.3 – Діаграма класів ПМК

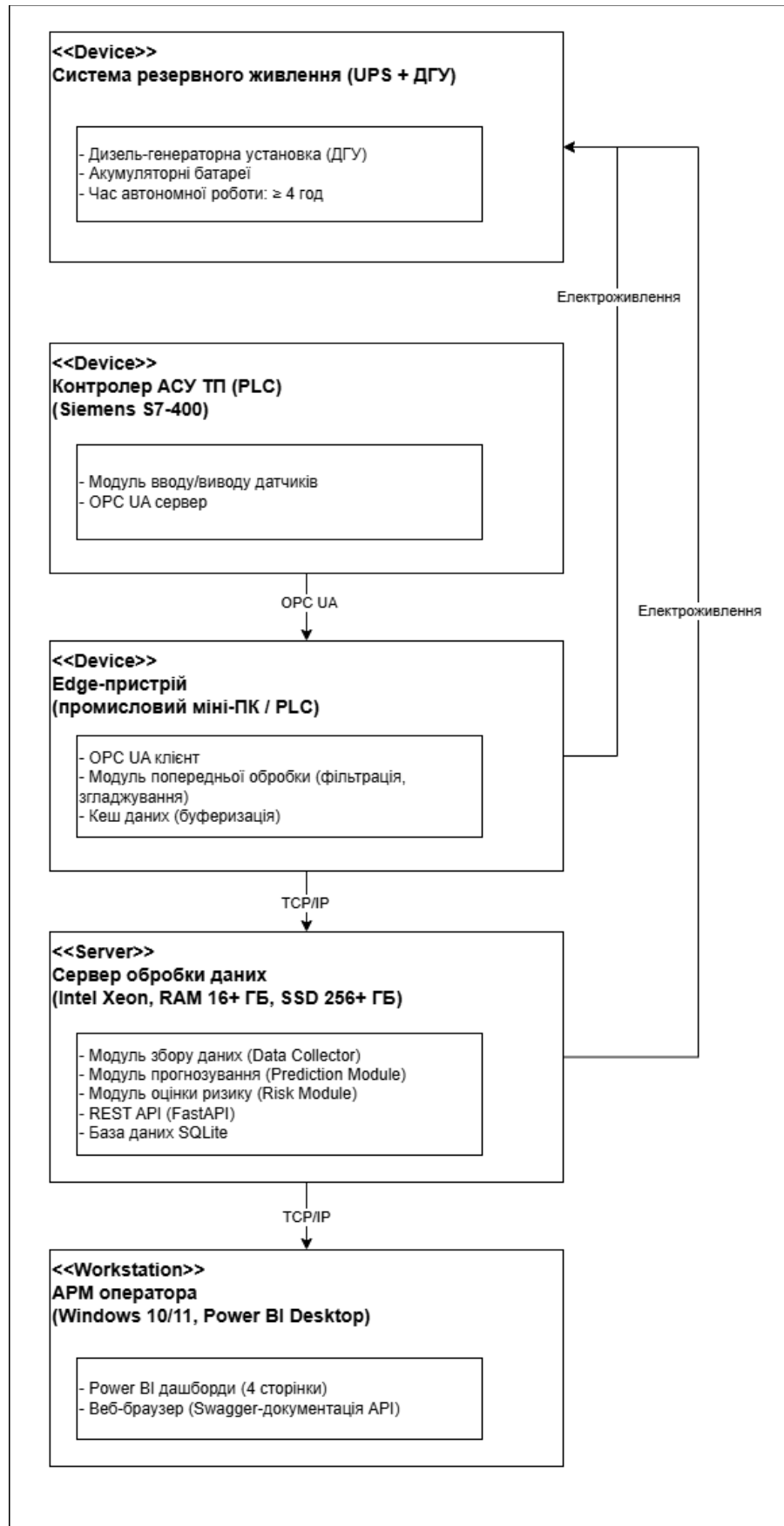


Рисунок В.4 – Діаграма розгортання ПМК

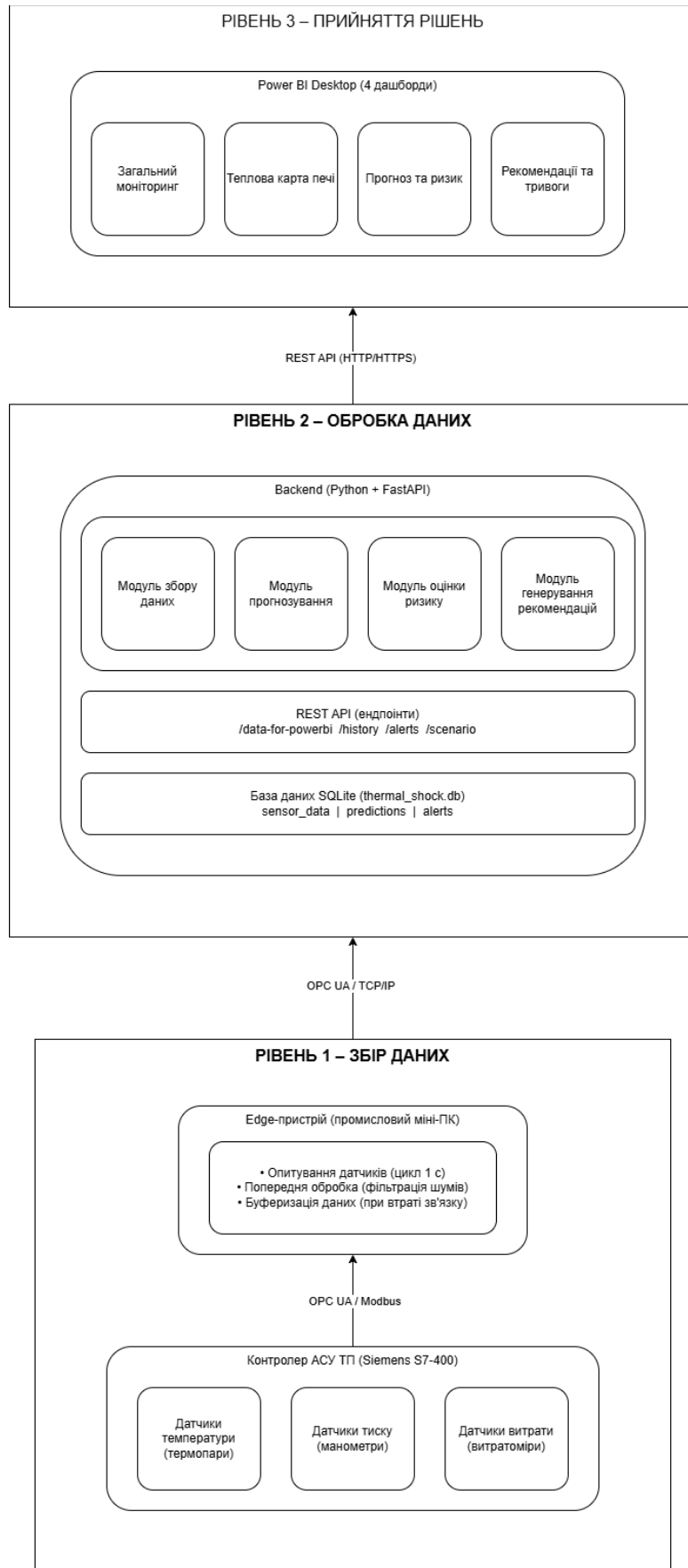


Рисунок В.5 – Тривірнева архітектура ПМК

ДОДАТОК Г. ЛІСТИНГИ ПРОГРАМНОГО КОДУ

ЛІСТИНГ Г.1 – main.py

```

from fastapi import FastAPI
from datetime import datetime
import random
import sqlite3
from typing import Dict, Any

app = FastAPI(title="ПМК Запобігання Термошоку ДП")

# ===== БАЗА ДАНИХ =====
def init_database():
    conn = sqlite3.connect('thermal_shock.db')
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS sensor_data (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            timestamp TEXT,
            temp_koloshnik REAL,
            temp_middle REAL,
            temp_bottom REAL,
            cooling_pressure REAL,
            power_status TEXT,
            time_since_failure INTEGER,
            gradient_vertical REAL,
            gradient_horizontal REAL
        )
    """)
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS predictions (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            timestamp TEXT,
            pred_temp_koloshnik REAL,
            pred_temp_middle REAL,
            pred_temp_bottom REAL,
            risk_score INTEGER,
            risk_level TEXT,
            recommendation TEXT
        )
    """)
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS alerts (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            timestamp TEXT,
            alert_type TEXT,
            severity TEXT,
            message TEXT,
            acknowledged BOOLEAN DEFAULT 0
        )
    """)
    conn.commit()
    conn.close()

def save_to_database(data: Dict[str, Any]):
    conn = sqlite3.connect('thermal_shock.db')
    cursor = conn.cursor()
    cursor.execute("""

