

Львівський національний університет імені Івана Франка

Кваліфікаційна наукова
праця на правах рукопису

Сінькевич Олег Олександрович

УДК 004.4:004.62

ДИСЕРТАЦІЯ

**ОПТИМІЗАЦІЯ ФУНКЦІОНУВАННЯ
ІНТЕЛЕКТУАЛЬНИХ ОБ'ЄКТІВ З ВИКОРИСТАННЯМ
МЕТОДІВ МАШИННОГО НАВЧАННЯ**

122 — Комп'ютерні науки

12 — Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело ___ (О. О. Сінькевич)

Науковий керівник Монастирський Любомир Степанович,
доктор фізико-математичних наук, професор

Львів – 2023

АНОТАЦІЯ

Сінькевич О.О. Оптимізація функціонування інтелектуальних об'єктів з використанням методів машинного навчання. — Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки». — Львівський національний університет імені Івана Франка, Львів, 2023.

Проблеми оптимізації функціонування інтелектуальних об'єктів визначають широкий клас задач, який охоплює як реальні апаратні комплекси, що забезпечують виробничі процеси, так і відповідне їм програмне забезпечення. Програмування таких комплексів в контексті задання оптимальних режимів роботи потребує вирішення задач стабільної апаратно-програмної інтеграції та розроблення ефективних алгоритмів для їх функціонування. Еволюція наявних мікрокомп'ютерних та мікроконтролерних технологій, які становлять “розумну” основу більшості інтелектуальних об'єктів зумовлює потребу в постійному вдосконаленні вже існуючих програмних засобів та їх адаптації під нові платформи. При цьому, зі зростанням вимог до таких інтелектуальних об'єктів підвищується і складність щодо проєктування та реалізації програмного забезпечення.

Використання машинного навчання при створенні програмних засобів для інтелектуальних об'єктів стає все більш поширеною практикою, оскільки дозволяє реалізовувати гнучкі та різноманітні моделі прогнозування стану об'єкта, його динамічного самонавчання (навчання з підкріпленням) або даних, що дозволяють продуктивніше використовувати робочу систему. Прикладом такого роду інтелектуального об'єкту, функціонування якого можна забезпечувати та оптимізувати методами машинного навчання, є інтелектуальний будинок. У дисертації основна увага приділена розробці та удосконаленню

прототипу апаратно-програмного комплексу для аналізу та обробки даних інтелектуального будинку, в межах чого здійснена інтеграція мікрокомп'ютерної та нейроконтролерної систем з відкритим програмним забезпеченням, яке ґрунтується на використанні сучасних підходів глибокого навчання, техніки MLOPS і REST архітектури.

Проведений аналіз наукових публікацій та відкритих джерел у тематиці інтелектуальних будинків продемонстрував, що зараз існує недостатньо рішень щодо апаратно-програмних систем, функціонування яких не залежить від хмарних технологій та доступу до глобальної мережі Інтернет. Також, не у повній мірі розкриті прикладні архітектурні концепції із залученням моделей глибокого навчання та граничних (edge) чи туманних (fog) обчислень, розподілених мікроконтролерних систем, які підтримують вбудовування й розгортання згаданих моделей. У зв'язку із цим завданням дисертаційної роботи є доповнення та розширення знань у предметній області проектування прототипу мікроконтролерної системи обробки даних інтелектуального будинку в контексті граничних обчислень у поєднанні з машинним навчанням, статистичними методами та практиками побудови прикладних програмних інтерфейсів (API).

Оскільки у більшості випадків функціонування інтелектуального будинку стосується оптимізації енерговитрат та встановленню зручних кліматичних умов для мешканців, у якості даних для розробки і досліджень були вибрані показники температур та споживання енергії. Проте, розроблені у роботі моделі та, загалом, програмне забезпечення можна розширити шляхом додавання інших кліматичних вимірювань: вологість повітря, кількість сонячних днів тощо.

Окремим та не менш важливим компонентом у проектуванні систем керування інтелектуальним будинком є врахування теплофізичних процесів, які описуються відповідними динамічними рівняннями теплопереносу у приміщенні. Розв'язування таких рівнянь на основі даних, накопичених типовими сенсорами (датчиками температур

DS18B20, DHT22; датчиком споживання газу Smartico, або електроенергії Emporia) дозволяє як моделювати теплові процеси, так і використовувати їх розв'язки для уточнення прогнозування споживання енергії. Оскільки огляд наявних публікацій та комерційних рішень у контексті застосування теплофізичного моделювання та його поєднання з методами машинного навчання показав достатньо незначну кількість впроваджень таких систем, постає актуальною розробка даних методик. Відтак, у дисертації запропоновано підходи у застосуванні теплофізичного моделювання до оцінки теплофізичних характеристик будівель.

У першому розділі дисертаційної роботи розглянуто проблематики створення систем інтелектуального будинку, які здебільшого базуються на граничних та туманних обчисленнях; дано розширене визначення інтелектуального будинку та визначено актуальні проблеми його проєктування. Через те, що використання хмарних технологій зумовлює низку проблем, які стосуються приватності та безпеки даних, високої ціни, наявністю мережевих затримок й вимогою до постійного зв'язку по глобальній мережі Інтернет для забезпечення відклику системи в реальному часі, актуалізується потреба в розробці недорогих та ефективних рішень на базі локальних (граничних) систем.

Проаналізовані сучасний стан та аспекти граничних обчислень для інтелектуального будинку, що ґрунтуються на мікрокомп'ютерах Raspberry Pi 3 та мікроконтролерах Arduino. Розглянута пошарова апаратна схема побудови такої системи інтелектуального будинку та характеристики мікроконтролерів серії STM32 F/H, які можна використовувати як граничний пристрій у такій системі. Оскільки вони підтримуються пакетом розширення X-CUBE-AI, це дозволяє легко розгортати навчені моделі машинного навчання. Також, розглянуті актуальні засоби та підходи до обробки сенсорних даних в межах регресійних та нейромережевих моделей, останні з яких можна вбудовувати у мікроконтролери з метою прогнозування. Використання отриманого прогнозу дозволяє оптимізувати режими роботи пристроїв

обігріву, або виявляти аномальні надходження з сенсорів, під'єднаних до мікроконтролера з розгорнутою нейронною мережею.

Для забезпечення гнучкого розгортання нейромережевих моделей, проведений огляд та аналіз автоматизованих засобів циклу обробки даних — навчання та валідації нейронної мережі — вбудовування нейромережі на мікроконтролер з одночасним її розгортанням на мікрокомп'ютері, що становить собою центральний обчислювальний хаб запропонованого прототипу апаратно-програмного комплексу.

У другому розділі дисертаційної роботи описано дані, які використовуються у моделюванні та тестуванні системи. До цих даних належить відкрита база вимірювань кліматичних й енергетичних показників REFIT, що містить температури та показники споживання енергії у 20 інтелектуальних будинках та вимірювання температур, здійснених у лабораторії інтелектуальних автономних систем факультету електроніки та комп'ютерних технологій Львівського національного університету імені Івана Франка за період з 1 лютого 2021 року по 1 вересня 2021 року. Для обробки цих даних розглянуті та реалізовані алгоритми виявлення викидів та аномалій (STL-декомпозиція, метод ізольованого лісу); проведений статистичний аналіз та побудовані регресійні моделі: а) зовнішніх та внутрішніх температур; б) споживання газу та температур на поверхнях батарей та в) зовнішньою, внутрішньою температури та температури на обігрівальних елементах, результати яких можна використовувати для прогнозу та аналізу зв'язків між вимірюваннями. Виявлено, що кут нахилу прямої регресії пов'язаний із теплофізичними характеристиками будинку.

Запропоновано підхід до дезагрегації даних споживання газу на основі кластерного аналізу, метою якого є виділення конкретних споживачів газу з сумарних (агрегованих) даних.

Для побудови нейромережевих прогностичних моделей температур детально описаний процес підготовки та перетворення даних та розглянуті актуальні архітектури рекурентних та гібридно-згорткових

нейронних мереж, на основі яких реалізовані відповідні моделі. Сформульована задача оптимізації гіперпараметрів спроектованих нейромереж, розв'язування якої здійснено за допомогою генетичного алгоритму. Здійснений аналіз результатів та підбір оптимальної архітектури для розгортання на мікроконтролері STM32.

Третій розділ дисертації присвячений апаратно-програмній реалізації прототипу комплексу для аналізу та обробки даних інтелектуального будинку. В якості апаратної платформи були обрані мікрокомп'ютер Nvidia Jetson Nano, що має достатні потужності для центрального обчислювального хабу і навчання нейромереж, та високопродуктивний мікроконтролер STM32 F767. Останній є базою первинних граничних обчислень завдяки розгорнутій на ньому нейронній мережі. При під'єднанні до нього датчиків температур, нейроконтролер може здійснювати короткотермінове прогнозування у реальному часі для порівняння з даними, що надходять на нього (випадок виявлення аномалій), чи використовуватися як окремий прогностичний модуль для оптимізації параметрів обігріву приміщення.

Програмна частина (створена мовою програмування Python 3) даного прототипу складається з REST API, що огортає базу даних з вимірюваннями SQLite, модулі статистичної обробки та нейромережевого моделювання, систему для організації конвеєру MLOPS — Mlflow, оркестратор Apache Airflow, що відстежує виконання усіх кроків конвеєру та модуль комунікації з нейроконтролером на базі TCP сокетів.

У четвертому розділі запропонована методика застосування теплофізичних моделей у поєднанні з даними температур та споживання газу для наближеної оцінки ефективних коефіцієнтів теплопровідності та теплоємності будівлі. Обчислення таких параметрів відбувається на основі розв'язків прямої та оберненої задач шляхом формулювання проблеми оптимізації функціоналу, який визначає різниця між обчисленими та реальними температурами у приміщенні відносно теплофізичних коефіцієнтів. Також, у межах задачі теплопереносу

розроблена математична модель оцінки ефективних теплових параметрів індивідуальних джерел обігріву, які можна використовувати як для задач прогнозування споживання, так і для визначення частини від сумарної енергії обігріву, яка споживається конкретним обігрівальним елементом.

Ключові слова: розумний будинок, машинне навчання, глибоке навчання, граничні обчислення, регресійні моделі, рекурентні нейронні мережі, теплофізичне моделювання, математичне моделювання, нейроконтролер, REST API, STM32, Nvidia Jetson Nano, об'єктно-орієнтоване програмування.

ABSTRACT

Sinkevich O.O. Optimization of the functioning of intelligent objects using machine learning methods. — Qualifying scientific work on manuscript rights.

Dissertation for obtaining the scientific degree of Ph.D in specialty 122 "Computer Science". — Ivan Franko National University of Lviv, Lviv, 2023.

The problems of optimizing the functioning of intelligent objects determine a wide class of problems, which includes both hardware complexes that provide production processes and their corresponding software. Programming of such complexes in the context of setting optimal operating modes requires solving the problems of stable hardware and software integration and developing effective algorithms for their operation. The evolution of existing microcomputer and microcontroller technologies, which constitute the "intelligent" basis of most intelligent objects, necessitates the need for constant improvement of already existing software tools and their adaptation to new platforms. At the same time, as the requirements for such intelligent objects grow, so does the complexity of software design and implementation.

The use of machine learning in the creation of software tools for intelligent objects is becoming an increasingly common practice, as it allows the implementation of flexible and diverse models for predicting the state of the object, its dynamic self-learning (reinforcement learning) or data that allow more productive use of the working system. An example of this kind of intelligent object, the functioning of which can be ensured and optimized by machine learning methods, is a smart home. In the dissertation, the main attention is paid to the development and improvement of the prototype of the hardware and software complex for the analysis and processing of smart home data, within which the integration of microcomputer and neurocontroller systems with open software, which is based on the use of modern deep

learning approaches, MLOPS techniques and REST architecture, is carried out.