```

```

INSERT INTO sensor_data
(timestamp, temp_koloshnik, temp_middle, temp_bottom, cooling_pressure,
power_status, time_since_failure, gradient_vertical, gradient_horizontal)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
", (
    data["timestamp"],
    data["temp_koloshnik"],
    data["temp_middle"],
    data["temp_bottom"],
    data["cooling_pressure"],
    data["power_status"],
    data["time_since_failure"],
    data["gradient_vertical"],
    data["gradient_horizontal"]
))
conn.commit()
conn.close()

def save_prediction(pred: Dict[str, Any]):
    conn = sqlite3.connect('thermal_shock.db')
    cursor = conn.cursor()
    cursor.execute("""
        INSERT INTO predictions
        (timestamp, pred_temp_koloshnik, pred_temp_middle, pred_temp_bottom,
        risk_score, risk_level, recommendation)
        VALUES (?, ?, ?, ?, ?, ?, ?)
    """, (
        pred["timestamp"],
        pred["pred_temp_koloshnik"],
        pred["pred_temp_middle"],
        pred["pred_temp_bottom"],
        pred["risk_score"],
        pred["risk_level"],
        pred["recommendation"]
    ))
    conn.commit()
    conn.close()

def save_alert(alert_type: str, severity: str, message: str):
    conn = sqlite3.connect('thermal_shock.db')
    cursor = conn.cursor()
    cursor.execute("""
        INSERT INTO alerts (timestamp, alert_type, severity, message)
        VALUES (?, ?, ?, ?)
    """, (datetime.now().strftime("%Y-%m-%d %H:%M:%S"), alert_type, severity, message))
    conn.commit()
    conn.close()

init_database()

# ===== ЦІЛЕВАПІЇ =====
current_scenario = "normal"

_last_values = {
    "temp_koloshnik": 600.0,
    "temp_middle": 950.0,
    "temp_bottom": 1380.0,
    "cooling_pressure": 3.5
}

# Цільові значення для плавного повернення до норми
_target_values = {