The analysis of scientific publications and open sources on the topic of intelligent buildings showed that at the moment there is a lack of solutions for hardware and software systems, the functioning of which does not depend on cloud technologies and access to the global Internet. Also, applied architectural concepts involving deep learning models and edge or fog computing, distributed microcontroller systems that support embedding and deployment of the mentioned models are not fully disclosed. Therefore, the task of the dissertation consists in supplementing and expanding knowledge in the subject area of designing a prototype of a microcontroller data processing system of an intelligent house in the context of boundary computing in combination with machine learning, statistical methods and practices of building application programming interfaces (API).

Since in most cases the functioning of an intelligent house concerns the optimization of energy consumption and the establishment of comfortable climatic conditions for residents, temperature indicators and energy consumption were selected as data for development and research. However, the models developed in the paper and the software in general can be extended by adding other climate measurements, such as air humidity, number of sunny days, etc.

A separate and important component in the design of intelligent building control systems is the consideration of thermophysical processes, which are described by the corresponding dynamic equations of heat transfer in the room. Solving such equations on the basis of data collected by typical sensors (temperature sensors DS18B20, DHT22; Smartico gas consumption sensor, or Emporia electricity consumption sensor) allows modeling both thermal processes and using their solutions to refine energy consumption forecasting. Since the review of available publications and commercial solutions in the context of the application of thermophysical modeling and its combination with machine learning methods showed a fairly small number of

implementations of such systems, the development of these methods becomes relevant. Therefore, the dissertation offers approaches in the application of thermophysical modeling to the assessment of thermophysical characteristics of buildings.

In the first chapter of the dissertation, an overview of the problems of creating intelligent home systems, which are mostly based on boundary and fuzzy calculations; an extended definition of an intelligent building is given and the actual problems of its design are identified. Due to the fact that the use of cloud technologies causes a number of problems related to data privacy and security, high price, the presence of network delays, and the requirement for constant communication over the global Internet to ensure real-time system response, the need for the development of inexpensive and effective solutions based on local (boundary) systems.

The current state and aspects of edge computing for the intelligent home, based on Raspberry Pi 3 microcomputers and Arduino microcontrollers, are analyzed. The layer-by-layer hardware scheme of building such an intelligent home system and the characteristics of STM32 F/H series microcontrollers, which can be used as an edge device in such a system, are considered. Since they are supported by the X-CUBE-AI extension package, it makes it easy to deploy trained machine learning models. Also, current means and approaches to sensor data processing within regression and neural network models are considered, the latter of which can be built into microcontrollers for the purpose of forecasting. Using the obtained forecast allows you to optimize the operating modes of heating devices, or detect abnormal inputs from sensors connected to a microcontroller with a deployed neural network.

To ensure the flexible deployment of neural network models, a review and analysis of automated data processing cycle tools — neural network training and validation — neural network embedding on a microcontroller with simultaneous deployment on a microcomputer, which is the central

computing hub of the proposed hardware and software complex prototype, was conducted.

The second chapter of the dissertation describes the data used in modeling and testing the system. These data include the REFIT open database of measurements of climate and energy indicators, which contains temperatures and energy consumption indicators in 20 intelligent houses and temperature measurements carried out in the laboratory of intelligent autonomous systems of the Faculty of Electronics and Computer Technologies of the Ivan Franko National University of Lviv in the period from February 1, 2021 to September 1, 2021. Algorithms for detecting emissions and anomalies (STL-decomposition, isolated forest method) were considered and implemented to process these data; statistical analysis was carried out and regression-correlation models of a) external and internal temperatures were built; b) gas consumption and battery surface temperatures and c) external, internal and heating element temperatures, the results of which can be used for forecasting and analysis of relationships between measurements. It was found that the slope angle of the regression line is related to the thermophysical characteristics of the house.

An approach to the disaggregation of gas consumption data based on cluster analysis is proposed, the purpose of which is to select specific gas consumers from aggregated (aggregated) data.

For the construction of neural network prognostic temperature models, the process of data preparation and transformation is described in detail, and current architectures of recurrent and hybrid-convolutional neural networks, on the basis of which the corresponding models are implemented, are considered. The problem of hyperparameter optimization of designed neural networks is formulated, the solution of which is implemented using a genetic algorithm. Analysis of the results and selection of the optimal architecture for deployment on the STM32 microcontroller was carried out.

The third section of the dissertation is devoted to the hardware and software implementation of the complex prototype for analyzing and

processing data of an intelligent building. As a hardware platform, the Nvidia Jetson Nano microcomputer, which has sufficient power to serve as a central computing hub and to train neural networks, and a high-performance STM32 F767 microcontroller were chosen. The latter serves as the basis of primary boundary calculations thanks to the neural network deployed on it. When temperature sensors are connected to it, the neurocontroller can make short-term predictions in real time for comparison with the data coming to it (in case of anomaly detection), or be used as a separate predictive module to optimize room heating parameters.

The software part (created in the Python 3 programming language) of this prototype consists of a REST API wrapping a database with SQLite measurements, modules for statistical processing and neural network modeling, a system for organizing the MLOPS pipeline — Mlflow, an Apache Airflow orchestrator that monitors the execution of all pipeline steps and a communication module with a neurocontroller based on TCP sockets.

In the fourth chapter, a method of applying thermophysical models in combination with temperature and gas consumption data is proposed for an approximate estimation of the effective coefficients of thermal conductivity and thermal capacity of the building. Calculation of such parameters takes place on the basis of solutions of direct and inverse problems by formulating the problem of optimization of the functional, which determines the difference between the calculated and real temperatures in the room relative to the sought coefficients. Also, within the framework of the heat transfer problem, a mathematical model for estimating the effective thermal parameters of individual heating sources has been developed, which can be used both for consumption forecasting tasks and for determining the part of the total heating energy consumed by a specific heating element.

Keywords: smart home, machine learning, deep learning, edge computing, regression models, recurrent neural networks, thermophysical modeling, mathematical modeling, neurocontroller, REST API, STM32, Nvidia Jetson Nano, object-oriented programming.

Список публікацій здобувача за темою дисертації

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. публікації у фахових виданнях України:

1. Sinkevych O. Statistical Analysis of the Thermal Parameters of Smart Homes / O. Sinkevych, L. Monastyrskii, B. Sokolovskyi // Electronics and Information Technologies. – 2018. – No. 10. – P. 99–108.
2. Sinkevych O. Gas Disaggregation Approach Based on Cluster Analysis / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Boyko // Computer Technologies of Printing. – 2019. – Vol.- 41 (1). – P. 23-33.
3. Sinkevych O. Estimation of effective thermal parameters of heating sources based on dynamic measurements in smart home / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Boyko // Bulletin of the National University "Lviv Polytechnic". Series: "Computer systems of design. Theory and practice – 2020 – No. 1(1). – P.–58-66.
4. Sinkevych O. Development of neuro-controller based on STM 32 / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Boyko // Electronics and Information Technologies. – 2020. – No. 13. – P. 118–125.
5. Sinkevych O. Low complexity recurrent neural networks for edge computing / O. Sinkevych // Electronics and Information Technologies. – 2021. – No. 16. – P. 11–19.
6. Sinkevych O. MLOPS prototype of AI system for edge computing / O. Sinkevych. Ya. Boyko, L. Monastyrskyu // Electronics and Information Technologies. – 2022. – No. 17. – С. 74–83.
7. Сінькевич О. Регресійні моделі для аналізу даних інтелектуального будинку / О. Сінькевич // Електроніка та інформаційні технології. – 2022. – Вип. 20. – С. 62–73.

2. публікації у виданнях, які проіндексовані у міжнародній наукометричній базі Scopus:

1. Estimation of smart home thermophysical parameters using dynamic series of temperature and energy data / Oleh Sinkevych [et al.] // 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), Lviv, Ukraine, 2–6 July 2019. P. 934-937.
2. Algorithm of tuning heating source thermophysical parameters in smart home / Oleh Sinkevych [et al.] // 2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, Ukraine, 22–26 April 2020. P. 9-12.
3. Embedding sequence model in STM32 based neuro-controller / Oleh Sinkevych [et al.] // 2021 IEEE 12th International Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 19–21 March 2021. P. 113-118.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Sinkevych O. Determination of Regression Parameters for the Thermal and Energy Components of Smart Homes / O. Sinkevych, L. Monastyrskiy, B. Sokolovskiy // International Scientific and Practical Conference "Electronics and Information Technologies" (ELIT-2018). A-92 A-95. 2018.
2. Сінькевич О. Встановлення кореляційних зв'язків між тепловими параметрами розумних будинків / О. Сінькевич, Л. Монастирський, Б. Соколовський // Матеріали Міжнародної наукової конференції студентів і молодих науковців з теоретичної та експериментальної фізики "Еврика-2018". – Н13.
3. Cluster analysis of smart home energy time series / Олег Сінькевич [та ін.] // Матеріали IV Міжнародної науково-технічної конференції "Теоретичні та прикладні аспекти радіотехніки, приладобудування і комп'ютерних технологій" присвячена 80-ти річчю з дня народження професора Я. І. Проця, 2019. – С. 237–240.

4. Sinkevych O. Dynamic approach to identification of smart home thermophysical parameters / Oleh Sinkevych // International Conference of Students and Young Researchers in Theoretical and Experimental Physics “HEUREKA-2019”, 14-16 May 2019. – II.
5. Sinkevych O. To the modeling of smart home heating behavior / Oleh Sinkevych // International Conference of Students and Young Researchers in Theoretical and Experimental Physics “HEUREKA-2020”, 6-7 October. 2020. – G7.
6. Sinkevych O. Genetic hyper-parameters optimization for STM32 deep learning model / Oleh Sinkevych, Oleksandr Rechynskiy // International Conference of Students and Young Researchers in Theoretical and Experimental Physics “HEUREKA-2021”, 18-20 May 2021. – G10.
7. Пат. № 140565 Україна, МПК G01N 25/20, G01N 27/18, G06F 15/00. Спосіб визначення теплофізичних характеристик архітектурних споруд / Монастирський Л. С., Соколовський Б. С., Сінкевич О. О., Оленич І. Б. Заявник і власник – Львівський національний університет імені Івана Франка. – № u201905725; заявл. 27.05.2019; опубл. 10.03.2020 р. Бюл. № 5.