```

```

"temp_koloshnik": 650.0,
"temp_middle": 1000.0,
"temp_bottom": 1400.0,
"cooling_pressure": 3.8
}

def generate_sensor_data() -> Dict[str, Any]:
    global _last_values, current_scenario, _target_values

    # =====
    # СЦЕНАРІЙ 1: ШТАТНА РОБОТА (з плавним відновленням)
    # =====
    if current_scenario == "normal":
        recovery_speed = 2.0 # градусів за крок (швидкість нагрівання)

        # Колошник
        if _last_values["temp_koloshnik"] < _target_values["temp_koloshnik"]:
            new_koloshnik = _last_values["temp_koloshnik"] + recovery_speed
        else:
            new_koloshnik = _last_values["temp_koloshnik"] + random.uniform(-3, 3)

        # Середина
        if _last_values["temp_middle"] < _target_values["temp_middle"]:
            new_middle = _last_values["temp_middle"] + recovery_speed * 0.8
        else:
            new_middle = _last_values["temp_middle"] + random.uniform(-2, 2)

        # Низ
        if _last_values["temp_bottom"] < _target_values["temp_bottom"]:
            new_bottom = _last_values["temp_bottom"] + recovery_speed * 0.6
        else:
            new_bottom = _last_values["temp_bottom"] + random.uniform(-1, 1)

        # Тиск охолодження
        if _last_values["cooling_pressure"] < _target_values["cooling_pressure"]:
            new_pressure = _last_values["cooling_pressure"] + 0.05
        else:
            new_pressure = _last_values["cooling_pressure"] + random.uniform(-0.03, 0.03)

        # Обмежуємо, щоб не вилізли за межі норми
        new_koloshnik = min(_target_values["temp_koloshnik"] + 30, new_koloshnik)
        new_koloshnik = max(550, new_koloshnik)

        new_middle = min(_target_values["temp_middle"] + 30, new_middle)
        new_middle = max(850, new_middle)

        new_bottom = min(_target_values["temp_bottom"] + 30, new_bottom)
        new_bottom = max(1350, new_bottom)

        new_pressure = max(3.5, min(5.2, new_pressure))

        power_status = "normal"
        time_since_failure = 0

    # =====
    # СЦЕНАРІЙ 2: КОЛИВАННЯ ЖИВЛЕННЯ
    # =====
    elif current_scenario == "fluctuation":
        new_koloshnik = _last_values["temp_koloshnik"] + random.uniform(-15, 15)
        new_middle = _last_values["temp_middle"] + random.uniform(-10, 10)
        new_bottom = _last_values["temp_bottom"] + random.uniform(-8, 8)
        new_pressure = _last_values["cooling_pressure"] + random.uniform(-0.3, 0.3)

```

```

new_koloshnik = max(480, min(780, new_koloshnik))
new_middle = max(750, min(1150, new_middle))
new_bottom = max(1250, min(1480, new_bottom))
new_pressure = max(2.5, min(5.2, new_pressure))

power_status = "emergency" if random.random() < 0.3 else "normal"
time_since_failure = random.randint(0, 15) if power_status == "emergency" else 0

# =====
# СЦЕНАРІЙ 3: ПОВНЕ ВІДКЛЮЧЕННЯ
# =====
elif current_scenario == "blackout":
    new_koloshnik = _last_values["temp_koloshnik"] - random.uniform(3, 8)
    new_middle = _last_values["temp_middle"] - random.uniform(2, 5)
    new_bottom = _last_values["temp_bottom"] - random.uniform(1, 3)
    new_pressure = _last_values["cooling_pressure"] - random.uniform(0.02, 0.08)

    new_koloshnik = max(420, new_koloshnik)
    new_middle = max(750, new_middle)
    new_bottom = max(1250, new_bottom)
    new_pressure = max(2.0, new_pressure)

    power_status = "emergency"
    time_since_failure = random.randint(10, 25)

# =====
# ЗАПАСНИЙ ВАРІАНТ
# =====
else:
    new_koloshnik = _last_values["temp_koloshnik"]
    new_middle = _last_values["temp_middle"]
    new_bottom = _last_values["temp_bottom"]
    new_pressure = _last_values["cooling_pressure"]
    power_status = "normal"
    time_since_failure = 0

# Зберігаємо нові значення
_last_values["temp_koloshnik"] = new_koloshnik
_last_values["temp_middle"] = new_middle
_last_values["temp_bottom"] = new_bottom
_last_values["cooling_pressure"] = new_pressure

data = {
    "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    "temp_koloshnik": round(new_koloshnik, 1),
    "temp_middle": round(new_middle, 1),
    "temp_bottom": round(new_bottom, 1),
    "cooling_pressure": round(new_pressure, 2),
    "power_status": power_status,
    "time_since_failure": time_since_failure,
}

data["gradient_vertical"] = round((data["temp_bottom"] - data["temp_koloshnik"]) / 2, 1)
data["gradient_horizontal"] = round(random.uniform(-5, 5), 1)

save_to_database(data)

return data

# ===== ПРОГНОЗУВАННЯ =====
def predict_temperature(current_data: Dict[str, Any], minutes_ahead: int = 30) -> Dict[str, Any]:

```

```
alpha_koloshnik = 0.95
alpha_middle = 0.85
alpha_bottom = 0.75
```

```
cooling_factor = current_data["cooling_pressure"] / 3.5
```

```
pred_koloshnik = current_data["temp_koloshnik"] - (alpha_koloshnik * cooling_factor * 0.1 * minutes_ahead / 15)
pred_middle = current_data["temp_middle"] - (alpha_middle * cooling_factor * 0.08 * minutes_ahead / 15)
pred_bottom = current_data["temp_bottom"] - (alpha_bottom * cooling_factor * 0.06 * minutes_ahead / 15)
```

```
if current_data["power_status"] == "emergency":
    pred_koloshnik -= 10 * (minutes_ahead / 15)
    pred_middle -= 8 * (minutes_ahead / 15)
    pred_bottom -= 5 * (minutes_ahead / 15)
```

```
return {
    "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
    "pred_temp_koloshnik": round(pred_koloshnik, 1),
    "pred_temp_middle": round(pred_middle, 1),
    "pred_temp_bottom": round(pred_bottom, 1),
    "minutes_ahead": minutes_ahead
}
```

```
# ===== ОЦІНКА РИЗИКУ =====
```

```
def calculate_risk(data: Dict[str, Any], prediction: Dict[str, Any]) -> Dict[str, Any]:
```

```
    risk_score = 0
    risk_factors = []
```

```
    if data["power_status"] == "emergency":
        risk_score += 35
        risk_factors.append({"factor": "Відмова живлення", "weight": 35})
```

```
    if data["cooling_pressure"] < 2.5:
        risk_score += 30
        risk_factors.append({"factor": "Критичний тиск охолодження", "weight": 30})
    elif data["cooling_pressure"] < 3.0:
        risk_score += 20
        risk_factors.append({"factor": "Низький тиск охолодження", "weight": 20})
```

```
    if data["temp_koloshnik"] < 500:
        risk_score += 20
        risk_factors.append({"factor": "Переохолодження колошника", "weight": 20})
    elif data["temp_koloshnik"] < 550:
        risk_score += 10
        risk_factors.append({"factor": "Зниження температури колошника", "weight": 10})
```

```
    if data["gradient_vertical"] > 500:
        risk_score += 15
        risk_factors.append({"factor": "Високий вертикальний градієнт", "weight": 15})
    elif data["gradient_vertical"] > 400:
        risk_score += 8
        risk_factors.append({"factor": "Підвищений вертикальний градієнт", "weight": 8})
```

```
    temp_drop = data["temp_koloshnik"] - prediction["pred_temp_koloshnik"]
```

```
    if temp_drop > 30:
        risk_score += 20
        risk_factors.append({"factor": f"Прогнозоване падіння на {round(temp_drop)}°C", "weight": 20})
    elif temp_drop > 15:
        risk_score += 10
        risk_factors.append({"factor": f"Прогнозоване падіння на {round(temp_drop)}°C", "weight": 10})
```

```
    if data["time_since_failure"] > 20:
```

```

    risk_score += 15
    risk_factors.append({"factor": "Тривалий час після відмови", "weight": 15})
elif data["time_since_failure"] > 10:
    risk_score += 8
    risk_factors.append({"factor": "Значний час після відмови", "weight": 8})

if risk_score >= 80:
    risk_level = "Критичний"
    priority = "A"
    recommendation = "Негайний випуск чавуну. Вимкнути нагрів, максимальне охолодження."
elif risk_score >= 60:
    risk_level = "Високий"
    priority = "A"
    recommendation = "Аварійна ситуація! Збільшити тиск охолодження. Підготувати резервне живлення."
elif risk_score >= 40:
    risk_level = "Середній"
    priority = "B"
    recommendation = "Підвищена увага! Знизити навантаження. Перевірити систему охолодження."
elif risk_score >= 20:
    risk_level = "Помірний"
    priority = "C"
    recommendation = "Посилений моніторинг. Перевірити градієнти температури."
else:
    risk_level = "Низький"
    priority = "D"
    recommendation = "Штатний режим. Продовжувати моніторинг."

if risk_level in ["Високий", "Критичний"]:
    save_alert(
        alert_type="THERMAL_SHOCK_RISK",
        severity=priority,
        message="Рівень ризику: {risk_level}. {recommendation[:50]}..."
    )

return {
    "risk_score": risk_score,
    "risk_level": risk_level,
    "priority": priority,
    "recommendation": recommendation,
    "risk_factors": risk_factors,
    "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
}

# ===== API ЕНДПОІНТИ =====

@app.get("/")
def home():
    return {
        "status": "ПМК Запобігання Термошоку ДП працює",
        "version": "2.2",
        "current_scenario": current_scenario,
        "modules": [
            "Збір даних",
            "Моніторинг реального часу",
            "Прогнозування теплопереносу",
            "Оцінка ризику (5 рівнів)",
            "Генерація рекомендацій",
            "Логування та архівація",
            "Оповіщення про тривоги"
        ],
        "timestamp": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    }