ЗМІСТ

Вступ.....	19
Розділ 1. Аналіз розробок інтелектуальних об'єктів на прикладі інтелектуального будинку.....	28
1.1. Інтелектуальний будинок. Концепція та актуальні проблеми.....	29
1.2. Граничні (edge) та туманні (fog) обчислення для ІБ.....	31
1.3. Сенсорні системи, граничні обчислення та інтелектуальний хаб для ІБ.....	34
1.4. Алгоритми та моделі обробки даних ІБ.....	38
1.4.1. Регресійні моделі клімату ІБ.....	41
1.4.2. Деагрегація споживання енергії ІБ.....	42
1.4.3. Трансформація та злиття даних ІБ.....	43
1.4.4. Нейромережеві та прогностичні моделі.....	45
1.5. Апаратно-програмні системи розгортання моделей МН для ГО.....	47
1.5.1. Апаратні системи розгортання.....	47
1.5.2. Програмні рішення розгортання.....	49
1.6. Теплофізичне моделювання ІБ.....	52
Висновки до розділу 1.....	54
Розділ 2. Розробка статистичних та нейромережевих моделей... 57	57
2.1. Опис та характеристика даних.....	58
2.2. Очищення та підготовка даних.....	60
2.3. Регресійні моделі та статистичний аналіз.....	63
2.3.1. Регресійні моделі для зовнішньої та внутрішньої температур. .65	65
2.3.2. Регресійна модель для споживання газу та температури обігрівальних елементів.....	68
2.3.3. Регресійна модель для внутрішньої, зовнішньої температури та споживанням газу.....	71
2.4. Модель деагрегації газу та кластерний аналіз.....	74
2.5. Розробка нейромережевих прогностичних моделей для нейроконтролера.....	75
2.5.1. Обробка та трансформація даних для нейромережі.....	77
2.5.2. Перетворення даних у тривимірні тензори.....	79
2.5.3. Моделі на базі рекурентних нейронних мереж.....	80
2.5.3.1. Довга короткочасна пам'ять LSTM.....	83

2.5.3.2. Вентильний рекурентний вузол GRU.....	85
2.5.3.3. Гібридна згортова рекурентна нейромережа (CNN- LSTM/GRU).....	86
2.6. Оптимізація гіпер-параметрів.....	88
2.7. Відбір нейромережі для нейроконтролера.....	90
Висновки до розділу 2.....	94
Розділ 3. Прототип нейроконтролерної системи.....	97
3.1. Концепція прототипу нейроконтролерної системи.....	97
3.2. Постановка задачі побудови нейроконтролерної системи.....	99
3.3. Неперервне розгортання моделей у системі нейроконтролера.....	100
3.4. Апаратна архітектура нейроконтролерної системи.....	102
3.5. Програмна архітектура нейроконтролерної системи.....	104
3.5.1. Структура серверної частини.....	105
3.5.2. Програмні компоненти та конвеєр розгортанням.....	108
3.6. Процес та результати роботи прототипу.....	113
3.7. Переваги, обмеження та подальший розвиток прототипу.....	121
Висновки до розділу 3.....	123
Розділ 4. Розробка теплофізичних моделей інтелектуального будинку.....	125
4.1. Постановка задачі оцінки теплових характеристик ІБ.....	126
4.2. Розв'язування та дослідження прямої задачі.....	129
4.3. Розв'язування та дослідження оберненої задачі.....	132
4.4. Оцінка теплової ефективності джерел обігріву.....	135
4.4.1. Обчислення ефективних теплових параметрів.....	137
4.4.2. Метод уточнення ефективних теплових параметрів.....	144
Висновки до розділу 4.....	148
Висновки.....	150
Список використаних джерел.....	152
Додаток А. Список публікацій здобувача.....	176

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ,
ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

- ІБ — інтелектуальний будинок
- МН — машинне навчання
- ГН — глибоке навчання
- ГО — граничні обчислення
- ПЗ — програмне забезпечення
- ІоТ — інтернет речей
- LSTM — довга короткочасна пам'ять
- GRU — вентильний рекурентний вузол
- CNN — згорткова нейронна мережа
- RNN — рекурентна нейронна мережа
- DAG — орієнтований ациклічний граф

ВСТУП

Актуальність теми. В епоху четвертої індустріальної революції (Industry 4.0) розробка, розгортання, забезпечення якості функціонування та оптимізація роботи інтелектуальних об'єктів становлять пріоритетні задачі перед дослідниками. Інтелектуальний об'єкт як кіберфізична система, яка об'єднує сучасні обчислювальні ресурси з фізичним світом, тісно пов'язаний із розвитком автономних систем, що передбачають корекцію своїх дій на основі отриманої сенсорної інформації з навколишнього середовища та власного апаратного стану. Комунікація таких об'єктів переважно базується на безпроводних технологіях передачі даних (Wi-Fi, Bluetooth LE, ZigBee, LoRa) та інтелектуальних модулях, які обробляють отримані сигнали з метою генерації керуючих команд. Розроблення архітектури інтелектуального об'єкту є процесом інтеграції комплексної апаратної складової з різноманітних сенсорів (температури, вологості, руху, оптичних та лазерних систем тощо), сімейства мікроконтролерів та більш потужних обчислювальних вузлів на базі мікрокомп'ютерів (наприклад, сімейства Raspberry Pi) чи серверних хмарних технологій. В якості програмної частини таких систем все більш актуальними і поширеними стають рішення на основі підходів машинного навчання (МН), які забезпечують автономність та відгук в межах реального часу.

Важливу категорію інтелектуальних об'єктів представляють інтелектуальні будинки (ІБ) як системи взаємопов'язаних сенсорних та виконавчих пристроїв, призначених для забезпечення комфортних кліматичних умов та енергоощадного споживання. Логіка задоволення таких умов ґрунтується на застосуванні різного роду оптимізаційних алгоритмів для планування оптимального графіку вмикання/вимикання обігріву на базі, наприклад, результатів прогнозування майбутніх

внутрішніх або зовнішніх температур [1]. Проте, більшість таких підходів спирається на комерційні хмарні рішення, які мають низку недоліків, такі як мережеві затримки, що ускладнюють прийняття рішень в реальному часі, закритість алгоритмів та архітектури, проблеми зі збереженням приватності даних, можливість несанкціонованого доступу, тощо [2].

Проектування інтелектуального будинку передбачає процес інтеграції апаратного і програмного забезпечення. В межах першого здійснюється розроблення сенсорних систем, які за допомогою мережевих протоколів передають дані як на локальний обчислювальний пристрій (хаб), так і на віддалений хмарний сервіс. Враховуючи недоліки хмарних технологій, актуалізується потреба у дослідженні та впровадженні апаратних систем, що використовують локальний хаб. Попри наявні пропозиції таких систем, зараз існує недостатність вичерпного аналізу застосування різних локальних пристроїв для інтелектуального будинку, шляхів їхнього проектування на основі потужних мікроконтролерів та готових реалізацій. Наприклад, до цього варто віднести розробку розподілених обчислювальних компонентів як сенсор-мікроконтролер-мікрокомп'ютер, де мікроконтролер здійснює частину обчислень (парадигма *mist computing* [3]). Для цього у дисертаційній роботі спроектована апаратна частина, що складається з мікроконтролера STM32, який з розгорнутою на ньому нейромережею становить нейроконтролерний пристрій. Вибір даного типу пристрою мотивувався підтримкою пакету розширення X-CUBE-AI, що полегшує розгортання моделей машинного навчання. Він відповідає за попередню обробку даних, а саме здійснює короткотермінове прогнозування температури та розвантажує ресурс основного обчислювального хабу на базі Nvidia Jetson Nano.

Зокрема, до актуальності теми дисертації варто віднести розгортання нейромережі на мікроконтролері сімейства STM32 та відповідне дослідження цього процесу у комбінації з хабом підходами MLOPS, що становить, де-факто, стандарт у комерційних та промислових

рішеннях на базі машинного навчання. Базуючись на сучасних практиках роботи з IoT-системами, у дисертації програмне забезпечення розроблялось з використанням REST архітектури [4] та програмного каркасу FastAPI з підтримкою асинхронних обчислень.

Оскільки моделі машинного навчання мають все більш поширене застосування в IoT, їх впровадження вимагає різнобічного та подальшого дослідження у контексті задач інтелектуального будинку. Наприклад, розроблення та аналіз прогностичних моделей для дослідження зв'язків між кліматичними показниками ІБ та даними споживання енергії, чи непряма оцінка теплофізичних характеристик будівлі мотивується задачами оптимальних режимів опалення з метою економії природних ресурсів.

Не менш важливим підходом є доповнення моделей машинного навчання фізичними моделями, які описують теплову поведінку у приміщенні. Аналіз існуючих методів та моделей інтелектуального будинку дає підстави стверджувати про необхідність розроблення та покращення фізико-математичних моделей з їх подальшою інтеграцією з програмними модулями, які оптимізують функціонування інтелектуального будинку [5].

Також, важливим напрямком є розробка оптимізаційних алгоритмів з використанням методів машинного навчання, які доповнюються фізико-математичними моделями. Наприклад, на основі таких моделей можна неявним чином оцінювати ефективні теплофізичні параметри будівлі, чи індивідуальних джерел обігріву [6, 7].

Отже, на підставі аналізу поточного стану даної проблематики залишаються відкрити низка проблем. По-перше, розробка та дослідження нейромережевих моделей для прогнозування кліматичних параметрів ІБ, які можна розгортати на мікроконтролері в умовах обмеження на пам'ять пристрою та його потужність. По-друге, розроблення регресійних моделей, які можна використовувати як в цілях

прогнозу, так і для розуміння динаміки температур та споживання енергії обігріву. Для коректного застосування цих моделей актуальною стає й задача дезагрегації енергії — виділення частки енергії, що відповідає конкретному споживачеві на основі сумарних (агрегованих) даних, де частина з них не стосується процесів обігріву приміщень. По-третє, це проектування апаратно-програмних комплексів, що забезпечують розподілені обчислення в контексті граничних та туманних обчислень (ГО та ТО відповідно) на основі поєднання високопродуктивних мікроконтролерів та обчислювального хабу. Програмне забезпечення такого комплексу повинно відповідати сучасним практикам створення надійних систем з можливістю неперервного розгортання та взаємодії між компонентами. По-четверте, розробка теплофізичних моделей інтелектуального будинку для оцінки ефективних теплофізичних параметрів як будівлі, так і джерел обігріву, що дає змогу аналізувати теплову ефективність. Вирішення цих проблем, зокрема, становить основу для оптимізації функціонування інтелектуального будинку.

Аналіз сучасного стану робіт, які стосуються функціонування ІБ показав, що зараз запропоновано ряд підходів та методів для вирішення проблем оптимізації інтелектуального будинку. У більшості випадків застосовуються комерційні хмарні рішення, які поряд з позитивними сторонами володіють низкою недоліків, описаних вище. В зв'язку з цим, актуальності набуває розробка більш доступних рішень на базі розподілених граничних обчислень.

Зв'язок роботи із науковими програмами, планами, темами.

Дисертаційні дослідження виконувалися у Львівському національному університеті імені Івана Франка на кафедрі радіоелектронних і комп'ютерних систем в рамках науково-дослідних тем в межах робочого часу «Проектування інтелектуальних мікрокомп'ютерних систем», «Апаратно-програмне забезпечення інтелектуальних систем розумного будинку» та «Розпізнавання образів і біометричний захист інформації»

засобами машинного навчання» (номери держреєстрації: 0116U001679, 0122U200452 та 0119U002328, науковий керівник професор Монастирський Л. С., 2016–2021 рр.), які відповідають науковому напрямку розробки та проектування інтелектуальних автономних систем.

Мета і задачі дослідження. Метою дисертаційної роботи є розробка нових апаратно-програмних рішень, які дозволяють підвищити ефективність функціонування інтелектуальних об'єктів шляхом використання підходів на базі граничних обчислень, адаптованих до апаратно-обчислювальних систем малої потужності. Для досягнення поставленої мети необхідно було розв'язати такі задачі:

1. здійснити аналіз існуючих методів обробки сенсорних даних інтелектуального будинку, зокрема підходів на базі алгоритмів машинного навчання в межах парадигми граничних обчислень;
2. побудувати та проаналізувати регресійні моделі для кліматичних даних інтелектуального будинку; розробити та дослідити підхід до задачі дезагрегації споживання енергії на базі машинного навчання;
3. розробити прототип апаратно-програмного комплексу — нейроконтролерну систему для розподіленої обробки даних для оптимізації функціонування інтелектуального будинку: розгортання нейромережевої моделі для прогнозування температури на базі мікроконтролера STM32 та центральний обчислювальний хаб на базі мікрокомп'ютера Nvidia Jetson Nano;
4. реалізувати програмне забезпечення для такої системи з використанням клієнт-серверної архітектури, практики MLOPS та забезпечити мережеву комунікацію між нейроконтролером та хабом з використанням сокетів;
5. розробити фізико-математичні моделі та алгоритми для аналізу теплових станів інтелектуального будинку шляхом розв'язання обернених параметричних оптимізаційних задач теплоспоживання й інтегрувати отримані результати з розробленим програмним

комплексом з метою оптимізації енергоспоживання інтелектуального будинку;

6. розробити методику обчислення ефективних теплових параметрів джерел обігріву.