```

```

@app.get("/scenario")
def get_scenario():
    return {"current_scenario": current_scenario}

@app.post("/scenario/{scenario_name}")
def set_scenario(scenario_name: str):
    global current_scenario
    if scenario_name in ["normal", "fluctuation", "blackout"]:
        current_scenario = scenario_name
        return {"status": "ok", "current_scenario": current_scenario}
    return {"status": "error", "message": "Invalid scenario. Use: normal, fluctuation, blackout"}

@app.get("/data-for-powerbi")
def data_for_powerbi():
    current_data = generate_sensor_data()
    prediction = predict_temperature(current_data, 30)
    risk = calculate_risk(current_data, prediction)

    timestamp = datetime.now()
    date_str = timestamp.strftime("%Y-%m-%d")
    time_str = timestamp.strftime("%H:%M:%S")

    return {
        "date": date_str,
        "time": time_str,
        "temp_koloshnik": current_data["temp_koloshnik"],
        "temp_middle": current_data["temp_middle"],
        "temp_bottom": current_data["temp_bottom"],
        "cooling_pressure": current_data["cooling_pressure"],
        "power_status": current_data["power_status"],
        "time_since_failure": current_data["time_since_failure"],
        "gradient_vertical": current_data["gradient_vertical"],
        "gradient_horizontal": current_data["gradient_horizontal"],
        "pred_temp_koloshnik_30min": prediction["pred_temp_koloshnik"],
        "pred_temp_middle_30min": prediction["pred_temp_middle"],
        "pred_temp_bottom_30min": prediction["pred_temp_bottom"],
        "risk_score": risk["risk_score"],
        "risk_level": risk["risk_level"],
        "priority": risk["priority"],
        "recommendation": risk["recommendation"],
        "current_scenario": current_scenario
    }

@app.get("/history")
def get_history(limit: int = 100):
    conn = sqlite3.connect('thermal_shock.db')
    cursor = conn.cursor()
    cursor.execute("""
        SELECT * FROM sensor_data
        ORDER BY id DESC LIMIT ?
    """, (limit,))
    rows = cursor.fetchall()
    conn.close()

    history = []
    for row in rows:
        history.append({
            "id": row[0],
            "timestamp": row[1],
            "temp_koloshnik": row[2],
            "temp_middle": row[3],

```

```

        "temp_bottom": row[4],
        "cooling_pressure": row[5],
        "power_status": row[6],
        "time_since_failure": row[7],
        "gradient_vertical": row[8],
        "gradient_horizontal": row[9]
    })
    return history

@app.get("/alerts")
def get_alerts(acknowledged: bool = False):
    conn = sqlite3.connect('thermal_shock.db')
    cursor = conn.cursor()
    cursor.execute("""
        SELECT * FROM alerts
        WHERE acknowledged = ?
        ORDER BY id DESC
    """, (1 if acknowledged else 0,))
    rows = cursor.fetchall()
    conn.close()

    alerts = []
    for row in rows:
        alerts.append({
            "id": row[0],
            "timestamp": row[1],
            "alert_type": row[2],
            "severity": row[3],
            "message": row[4],
            "acknowledged": bool(row[5])
        })
    return alerts

@app.post("/acknowledge-alert/{alert_id}")
def acknowledge_alert(alert_id: int):
    conn = sqlite3.connect('thermal_shock.db')
    cursor = conn.cursor()
    cursor.execute("""
        UPDATE alerts SET acknowledged = 1
        WHERE id = ?
    """, (alert_id,))
    conn.commit()
    conn.close()
    return {"status": "Alert acknowledged", "alert_id": alert_id}

```

ЛИСТИНГ Г.2 – menu.py

```

import requests
import sys

API_URL = "http://127.0.0.1:8000"

def scenario_selector():
    # Примусово виводимо меню одразу
    sys.stdout.write("\n" + "="*60 + "\n")
    sys.stdout.write("ПМК ЗАПОБІГАННЯ ТЕРМОШОКУ - КОНСОЛЬНЕ КЕРУВАННЯ\n")
    sys.stdout.write("="*60 + "\n")
    sys.stdout.write("Оберіть сценарій роботи печі:\n")
    sys.stdout.write(" 1 - Штатна робота\n")
    sys.stdout.write(" 2 - Коливання живлення\n")
    sys.stdout.write(" 3 - Повне відключення\n")
    sys.stdout.write(" 4 - Вихід\n")

```

```

sys.stdout.write("-"*60 + "\n")
sys.stdout.flush()

while True:
    try:
        choice = input("Ваш вибір (1-4): ").strip()

        if choice == "1":
            response = requests.post(f"{API_URL}/scenario/normal")
            if response.status_code == 200:
                sys.stdout.write("\n[OK] Сценарій змінено: ШТАТНА РОБОТА\n")
                sys.stdout.write("    Температури в межах норми, тиск стабільний.\n\n")
            else:
                sys.stdout.write(f"[ERROR] Помилка: {response.text}\n\n")
                sys.stdout.flush()

        elif choice == "2":
            response = requests.post(f"{API_URL}/scenario/fluctuation")
            if response.status_code == 200:
                sys.stdout.write("\n[OK] Сценарій змінено: КОЛИВАННЯ ЖИВЛЕННЯ\n")
                sys.stdout.write("    Температури нестабільні, тиск коливається.\n\n")
            else:
                sys.stdout.write(f"[ERROR] Помилка: {response.text}\n\n")
                sys.stdout.flush()

        elif choice == "3":
            response = requests.post(f"{API_URL}/scenario/blackout")
            if response.status_code == 200:
                sys.stdout.write("\n[OK] Сценарій змінено: ПОВНЕ ВІДКЛЮЧЕННЯ\n")
                sys.stdout.write("    Температура знижується, тиск охолодження падає.\n\n")
            else:
                sys.stdout.write(f"[ERROR] Помилка: {response.text}\n\n")
                sys.stdout.flush()

        elif choice == "4":
            sys.stdout.write("\n[INFO] Завершення роботи меню.\n")
            sys.stdout.flush()
            break

        else:
            sys.stdout.write("[ERROR] Некоректний вибір. Введіть 1, 2, 3 або 4.\n\n")
            sys.stdout.flush()

    except requests.exceptions.ConnectionError:
        sys.stdout.write("[ERROR] Сервер не запущено! Запустіть uvicorn main:app --reload у першому
терміналі.\n\n")
        sys.stdout.flush()
    except KeyboardInterrupt:
        sys.stdout.write("\n\n[INFO] Переривання користувачем...\n")
        sys.stdout.flush()
        break
    except Exception as e:
        sys.stdout.write(f"[ERROR] {e}\n\n")
        sys.stdout.flush()

if __name__ == "__main__":
    scenario_selector()