Об'єктом дослідження є процеси функціонування інтелектуальних об'єктів.

Предметом дослідження є методи, моделі та алгоритми обробки даних для оптимізації функціонування інтелектуального будинку.

Методи дослідження. У дисертаційній роботі для розв'язання поставлених завдань використані такі методи: для обробки та аналізу сенсорних даних інтелектуального будинку – методи математичної (регресійно-кореляційний аналіз) та описової статистики, методи апроксимації, методи кластеризації, методи виявлення аномалій на базі навчання без учителя; для розробки нейроконтролерної системи – теорії вбудованих систем та штучних нейронних мереж; для побудови фізико-математичних моделей – теорію термодинаміки, варіаційні методи; для розв'язання обернених задач та задач мінімізації – квазі-ньютонівський метод, генетичні та еволюційні алгоритми; методи градієнтного підсилення та регресійний аналіз; а при розробленні програмних модулів – принципи об'єктно-орієнтованого, клієнт-серверну архітекту, REST, теорію реляційних баз даних та техніки MLOPS.

Наукова новизна отриманих результатів. На основі виконаних теоретично-практичних досліджень отримано наступні результати:

- вперше запропонована модель оцінки теплофізичних параметрів інтелектуального будинку на базі температурної динаміки та відповідних даних споживання енергії для обігріву, на основі якої можна аналізувати і оптимізувати енергоспоживання в інтелектуального будинку;
- вперше розроблена фізико-статистична модель непрямої ідентифікації ефективного теплофізичного параметра нагрівального елемента з використанням рівняння теплопереносу та динаміки внутрішніх і зовнішніх температур повітря, а також температур на

нагрівальних елементах, встановлених у кімнатах інтелектуального будинку, що дає змогу опосередковано оцінювати відносну ефективність обігрівальних елементів;

➤ отримали подальший розвиток моделі та методи дезагрегації даних споживання енергії, які дають змогу визначати парціальні вклади в енергоспоживання конкретними споживачами, а також використано регресійні методи для моделювання кліматичних та енергетичних даних інтелектуального будинку;

➤ запропоновано новий підхід до обробки сенсорних даних інтелектуального будинку з використанням нейроконтролеру на базі STM32 та розгорнутої на ньому рекурентної нейронної мережі для ефективного прогнозування внутрішніх температур в межах концепції граничних обчислень;

➤ відповідно до концепції граничних обчислень запропоновано і реалізовано прототип апаратно-програмного комплексу для оптимізації функціонування інтелектуального будинку з використанням рекурентних нейромереж, REST архітектури, MLOPS підходів та технології мережевих сокетів.

Практичне значення отриманих результатів полягає в тому, що вони дають можливість розробляти нові підходи для проектування інтелектуальних об'єктів в межах парадигми граничних та туманних обчислень, в яких використовуються методи машинного навчання, зокрема:

➤ спроектований прототип нейроконтролерної системи дозволяє забезпечити функціонування інтелектуального будинку без використання хмарних сервісів та доступу до глобальної мережі Інтернет;

➤ розроблені регресійні моделі для кліматичних показників інтелектуального будинку дозволяють здійснювати прогнозування кімнатних температур на основі даних споживання газу, зовнішніх температур і температур на обігрівальних елементах, а також оптимізувати енергоспоживання;

- створена нейромережева модель для нейроконтролера з оптимізованими гіперпараметрами для прогнозування майбутніх значень кімнатних температур на основі рекурентних нейронних мереж, яка дозволяє підвищувати ефективність роботи систем автоматизації для забезпечення теплових режимів в інтелектуальному будинку;
- реалізоване програмне забезпечення нейроконтролерної системи згідно REST архітектури та MLOPS підходів дозволяє вдосконалювати та покращувати процес розробки систем для інтелектуального будинку в межах граничних обчислень;
- запропоновані фізико-математичні моделі в комбінації з методами обробки даних дозволяють опосередковано визначати теплофізичні характеристики інтелектуального будинку та джерел обігріву, що забезпечує більш гнучке налаштування керувальних автоматизованих систем;
- створений прототип та відповідне програмне забезпечення може використовуватись та доповнюватись для обробки більш складних класів даних та слугувати базою для реалізації комплексної системи керування та оптимізації інтелектуального будинку.

Отримані у роботі результати впроваджені у відповідні навчальні курси у Львівському національному університеті імені Івана Франка.

Особистий внесок здобувача. Результати роботи, наведені в дисертації, отримано автором особисто. В рамках роботи над публікаціями здобувачеві належать такі результати: у статтях [8, 9] розроблені статистичні моделі кліматичних характеристик, у статті [10] – алгоритм та модель дезагрегації споживання газу, у статті [6] – за допомогою програмного забезпечення отримані значення ефективних теплових параметрів обігрівальних елементів, у [11] – побудована нейромережева модель прогнозу внутрішніх температур ІБ та здійснена програмна реалізація, в праці [12] – розроблений підхід та здійснена програмна реалізація обчислення гіперпараметрів нейромережі з використанням генетичного алгоритму, у [13] – реалізований прототип

системи розгортання нейронної мережі на мікрокомп'ютері Nvidia Jetson Nano.

Апробація результатів дисертації. Основні положення та результати дисертації доповідались та обговорювались на: Міжнародній науковій конференції студентів і молодих науковців з теоретичної та експериментальної фізики “ЕВРИКА”, (Львів, 2018, 2019, 2020, 2021); Міжнародній науково-практичній конференції ЕЛІТ-2018, (Чинадієво, 2018); Міжнародній науковій конференції UKRCON 2019 (“IEEE 2nd Ukraine Conference on Electrical and Computer Engineering”), (Львів, 2019); Всеукраїнській науково-технічній конференції “Теоретичні та Прикладні Аспекти Радіотехніки, Приладобудування і Комп'ютерних Технологій” (Тернопіль, 2019); Міжнародній науково-технічній конференції “Перспективні технології і методи проектування MEMS” (“2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design”), (Львів, 2020); Міжнародній конференції 2021 IEEE XIIth International Conference on Electronics and Information Technologies (ELIT), (Львів, 2021), а також на наукових семінарах кафедри радіоелектронних і комп'ютерних систем Львівського національного університету імені Івана Франка (2018-2022).

Публікації. За час написання дисертації опубліковано 7 статей [6, 8-13] у журналах, що входять до переліку фахових видань, затверджених МОН України. Також видано 14 тез і матеріалів всеукраїнських та міжнародних наукових конференцій [5, 14-22] ([5, 17 20, 21] включено до міжнародної наукометричної бази Scopus).

Структура дисертації. Дисертація складається із анотації, вступу, чотирьох розділів із висновками, списку використаних джерел (192 позиції) та додатку із публікаціями здобувача. Загальний обсяг становить 179 сторінок. Робота містить 61 рисунок та 9 таблиць.

РОЗДІЛ 1. АНАЛІЗ РОЗРОБОК ІНТЕЛЕКТУАЛЬНИХ ОБ'ЄКТІВ НА ПРИКЛАДІ ІНТЕЛЕКТУАЛЬНОГО БУДИНКУ

Дослідження IDC [23] показало, що кількість пристроїв, підключених до Інтернету, включаючи автомобілі, датчики та камери, які складають IoT, перевищить 41,6 мільярда, генеруючи 79,4 зеттабайт (ZB) даних у 2025 році. Statista [24] оцінює, що загальна встановлена база інтелектуальних пристроїв, підключених до Інтернету речей, до 2025 року становитиме 30,9 мільярда одиниць, що є різким стрибком із 13,8 мільярдів одиниць, оцінених на 2021 рік. Прогнозована кількість пристроїв та зростаючий тренд у розвитку інтелектуальних об'єктів визначає довгостроковий горизонт не лише для апаратного і програмного вдосконалення компонентів Інтернету речей, а й для побудови ринкових стратегій щодо ефективного впровадження інтелектуальних технологій у виробництво та залучення інвестицій.

Визначальною характеристикою інтелектуального об'єкта є його локальна автономність прийняття рішення, забезпечена ефективними програмними компонентами на основі штучного інтелекту, зокрема моделями машинного або глибокого навчання. Такий інтелектуальний об'єкт належним чином “реагує” на дані, отримані безпосередньо сенсорними системами, або через локальну чи глобальну мережу. Масштаб та технологічна складність інтелектуального об'єкту може варіюватися від простих систем адаптивного термостату [25] до багаторівневої системи керування виробничими процесами чи організації колективного ройового інтелекту [26].

До актуальних проблем розробки інтелектуальних об'єктів у межах Інтернету речей належать [26-29]: а) апаратно-програмний розрив — у зв'язку з високою інтенсивністю появи нових програмних засобів виникає

неоднорідність між перевіреними і комерційно успішними апаратними рішеннями та процесом адаптації до нових програмних систем; б) вразливість програмного забезпечення — чутливість пристроїв до злому та перехоплення конфіденційних даних; в) системна складність — відсутність єдиного стандарту під'єднання пристроїв, забезпечення інфраструктури зберігання даних та багаторівневий контроль безпеки; г) недостатня обчислювальна потужність та масштабованість — широке застосування віддалених хмарних технологій (Microsoft Azure, Google Cloud, AWS) хоч і вирішує проблему з обчислювальними ресурсами, проте використання у повному обсязі рішень на базі локального хабу може бути більш економічно виваженим, швидким та безпечним.

Окремим класом інтелектуальних об'єктів, для якого окреслені проблеми розробки є досі актуальними та потребують комплексного вирішення, є інтелектуальний будинок.

1.1. Інтелектуальний будинок. Концепція та актуальні проблеми

Інтелектуальні будинки, також відомі як автоматизовані будинки, є прикладом квазі-автономної системи, яка може коректувати свій стан в залежності від отриманих сенсорних даних. Спочатку технологія розумного дому використовувалася для управління кліматичними та побутовими системами, такими як освітлення та опалення, але останнім часом використання розумних технологій розвинулося так, що майже будь-який електронний компонент будинку можна включити в систему [31]. Основою ІБ є система безперебійної комунікації між сенсорами, устаткуванням та інтелектуальним центром прийняття рішень, який реалізовується у вигляді хмарного серверу [32] чи локально встановленого хабу [33]. Зараз на ринку існує низка комерційних систем ІБ: екосистема продуктів Google Nest (термостат, системи безпеки, “розумні” замки тощо), esobee, Xiaomi, Apple HomeKit тощо, які в тій чи

іншій мірі забезпечують інтеграцію та функціонування багатьох компонент ІБ.

Користувач може корегувати, змінювати умови, або задавати розклад роботи ІБ через веб-браузер, мобільний додаток та інші засоби доступу до API.

Прикладом розробки такої системи є робота [34], де адаптивні хмарні пристрої IoT під назвою Intelligent Tetris Switch (ITS) у поєднанні з аналітикою великих даних та глибоким навчанням використовуються для надання послуг персоналізації ІБ. У статті [35] авторами описаний підхід до об'єднання сенсорів в ІБ за допомогою платформи Arduino, технології ZigBee, взаємодії з використанням хмарних сервісів та покращенні ефективності обміну даними у JSON форматі. Зокрема, загальний огляд стану та розробки хмарних сервісів ІБ наведений у статтях [36, 37].