```

ДОДАТОК Д. СКРІНШОТИ ДАШБОРДІВ POWER BI

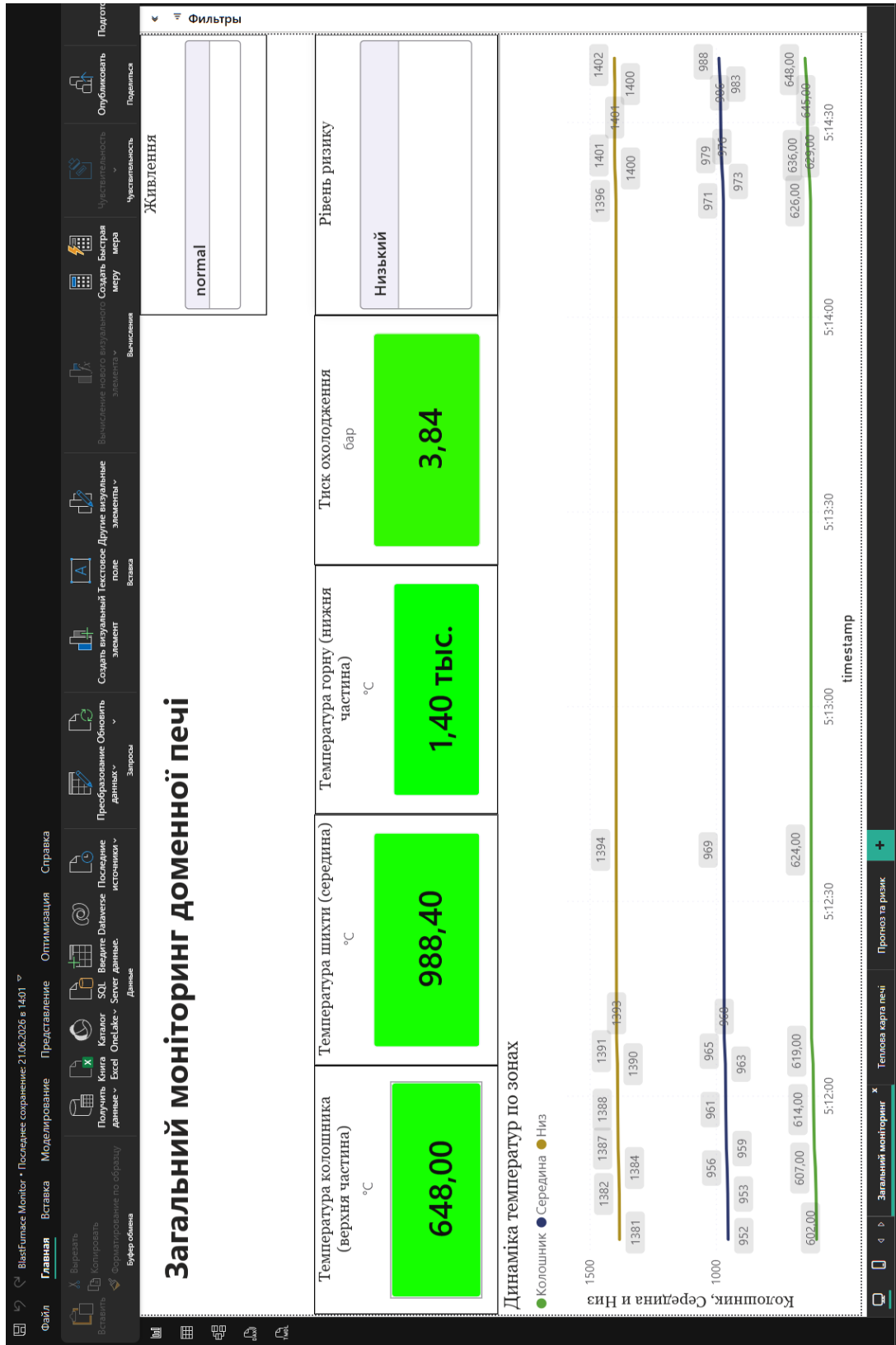


Рисунок Д.1 – Дашборд «Загальний моніторинг» у штатному режимі

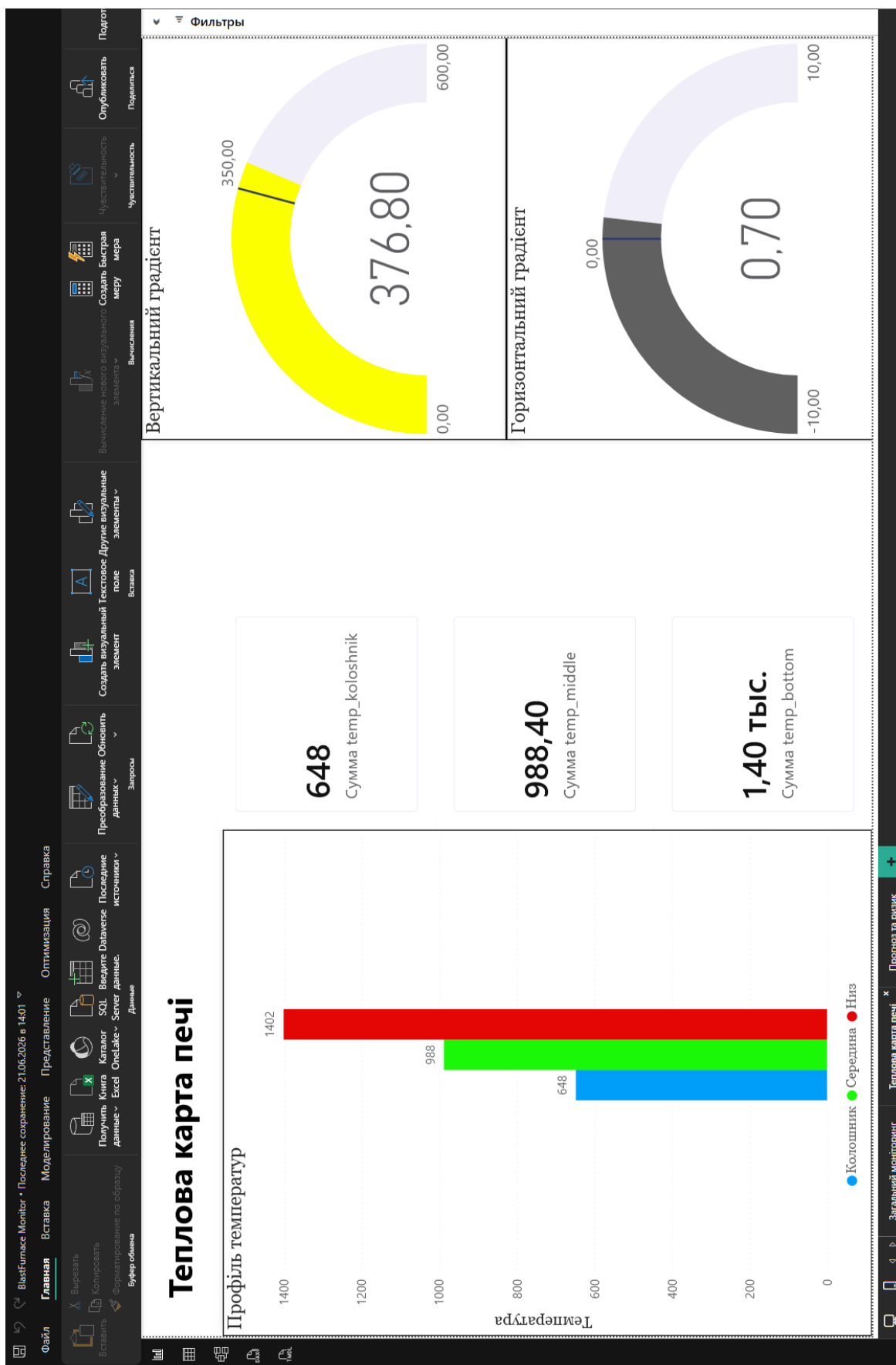


Рисунок Д.2 – Дашборд «Теплова карта печі» у штатному режимі



Рисунок Д.3 – Дашборд «Загальний моніторинг» у режимі коливань живлення

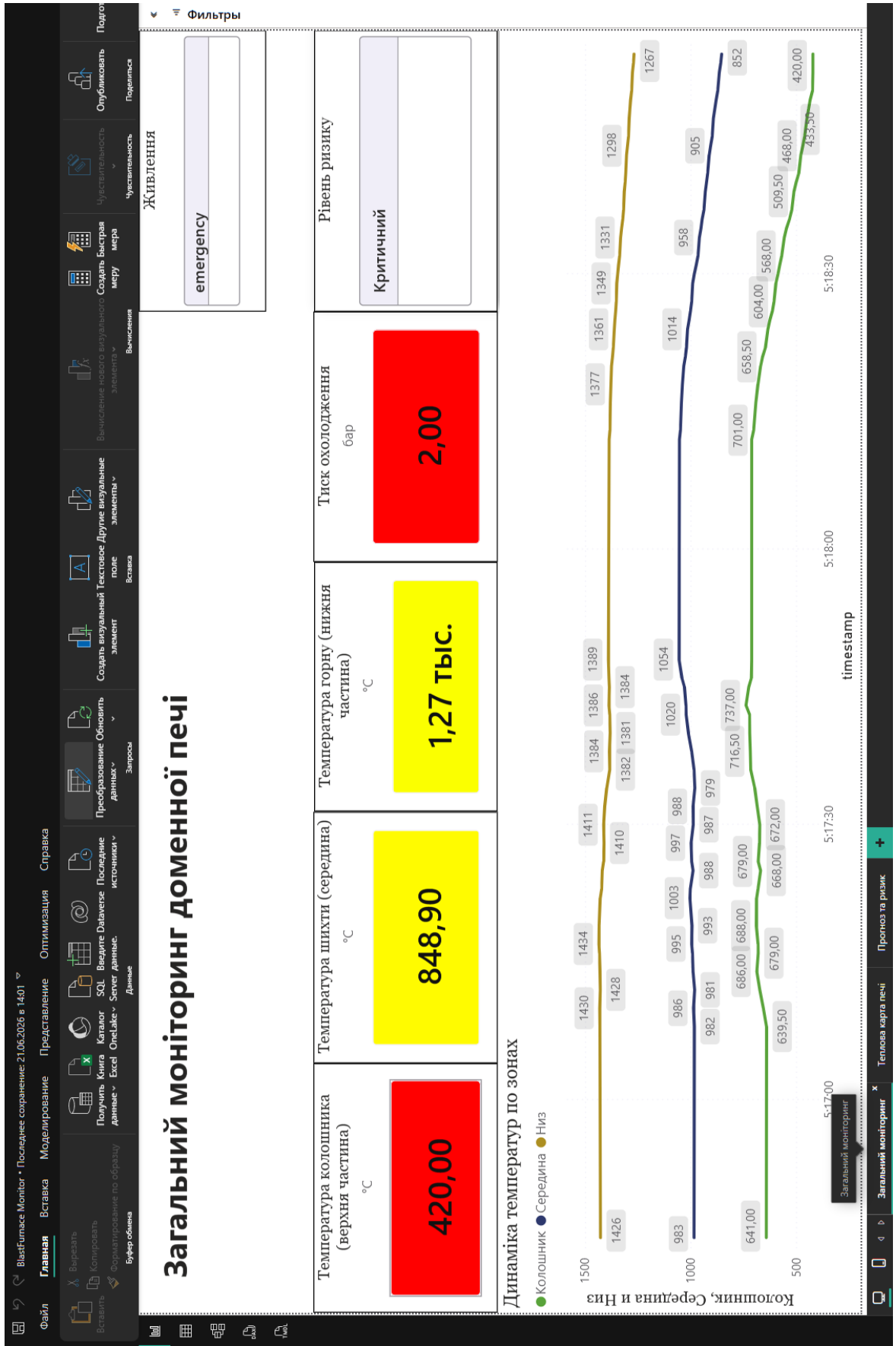