Однією з найбільш поширених систем ІБ є комплекси щодо управління температурними режимами: датчики температур, адаптивний (інтелектуальний) термостат та механізм регуляції опалення. Попри наявні на ринку комерційні хмарні апаратно-програмні рішення (Nest, eesobee), тривають розробки недорогих і локально-встановлених систем. Наприклад, у роботі [38] запропонована мікроконтролерна система на базі Arduino і ESP8266 для регуляції режиму опалення, а у роботі [39] представлено мікроконтролерне рішення системи управління ІБ, що ґрунтується на мікропроцесорі STM32F407VGT6 і вбудованій операційній системі реального часу $\mu\text{C}/\text{OS-II}$.

Причинами цього явища згідно з [40] є такі проблеми: а) приватність (дані, які збираються інтелектуальними термостатами можуть надаватися та відстежуватися сторонньою особою); б) безпека (можливість отримання доступу до системи сторонньою особою з метою здійснення шахрайських дій); в) помилка у навчанні інтелектуального термостату впливає на період робочого функціонування та призводить до неправильних рішень для системи опалення; г) наявність екстремальних

умов, що завжди ігноруються інтелектуальними термостатами; д) висока ціна; е) проблеми з мережею Інтернет (мережеві затримки) можуть суттєво знизити ефективність.

Окреслені виклики зумовлюють актуальність дослідження та проектування апаратно-програмних систем ІБ, яка підкріплюється як динамікою появи нових технологій, так і відповідними проблемами, пов'язаними з адаптацією до поточних потреб.

1.2. Граничні (edge) та туманні (fog) обчислення для ІБ

Одним зі шляхів вирішення вищенаведених проблем розглянемо підхід, що передбачає застосування граничних та туманних обчислень [41] для проектування простої системи ІБ.

Граничні обчислення (ГО) над накопиченими даними здійснюються лише на найближчих до сенсорів *вузлах* (мікроконтролери, мікрокомп'ютери, локальні сервери, смартфони тощо) замість передачі в хмарне сховище. Дані обробляються в околі області, де відбувається їх генерація, а *вузол* виступає в якості основного, або проміжного обчислювального хабу. Система вузлів дозволяє легко додавати нові компоненти, масштабувати ресурси в межах представленої архітектури та реалізовувати розподілені мікроконтролерні обчислення [42]. ГО займають нішу між хмарними пристроями та пристроями IoT [43]. Позаяк, вони не є заміною хмарним сервісам, а можуть слугувати в якості її розширеної версії.

Вагомою перевагою ГО над хмарними технологіями є оптимізація мережевого навантаження за допомогою проведення периферійних обчислень у IoT *вузлах*, підключених до локальної мережі або всередині самого обладнання локальної мережі, що в результаті дає меншу фізичну відстань між обчислювальним центром і сенсорами без додаткової мережевої затримки [43].

На противагу перевагам ГО, досі існує низка недоліків та відкритих проблем [44]: складність масштабування системи та узгодженості різнорідних даних, мережева безпека у безпроводних технологіях, збільшення використання енергії у багатокomпонентних системах; ці виклики актуалізують подальше дослідження та інженерію ГО. Розширений аналіз цієї проблематики розглянутий у роботі [45].

У роботі [46] розглянута одна з проблем ГО (узгодженості різнорідних даних) та запропоновано проект взаємосумісності рішення для ІБ, що використовують концепції ГО, віртуальних пристроїв IoT (VID) та Інтернету речей. У статті [47] описана мікроконтролерно-серверна система ІБ на базі Arduino Uno та RESTful архітектури для керування пристроями у будинку. Недоліком такої схеми є використанням малопотужного мікроконтролера. Також, можливим покращенням такої системи є її доповнення алгоритмами машинного навчання для “інтелектуалізації” автоматизації процесу. У роботі [48] ідея, викладена в [47] розширена додаванням простих правил для регулювання освітлення та температури, проте все ще не передбачає адаптивності системи шляхом її “навчання” до зміни умов функціонування. Інтелектуалізація систем ІБ розглянуті в [48-50] у контексті питань безпеки з використанням моделей машинного навчання та нечіткої логіки.

Оскільки, однією з першочергових проблем проектування ІБ є зниження споживання енергії та забезпечення оптимальних і зручних температурних умов, у роботах [51-54] опрацьовані підходи до імплементації “розумних” кліматичних систем, а саме регресійна модель для прогнозування кімнатних температур на базі онлайн-навчання [52]; моделі на базі глибокого навчання (LSTM, Encoder-Decoder LSTM) для інтелектуального термостату [53]; використання *глибокого навчання з підкріпленням* для керування системою HVAC (опалення, вентиляція та кондиціонування повітря) [54] та розглянутий нейромережевий прогностичний алгоритм для точного прогнозування мінімальної та

максимальної температури у приміщенні [55]. Регресійні моделі у [52] можна покращити використавши рекурентні нейронні мережі, або градієнтне підсилення. У [53] прогностична модель на основі LSTM використовувалась для обчислення оптимального розкладу термостату та продемонструвала найвищу точність, яку позаяк можна підвищувати шляхом додавання нових ознак до даних, наприклад, показників сонячної радіація, вологості тощо. В роботі [54] автори запропонували алгоритм для енергоменеджменту для ІБ для ефективного контролю HVAC за відсутності моделі теплової динаміки будівлі, з урахуванням комфортного температурного діапазону в умовах невизначеностей параметрів будинку, який можна доповнювати більш складними поведінковими моделями.

Комплексний апаратний підхід до ГО для ІБ описаний у [56], де в якості обчислювального вузла автори використали мікрокомп'ютер Raspberry Pi та запропонували багат шарову систему ІБ, яка включає підходи REST API та залучення технології Node-RED [57]. Відкритим питанням у даній роботі лишається доповнення такої системи засобами машинного навчання, що дозволить більш ефективно реалізовувати алгоритми автоматизація ІБ.

У більшості проаналізованих публікаціях ГО для ІБ побудовані на малопотужних пристроях, таких як сімейство мікроконтролерів Arduino та мікрокомп'ютерів Raspberry Pi 2/3. Попри невисоку вартість, такі пристрої не надають достатньо великої обчислювальної потужності для проектування та розгортання моделей машинного, чи глибокого навчання. Для вирішення цієї проблеми у даній роботі вибір був зроблений на користь мікрокомп'ютера Nvidia Jetson Nano, який містить 128-ядерний графічний прискорювач та дозволяє створювати прототипи моделей машинного і глибокого навчання для ІБ, користуючись спеціалізованою бібліотекою TensorFlow 2.

Оскільки проектування ГО для ІБ зумовлює появу нових підходів, у дисертаційній роботі розвивається підхід ГО, що спирається на апаратно-

програмну інтеграцію мікрокомп'ютера Nvidia Jetson Nano, мікроконтролера сімейства STM32 й засобів розгортання моделей МН.

1.3. Сенсорні системи, граничні обчислення та інтелектуальний хаб для ІБ

Типову багат шарову апаратну структуру ІБ зображено на рис. 1.1. Сенсорні пристрої (камери, інтелектуальні лампи/замки, датчики температури, вологості та руху) інтегруються з відповідними мікроконтролерами для забезпечення мережевої передачі даних, первинної обробки сигналів тощо.

Така інтеграція може бути у вигляді готового, зазвичай комерційного приладу, чи шляхом об'єднання окремих датчиків з мікроконтролером та модулями передачі даних. Прикладом останнього є система з температурним датчиком DHT11, мікроконтролером STM32L100 та модулем ZigBee, де в якості хабу використано Raspberry Pi [57].

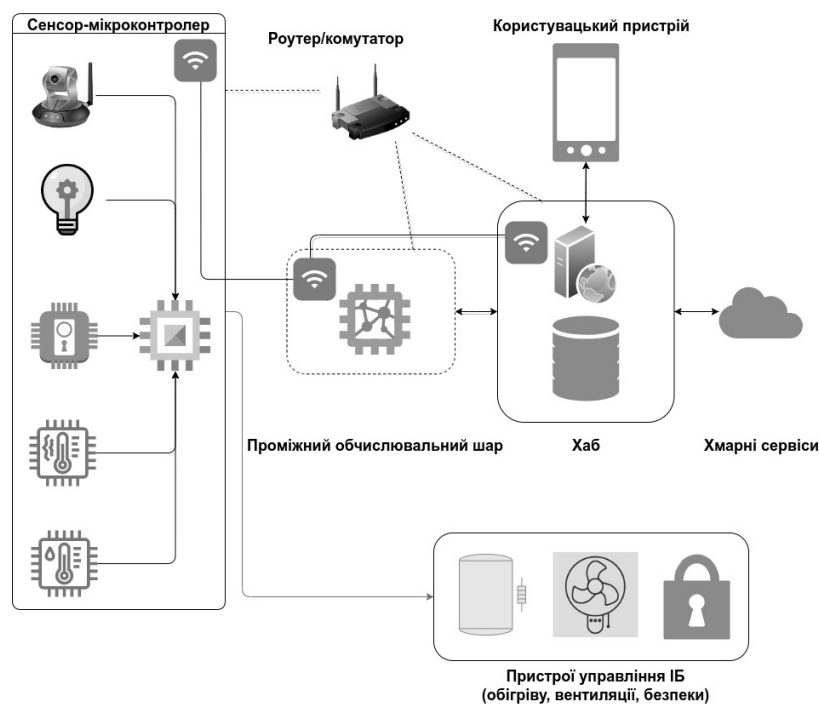


Рис. 1.1. Апаратна структура ІБ

Мережеві комунікації зазвичай реалізуються бездротовими технологіями на основі стандартів ZigBee, LoRaWAN, Bluetooth LE, що можуть бути або інтегровані у сенсорний пристрій, або реалізовуватися окремими модулями [59, 60]. В окремих випадках з'єднання може відбуватися через Wi-Fi роутер чи мережевий комутатор, позаяк таке рішення не є енергоефективним.

Другим *допоміжним* обчислювальним шаром може виступати мікроконтролерна система, метою якої є розвантаження обчислювального процесу на хабі. Мікроконтролер здійснює деякі проміжні обчислення над даними, що надходять з сенсорних пристроїв та передає результати на хаб. В цілому такий процес забезпечує розподільність, масштабованість та оптимальне навантаження на мережу IoT. До прикладу, у роботі [61] пропонується система з кількома мікроконтролерами, що поєднує апаратний міст і кілька масштабованих мікроконтролерів, щоб реалізувати граничний шлюз для інтелектуальних датчиків в режимі реального часу та покращуючи масштабованість порівняно з наявними промисловими рішеннями IoT.

Програмне забезпечення (ПЗ) таких мікроконтролерних систем для ГО визначається обмеженнями, що накладаються на допустимі апаратні обчислювальні ресурси. Наприклад, мікроконтролер ESP32-S3 має двоядерний процесор Xtensa LX7 240 MHz, 400 KB SRAM та модуль Bluetooth 5 (LE) [62]. Ці характеристики дозволяють розгорнути невеликі моделі машинного навчання для обробки вхідних даних [63].

Інше сімейство мікроконтролерів, яке часто використовують у таких задачах є STM32 F/H серії. Розглянемо характеристики найбільш продуктивних варіантів (STM32H745 та STM32F767) та схему розгортання моделей машинного навчання [64]:

- STM32H745 побудований за двоядерною схемою (32 бітні 480 MHz Cortex-M7 та 240 MHz Cortex-M4) та має 1 MB SRAM та 2 MB флеш-пам'яті при можливості роботи за різних режимів енергоспоживання;

- STM32F767 побудований за одноядерною схемою (32 бітний 216 MHz Cortex-M7) та має 512 KB SRAM та 2 MB флеш-пам'яті при можливості роботи за різних режимів енергоспоживання.

Ці мікроконтролери завдяки підтримці пакету розширення X-CUBE-AI придатні до розгортання моделей машинного навчання, реалізованих у популярних фреймворках Tensorflow Lite, scikit-learn, XGBoost, PyTorch [65]. Моделі, створені в цих бібліотеках конвертуються у бінарні формати .h5, .tflite, або .onnx (з використанням сервісу ONNX) [66].

Наприклад, процес розгортання нейронної мережі в загальному випадку можна описати як (рис. 1.2):

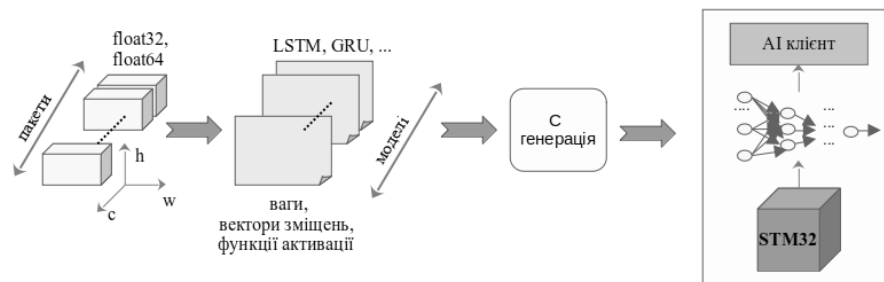


Рис. 1.2. Схема розгортання нейромережі на STM32

- завантаження натренованих моделей на мікроконтролер через та конвертацію у типи даних мови C;
- моделі визначаються відповідними топологіями (LSTM, GRU тощо), масивами ваг, зміщень, функціями активацій та розмірностями вхідних тензорів;
- вхідні тензори даних для системи X-CUBE-AI повинні задовольняти обмеження у розмірності $batch * h * w * c$, де $batch$ — розмір пакету даних, h, w, c — параметри розмірності тензора;

- типи даних вхідного тензора для мікроконтролерів STM32H745 та STM32F767 можуть бути як цілочисельні типи int8, uint8 так і типи з плаваючою комою float32 та float64;
- згенерована модель, яка повністю оптимізована під STM32 Arm® Cortex®-M4/M7/M33 розгортається на мікроконтролері, доступ до якої реалізується через клієнтську частину, що розробляється під конкретну задачу.

Глибокі нейронні мережі для розгортання на таких мікроконтролерах потребують процесу оптимізації, що передбачає зменшення як кількості шарів, так і кількості нейронів у відповідних шарах [67]. Під час проєктування вбудованих рішень на основі глибокого навчання необхідно визначити компроміс між швидкістю прогнозування, його точністю, споживанням енергії та об'ємом пам'яті мікроконтролера. Для цього існує декілька широковживаних підходів до розробки та оптимізації моделей глибокого навчання: а) квантизація моделей [68] — процес зменшення кількості біт, які використовуються для кодування ваг і тим самим скорочують розмір моделі, зокрема, внаслідок цілочисельного представлення вагових коефіцієнтів; б) застосування адаптованих до мобільних та мікрокомп'ютерних пристроїв нейромережевих архітектур, типу MobileNet [69], SqueezeNet [70], NBONext [71] тощо, які, тим не менше, не є універсальним рішенням та потребують модифікації під конкретні проблеми; в) використання підходу “дистиляції знань” (knowledge distillation), за яким нейронна мережа, спроектована за більш великою топологією тренує меншу нейронну мережу з мінімізованою втратою точності [72].

Далі, опрацьовані дані передаються до центрального хабу, де можуть підлягати наступній обробці, зберігатися у локальній базі даних, надсилатися до хмарного серверу чи/або на мобільний користувацький пристрій. На основі аналізу, хаб може автономно приймати ті чи інші рішення по управлінню ІБ, або коригувати їх за допомогою комунікації з користувачем. Наприклад, у роботі [73] запропоновано систему з хабом

Raspberry Pi 4 у поєднанні з датчиками температури та освітленості, а в [74] — поданий загальний огляд використання машинного навчання для ІБ. В контексті масштабування проблеми ІБ у роботі [75] розглянута мережа ІБ на базі ройового інтелекту, де ухвалення рішень мережевої системи ІБ реалізується шляхом застосування рекурентної нейронної мережі та методу навчання з підкріпленням.

Оскільки, ГО варто розглядати як частину більш загальних хмарних обчислень, дані можна передавати на віддалений сервер, наприклад, якщо необхідно використовувати складні моделі машинного навчання [76], або встановлений хаб є складовою високонавантаженого індустріального сервісу [77]. Керування пристроями ІБ (регуляція опалення, освітлення тощо) зазвичай виконується за допомогою PID-контролерів [78].

Загалом, аналіз наукових публікацій в контексті ГО для ІБ показав, що на даний момент підхід до інтеграції інтелектуальної мікроконтролерної системи з хабом не є у повній мірі висвітленим та актуалізує процес подальших досліджень, позаяк більшість напрацювань у даній галузі є комерційними закритими продуктами. Тому у роботі представляється розроблений прототип апаратно-програмного комплексу інтегрованої нейроконтролерної системи для ІБ.

1.4. Алгоритми та моделі обробки даних ІБ

Програмна реалізація ГО для ІБ потребує аналізу та застосування методів статистичної обробки даних та побудови різноманітних функціональних моделей для прогнозування, допомоги користувачеві у прийнятті рішень і керування системами ІБ. Сучасні комерційні концепції IoT переважно передбачають залучення алгоритмів машинного навчання [79], або застосування мереж Петрі [80]

Типово, конвеєр обробки даних має вигляд (рис. 1.3) [81]. Необроблені дані від сенсорів є складними, характеризуються значним

об'ємом та динамічними змінами, а відтак часто вимагають усунення шумів, зокрема Вейвлет-перетворенням [82]. Наприклад, у статті [82] автори використовують Вейвлет-перетворення для знешумлення у проблемі прогнозування CO₂ в ІБ. У випадку обробки температурних даних ІБ немає необхідності у накопиченні даних високої щільності, тому етап знешумлення в загальному випадку можна оминати.



Рис. 1.3. Схема обробки даних для IoT

Обробка пропущених значень відносно визначеного часового кроку має вагомий вплив, оскільки алгоритми аналізу даних переважно передбачають їх повноту та цілість, а наявність пропусків зумовлює низьку точність відповідних моделей. Існує низка підходів до вирішення цієї проблеми, що ґрунтуються як на машинному навчанні, так і класичних чисельних методах (лінійна інтерполяція). До першого типу можна віднести роботи [82-84], де у [83] запропонована кластерна модель Гауссівської суміші для заміщення пропущених значень, у [83, 84] — алгоритми на основі просторової та часової кореляцій. Також використовуються попередньо навчені прогностичні моделі [86] на зразок багат шарового перцептрона і глибокої мережі переконань. Недоліком

останнього є потреба у наявності значної кількості розмічених даних та ускладнення, обумовлене потребою у тренуванні нейромережі.

Більш прості підходи базуються на лінійні чи поліноміальній інтерполяції [87], що є доволі ефективним рішенням.

В реальному середовищі різні сенсори IoT системи є розподіленими, а помилки у їх налаштуваннях та апаратні збої можуть продукувати отримання помилкових вимірювань, наприклад надто великих, малих чи некоректних значень. Такі аномалії можуть бути як точковими, контекстними (у часових рядах), так і у вигляді інтервалів підпоследовностей періодичного характеру. Розв'язування задачі виявлення таких аномалій здійснюється статистичними чи методами машинного навчання. Прості приклади виявлення точкових викидів у даних наведені у статтях [88, 89] зі застосуванням методів ізольованого лісу та головних компонент відповідно. Деталізуючи процес пошуку аномалій у даних слід зважати на природу даних: одновимірний часовий ряд (дані температур, вологості тощо), багатовимірні часові ряди [90], відео-потік чи складні багатоконпонентні вимірювання, що описують функціонування одного пристрою.

Для виявлення аномалій в одновимірних часових рядах, які представляють IoT дані, пасують як STL-декомпозиція [91] (аналіз різницевої компоненти за деяким пороговим значенням), метод ізольованого лісу, так і прогностичні моделі, наприклад SARIMA [92] (порівняння прогнозованого значення з отриманим) тощо.

Використання машинного навчання для виявлення й усунення аномалій детально проаналізовано у [93], а глибокого навчання у [94]; показано, що дана проблема критично залежить від багатьох параметрів, як тип, кількість та якість даних і не може бути розв'язана одним універсальним методом.

Оскільки в загальному типи даних для ІБ мають доволі різноманітну природу (кліматичні дані, дані відеоспостереження, руху

тощо), розглянемо актуальні підходи статистичного аналізу до кліматичних показників.

1.4.1. Регресійні моделі клімату ІБ

Регресійний та кореляційний аналіз разом з описовою статистикою та розвідувальним аналізом даних дає розуміння розподілів та взаємозв'язків кліматичних даних ІБ. Так у роботах [8, 95] здійснений регресійно-кореляційний аналіз температурних та даних споживання енергії та побудовані лінійні прогностичні регресійні моделі. У роботі [96] поряд з вичерпним розвідувальним аналізом побудовані регресійні залежності між середньодобовою тривалістю обігріву та середньодобовою температурою зовнішнього повітря, а у роботі [97] розглянута регресійна модель для аналізу даних про енергоспоживання двох ІБ. З метою покращення результатів лінійних регресійних моделей, у [98] авторами пропонується нова модель ансамблевого прогнозування на основі регресії з вбудованою автоматизованою оптимізацією параметрів для прогнозування споживання електроенергії; в [99] в якості регресійної моделі була використана LSTM-нейромережа. Оскільки комбінація багатьох сенсорів в ІБ може вносити різний вклад у точність моделей, у статті [100] регресійний підхід на базі MIDFEL LASSO регресії розвинутий для визначення найвпливовішого датчика в кожному сегменті горизонту прогнозу.

Реальні дані ІБ часто доволі складно піддаються адекватного моделюванню лінійними моделями, тому виникає необхідність у застосуванні таких підходів, як регресія випадковим лісом, або градієнтним підсиленням [101].

Визначені статистичні параметри, наприклад, коефіцієнт детермінації у моделі між зовнішніми та внутрішніми температурами без обігріву може неявно та наближено оцінювати теплову інерційність ІБ і властивість зберігати тепло будівлею. Регресійна модель, наведена в [102]

дозволяє визначати теплові навантаження у будинку на основі фізичних властивостей будівельних компонент.

В цілому, регресійний аналіз є ваговою і ключовою компонентною при створенні цифрових аналітичних рішень для ІБ.

1.4.2. Дезагрегація споживання енергії ІБ

Проблемою у моделюванні кліматичних залежностей в ІБ є те, що не завжди є можливим виокремлювати частину енергії, яка йде на обігрів приміщень з сумарної її кількості. Задача дезагрегації полягає у розділенні сумарних даних споживання енергії на складові компоненти, що стосуються конкретного пристрою. Існують апаратний та програмний типи підходів до цієї проблеми [103]. До першої категорії належать “розумні” лічильники, розетки тощо, приєднані до кінцевого пристрою, а до другої — реалізації статистичних чи алгоритмів МН на основі аналізу та видобутку даних. Серед підзадач дезагрегації виділяють області електро та газоспоживання.

Один з перших відомих підходів NILM (non-invasive load monitoring) опублікований в [104] та являє собою процес виявлення зміни вимірних параметрів — дійсної та реактивної потужності приладу. Ці зміни являють собою переходи між майже постійними стаціонарними значеннями підлягають кластеризації. Кластери подібної величини та протилежного знаку є парними. Таким чином, визначаються два стани навантаження, як обігрівачі, які можуть бути лише увімкнені або вимкнені. Недоліком такого методу є неможливість об’єднати кластери складних пристроїв, таких як посудомийні машини, які мають багато станів.