Рисунок Д.5 – Дашборд «Загальний моніторинг» в аварійному режимі (Blackout)

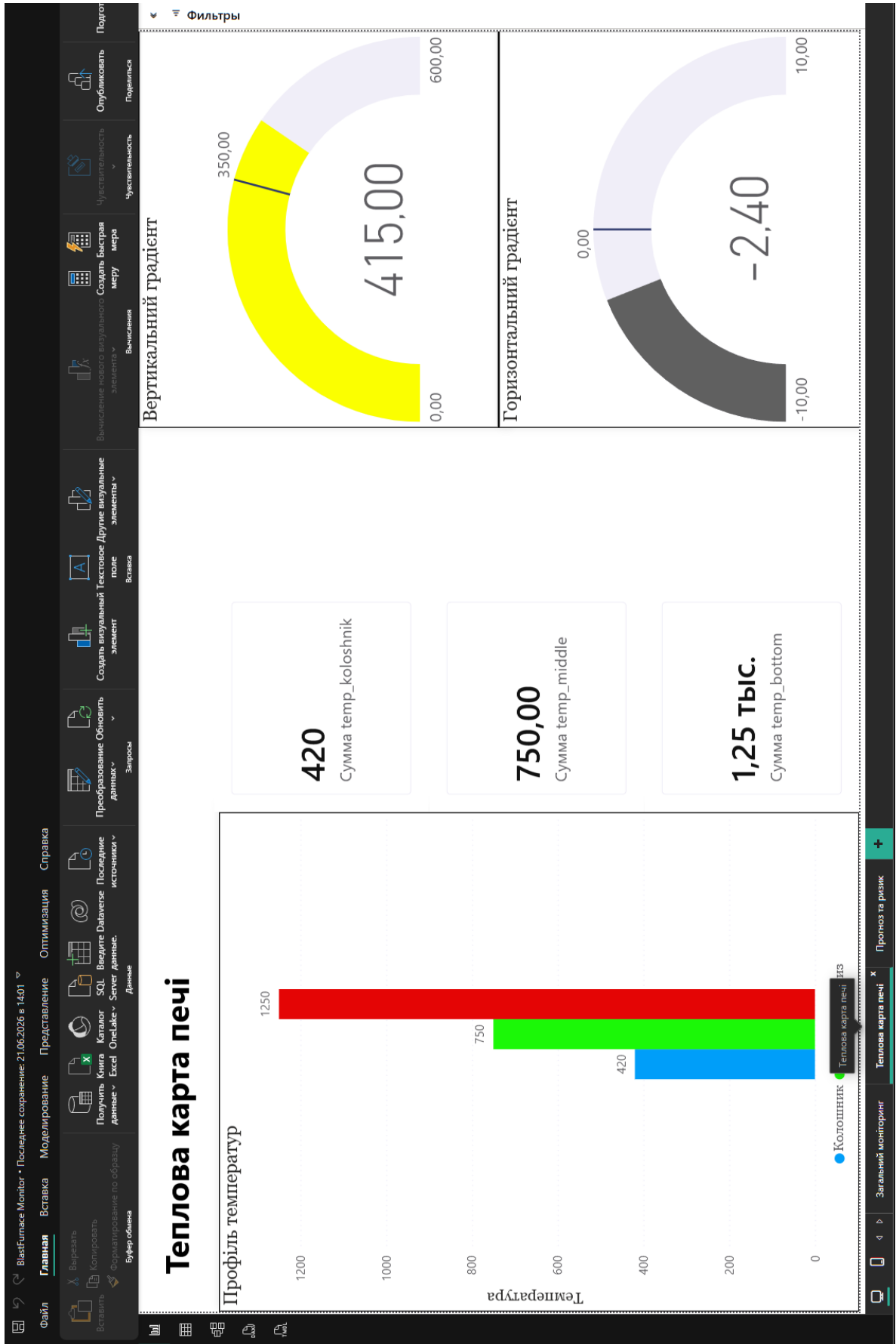


Рисунок Д.6 – Дашборд «Теплова карта печі» в аварійному режимі (Blackout)