Іншим популярним підходом є моделювання приладів в ІБ за допомогою факторіальних прихованих Марковських ланцюгів [104, 105], де метод використовує експертні знання для встановлення початкових моделей станів відомих приладів. Також у [106, 107] запропонований

підхід на базі навчання з вчителем, який використовує лише градієнти активної потужності для побудови моделі дерева рішень, що використовується для класифікації. Така модель повинна перетренуватися при появі нового приладу. Основна проблема вищезазначених підходів полягає в тому, що вони мало підходять для програм реального часу через їх висока обчислювальна складність. На противагу статистичним методам, у [108] застосовано глибоке навчання для задачі дезагрегації, що дало змогу підвищити точність класифікації.

Більшість алгоритмів дезагрегації електроенергії реалізуються у контексті навчання з учителем і передбачають наявність розміченої та промаркованої бази даних. Проте, у реальному використанні таке далеко не завжди є можливим, що зумовлює необхідність розробки алгоритмів навчання без учителя, або напівавтоматичного навчання [109].

У випадку дезагрегації споживання газу можна застосовувати простіші алгоритми, що пояснюється меншою кількістю споживачів та простішою природою даних. Наприклад, у [10] розглянутий кластерний підхід на основі подій (споживання газу) для наближеної ідентифікації конкретного користувача, а в [110] проведений ретельний аналіз публікацій і запропонований підхід для визначення та порівняння вікон подій з використанням DTW-метрики. Використання фільтру Калмана для дезагрегація наведено у [111].

На даний момент проблема дезагрегація енергії не є остаточно вирішеною і потребує окремих досліджень для конкретних даних.

1.4.3. Трансформація та злиття даних ІБ

Програмна реалізація схеми (рис. 1.3) в загальному зумовлює трансформацією та злиття різнорідних даних. Під трансформацією розуміється перетворення даних у такі структури, які можна використовувати в якості вхідних для машинного, або глибокого навчання. До цього процесу належать:

а) конвертація даних з одного формату в інший (наприклад, асоційовані масиви у звичайні масиви);

б) приведення даних у вигляді часових рядів до матричних форм (предиктори — залежні значення), розбиття їх на навчальну, перевірочну та тестувальні вибірки;

в) формування зручних для обробки масивів даних за допомогою доступних бібліотек (наприклад, Pandas [112], Dask [113] у середовищі Python 3);

г) перетворення до відповідних тензорних представлень за моделювання рекурентними, або згортковими нейронними мережами [114].

При повторному використанні трансформовані IoT дані для зручності можна зберігати у проміжному форматі (JSON, NumPy масиви, у таблицях SQL баз даних). Подальший розвиток у трансформації даних запропоновано у [115], де автори здійснили інтеграцію машинного навчання у СКБД з використанням апаратного прискорення.

Злиття даних — процес об'єднання множини даних і знань, які описують один і той самий об'єкт реального світу [116, 117]. Застосування методики злиття даних для ІБ розкрито у [118], де розроблено: переносний датчик руху для розпізнавання 3D-жестів; датчик руху, і його алгоритм позиціювання в приміщенні та модуль мультисенсорної схеми й інтелектуальний алгоритм виявлення пожежі та сигналізації для реалізації системи домашньої безпеки. Також, значну увагу у контексті злиття даних приділено в [119] та запропоновано систему локалізації мешканців ІБ з використанням інформаційного фільтру, де введено біомеханічну модель перевернутого маятника.

Після підготовки і перетворення даних та перед використанням безпосередньо методів машинного навчання доцільним буває здійснити виділення та відбір релевантних ознак для підсилення прогностичних якостей моделей. До прикладу, у [120] для точного прогнозування електричного навантаження в ІБ пропонується модель

середньострокового прогнозування, яка використовує погодинні дані про електричне навантаження та температуру для прогнозування погодинного споживання електроенергії на місяць вперед. Для попередньої обробки даних використовується модифікований вибір ознак на основі взаємної інформації про ентропію, що усуває надмірність і нерелевантність із даних. В [121] опубліковане дослідження щодо розробки нового підходу до виділення ознак у даних споживання електроенергії, швидкості вітру та параметрів фотоелектричних елементів, заснованому на кластеризації, вкладеному моделюванні та різноманітних методах лінійної регресії (CNR). Підхід на базі контекстного вибору ознак і оптимального механізму керування актуатором для мінімізації споживання енергії в ІБ розглянутий в [122], де описане впровадження алгоритму оптимізації рою частинок на основі штучної нейронної мережі (ANN-PSO) для тренування моделі оцінки важливості ознак. Виділені ознаки застосовуються як вхідні параметри для тренування LSTM нейронної мережі і модуля оптимального керування.

1.4.4. Нейромережеві та прогностичні моделі

В загальному, різноманітні рішення оптимізації функціонування ІБ, які ґрунтуються на залученні машинного навчання можна поділити на а) прогностичні моделі (задачі навчання з учителем), б) гібридні самооптимізовані системи (навчання з підкріпленням) [123] та в) прості системи на основі певних правил та детермінованих дерев знань [124]. Позаяк прогностичні моделі передбачають різноманітну кількість застосувань від відео-ідентифікації до передбачення кімнатних температур в ІБ, беручи до уваги завдання та тематику дисертаційної роботи, розглянемо сучасні підходи і результати розробки прогностичних моделей для кліматичних параметрів ІБ.

При побудові прогностичних моделей температур/вологості/споживання енергії в ІБ необхідно визначити характер та можливі комбінації вхідних даних. Одновимірні температурні часові розподіли можна представляти в якості часового ряду для захоплення певної закономірності у часі. У роботі [125] застосовано ARIMA модель для прогнозування внутрішньої та зовнішньої температури ІБ, а в [126] використана розширена модель ARIMAX (ARIMA, доповнена незалежними змінними) для прогнозування температури, вологості та кількості електроенергії, яка споживатиметься в ІБ.

Оскільки часто дані, що використовуються для прогнозування володіють нелінійними часовими залежностями, ARIMA моделі можуть демонструвати гіршу точність у порівнянні з алгоритмами на базі глибокого навчання (ГН) і рекурентних нейромереж [127-130]. Також для використання ARIMA моделей часовий ряд повинен бути стаціонарним, або приведеним до стаціонарного шляхом диференціювання, що не є обов'язковою вимогою для нейромережеских моделей [131].

В окремих випадках є ефективною комбінація LSTM нейромережі та ARIMA [132] для реалізації моделі прогнозу температури. Проте, в залежності від представлених даних в деяких випадках ARIMA моделі показують кращу точність за РНН [133].

В інших випадках дані, наприклад, кількох часових рядів, які становлять багатовимірну структуру, для моделювання використовуються як VAR [134], так і рекурентні нейромережескі моделі [135]. Це, зокрема, свідчить про універсальність застосування рекурентних нейромережеских моделей як для одновимірних, так і багатовимірних часових рядів.

З врахуванням позитивних сторін і гнучкості застосування моделей ГН до часових рядів, зокрема в контексті а) вивчення нелінійних часових залежностей; б) слабкої умови щодо забезпечення стаціонарності; в) підтримки багатовимірних часових рядів та їх багатокрокового

прогнозування; г) ефективного вивчення ознак (CNN) і часових залежностей (LSTM/GRU комірки); г) можливості гібридизації шарів (CNN-LSTM, ConvLSTM), у роботі реалізуються моделі ГН для програмного забезпечення нейроконтролера та досліджується ефективність використання таких моделей.

Підсумовуючи результати аналізу публікацій можна стверджувати, що процес, зображений на рис. 1.3 можна доповнювати, розширювати та адаптовувати як до розробки систем ІБ, так і для складніших комерційних рішень. Відтак у дисертаційній роботі застосовані алгоритми інтерполяції даних, виявлення аномалій (STL-декомпозиція, метод ізольованого лісу), регресійно-кореляційний аналіз для дослідження зв'язків між даними ІБ та побудовані прогностичні нейромережеві моделі для температурних даних, досліджена їхня ефективність в контексті розгортання на мікроконтролері та мікрокомп'ютері як складових системи ГО.

1.5. Апаратно-програмні системи розгортання моделей МН для ГО

Згідно схеми (рис. 1.3), оброблені дані та побудовані прогностичні моделі необхідно розгорнути в межах граничного пристрою. Відтак розглянемо наявні рішення для апаратного розгортання програмного рішення для ІБ та рекомендовані сервіси і програмні каркаси для зручного огортання коду і реалізації конвеєру.

1.5.1. Апаратні системи розгортання

Для реалізації нейроконтролера, що об'єднує мікроконтролер та розгорнуту на ньому нейромережу необхідно обрати апаратну платформу з підтримкою інтеграції відомих програмних каркасів TensorFlow 2 чи PyTorch. Серед доступних на ринку рішень можна виокремити рішення від компаній STMicroelectronics (STM32), Espressif Systems (ESP32),

Arduino Software (Arduino Nano) та Coral (Dev Board Micro). Оскільки апаратно-програмна реалізація у дисертаційній роботі зроблена на базі мікроконтролерів STM32, розглянемо деякі основні публікації та результати.

У статті [136] проведений аналіз застосування алгоритмів машинного навчання (нейронної мережі, методу опорних векторів, дерев рішень) на шістьох мікроконтролерах STM32, продемонстрована принципова можливість розгортання натренованих моделей МН на даних платформах. Як подальші кроки дослідження авторами запропоновано тестування цих платформ для розгортання моделей ГН (рекурентних та згорткових нейромереж). Для забезпечення зручності роботи з МН на платформах STM32, компанією STMicroelectronics розроблено рішення STM32Cube.AI як частина екосистеми STM32Cube для полегшеного впровадження моделей МН безпосередньо на сам пристрій [137]. Подальший розвиток дослідження цього напрямку описаний у роботах [138-140], де розглянутий процес навчання та розгортання рекурентних нейромереж для нейроконтролерів STM32 F серії з використанням STM32Cube.AI.

У контексті вибору програмних каркасів у роботі [137] здійснено порівняльний аналіз двох популярних фреймворків: Tensorflow Lite Micro на Arduino Nano BLE та CUBE AI для STM32-NucleoF401RE задля забезпечення стандартизованого критерію вибору каркасу для конкретних задач та виявлено, що структура CUBE AI краще підходить для високопродуктивних застосунків ГО. Процес розгортання та вбудовування моделей ГН описаний у третьому розділі роботи, а базові відомості щодо цього процесу опубліковані в електронних джерелах [139-140]. Результати, отримані у даних роботах свідчать про якісну можливість реалізації мікроконтролерних систем STM32 зі зручною інтеграцією машинного та глибокого навчання.

Для повної апаратної інтеграції (рис. 1.1) нейроконтролер необхідно з'єднати з основним обчислювальним хабом (розділ 3). Оскільки

передбачено використання саме моделей ГН, у роботі обрано продуктивний мікрокомп'ютер Nvidia Jetson Nano, який має підтримку технології CUDA та графічний прискорювач з 128 ядрами [141]. Така конфігурація дозволяє ефективно використовувати даний пристрій для тренування нескладних моделей ГН [142]. Перевагою Jetson Nano над широкоживаним у системах ІБ Raspberry Pi 3/4 є апаратне супроводження програмних каркасів ГН TensorFlow 2/PyTorch, що дозволяє здійснювати як локальне тренування нейромереж, так і одночасне функціонування в якості веб-серверу.

1.5.2. Програмні рішення розгортання

Безпосереднє розгортання натренованої моделі ГН на мікроконтролері STM32 з ядрами ARM® Cortex®-M здійснюється завдяки пакету розширення X-CUBE-AI та платформи STM32Cube, що складається з середовища візуальної конфігурації STM32CubeMX, інтегрованого середовища розробки STM32CubeIDE, програмованого пакету STM32CubeProgrammer та пакету вимірювання струму STM32CubeMonitor-Power.

Пакет X-CUBE-AI розширює функціональні можливості середовища STM32CubeMX, додаючи до неї автоматичний генератор бібліотеки нейронної мережі. Цей генератор перетворює файли моделей попередньо навчених нейронних мереж, що формуються різними фреймворками глибокого навчання (такими як Keras, TensorFlow Lite), в оптимізовану за швидкістю виконання та обсягом пам'яті (ОЗП та Flash) бібліотеку. Згенерована бібліотека автоматично інтегрується з додатком користувача, в результаті чого отримується повністю налаштований проект, готовий до компіляції та виконання на мікроконтролері STM32 [143]. Для першого налаштування системи мікроконтролера, ПЗ необхідно встановити на платформу x86_64, оскільки на даний момент не існує підтримки платформ ARM. Основою цього ПЗ є середовище

STM32CubeIDE, доступне для ОС Ubuntu® 16.04+, Windows® 10 та macOS® (x64). Також, існує інтерфейс командного рядка, що дозволяє автоматизувати основні операції та дозволяє уникнути використання STM32CubeIDE шляхом інсталяції STM32CubeMX.

Після встановлення та налаштування ГН моделі на мікроконтролері, отриманий нейроконтролер можна під'єднувати у систему обчислювального хабу, який керує подальшим процесом функціонування, розгортання та оновлення параметрів нейромережі. Для цього існує типовий конвеєр (рис. 1.4) MLOPS (*machine learning operations*) [144].

Відповідно до [145], техніка MLOPS підвищує продуктивність шляхом надання автоматизованого середовища з доступом до підібраних наборів даних; забезпечує повторювальність процесу, включаючи те, як модель навчається, оцінюється, версифікується та розгортається; відлагоджує практики CI/CD, що дає змогу не тільки швидко розгорнути, але й покращувати якість і послідовність виконання та дозволяє перевіряти і звіряти версії всіх даних та навченої моделі для їх відстежування і налагодження.

В якості СКБД для проектів ГО рекомендовано використовувати швидкі та легкі рішення SQLite, InfluxDB, Realm [146], що не потребують складного налаштування, проте за наявності достатньо продуктивного хабу та інтеграції з окремими програмними компонентами (Mlflow), сервіс БД можна ускладнити (PostgreSQL, MongoDB) та ізолювати у окремий Docker контейнер.

Окремою важливою частиною схеми є сховище ознак (Feast, Vertex.ai) [147] для збереження та їх повторного використання (для нових зразків даних, ознаки відразу обчислюються та зберігаються, що визначає більш гнучку архітектуру).

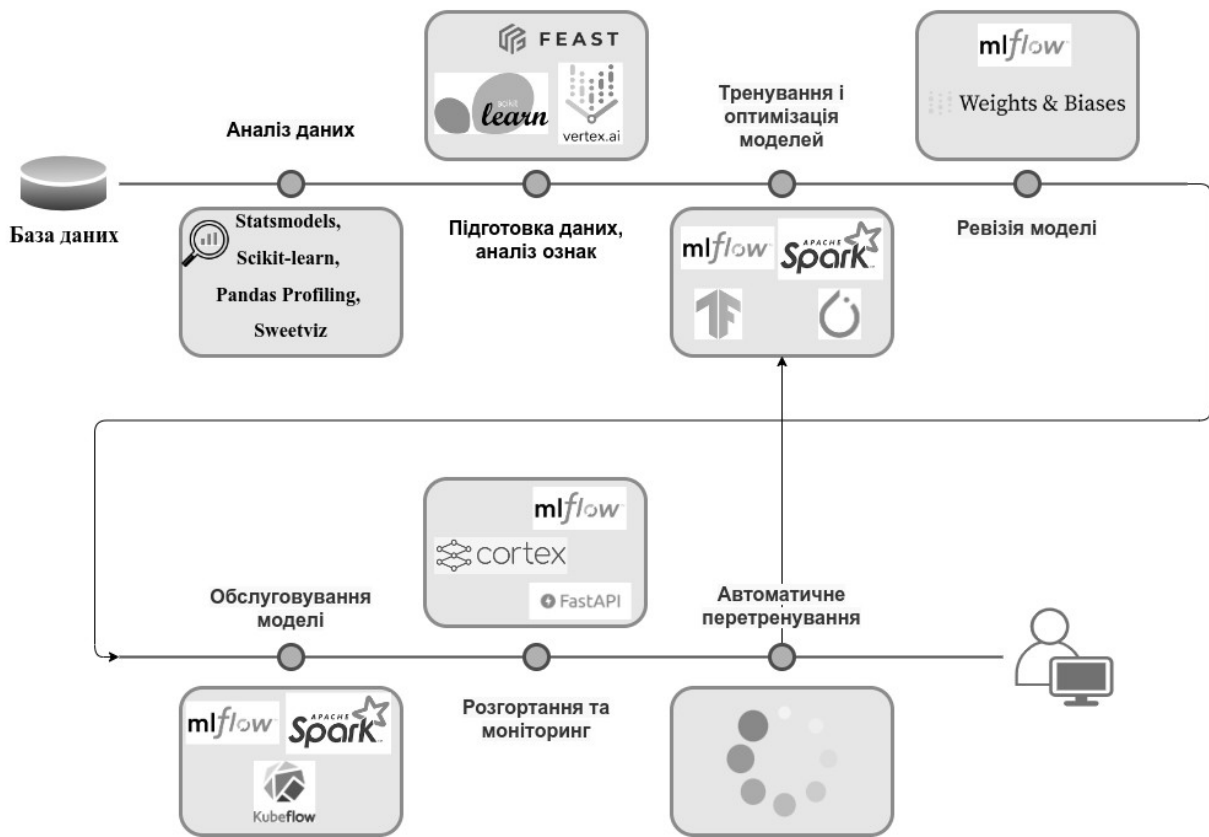


Рис. 1.4. Компоненти конвеєру MLOPS

Побудова нейромережових моделей (TensorFlow 2, PyTorch) та їх тренування з подальшим обслуговуванням можна контролювати за допомогою сервісу реєстрації, збереження та розгортання Mlflow [148]. Важливою частиною цього етапу є оптимізація гіперпараметрів та конфігурації моделі, що може бути здійснена як вбудованими рішеннями (TensorFlow 2/Keras), так і з використанням генетичних алгоритмів чи еволюційного програмування.

Для розгортання та моніторингу моделей поряд з Mlflow для створення REST API можна використовувати легкий мікро-фреймворк FastAPI [149], що має вбудовану підтримку асинхронному програмуванню. Окрім цього, такий функціонал існує і в межах TensorFlow 2 (TensorFlow Serving). Оркестрування, або керування процесом виконання кожної компоненти з рис. 1.4. забезпечується

сервісами Apache Airflow, або Mlflow Pipelines та детально розглянута в розділі 3.

В цілому, вибір програмних засобів для MLOPS залежить від багатьох факторів та визначається складністю проекту [150].

Дослідивши технології у процесах автоматизації MLOPS, у дисертаційній роботі розроблений та застосований конвеєр для побудови і розгортання нейромережевих моделей прогнозування температур ІБ, реалізований процес з'єднання нейроконтролера з обчислювальним хабом та проаналізована ефективність такого комплексу (розділ 3).

1.6. Теплофізичне моделювання ІБ

Паралельно з розробкою систем керування ІБ, побудовою моделей МН для оптимізації функціонування та реалізації прогностичних моделей кліматичних показників, особливий інтерес становить ідентифікація теплофізичних параметрів ІБ. Під такими параметрами мається на увазі як ефективна теплоємність, так і теплопровідність стін будинку. Знання цих параметрів у поєднанні з даними про енергію та температуру, конфігурацію термостата та моделі активності мешканців уможливають створення нового підходу до розробки високоефективних систем енергоменеджменту ІБ.

Серед різноманіття існуючих підходів до моделювання теплової поведінки будівлі (методи «білої», «чорної» і «сірої» скриньки), останній поєднує в собі статистичну та фізичну сутність теплових параметрів будівлі [96]. Отже, метою вищезазначеного методу є моделювання як надходження тепла, так і теплові втрати будівлі. Наприклад, в [151] виміряні дані про температуру та енергію були використані у стохастичних диференціальних рівняннях для моделювання теплової поведінки ІБ на базі методу «сірої» скриньки. В [152] запропонований метод обчислення коефіцієнту тепловтрат ІБ на основі даних короткострокового моніторингу з використанням двох підходів: методу

«сірої скриньки» та авторегресійного моделювання. Наступний внесок у згадані підходи зроблено в [153], де автори використали просте динамічне рівняння теплопередачі, щоб встановити співвідношення між спожитою та втраченою енергією, а також кількістю першої для підвищення температури в приміщенні за допомогою концепції градусо-днів. Щоб удосконалити наявні методики, в [96] створені нові метрики для оцінки впливу теплової маси на необхідну енергію для опалення та охолодження будівель.

Зокрема, в [5] запропоновано методику визначення ефективних коефіцієнтів теплопровідності та теплоємності в ІБ на основі розв'язування оберненої задачі, сформульованої в рамках моделі теплопровідності.

Окремим напрямком у дослідженні теплової ефективності ІБ є оцінка ефективних теплових параметру джерела опалення. Цей параметр наближено відображає теплову поведінку розумного будинку та може бути оцінений, використовуючи динамічну модель на основі рівняння теплопереносу у інтелектуальному будинку [6]. У роботі [20] ідею, викладену в [6] доповнено алгоритмом оптимізації на базі методу рою часток для уточнення шуканого параметру джерела опалення.

Проблема теплофізичного моделювання ІБ із залученням статистичних алгоритмів обробки даних чи моделей машинного навчання на даний момент є актуальною, що підтверджується наведеними публікаціями. У дисертаційній роботі сформульована обернена задача ідентифікації теплофізичних параметрів ІБ, запропонована методика її розв'язування, а також проведене дослідження щодо визначення ефективного теплового параметру обігрівальних елементів ІБ, що є корисним для побудови більш точних моделей теплової поведінки та оптимізації функціонування ІБ.

Висновки до розділу 1

Інтелектуалізація об'єктів IoT та забезпечення їх оптимального функціонування важлива при створенні надійних та ефективних автономних апаратно-програмних комплексів для вирішення багатьох інженерних задач. Відтак розробка нових IoT пристроїв та одночасний розвиток існуючих концепцій граничних обчислень потребує проведення нових і глибоких досліджень. Зокрема, важливою частиною таких досліджень є впровадження сучасних методів машинного та глибокого навчання, їх аналіз та оцінка ефективності.

Задля розширення та доповнення наявних технологій, у дисертаційній роботі розроблено прототип апаратно-програмного комплексу для системи інтелектуального будинку, метою якого є аналіз даних та реалізації прогностичних моделей (розділи 2, 3) з метою оптимізації функціонування інтелектуального будинку. В даному розділі проаналізовані публікації та підходи, що відображають поточний стан у проблематиці інтелектуальних рішень для інтелектуального будинку, які здебільшого ґрунтуються на застосуванні статистичних методів та підходів машинного навчання. Ці методи можуть ефективно застосовуватися у комплексі з пристроями граничних обчислень, такими як мікроконтролери та мікрокомп'ютери. Оскільки хмарні рішення для ІБ стикаються з низкою проблем (приватність, безпека даних, тимчасова втрата доступу до глобальної мережі Інтернет, мережеві затримки), виникає необхідність у подальшому проектуванню рішень на базі граничних обчислень.

З огляду на це, розглянуті апаратні засоби граничних обчислень, їх типи, а також архітектура багатокomпонентних систем. Для інтелектуалізації граничних обчислень проаналізовані програмні каркаси машинного і глибокого навчання та механізми їх використання з мікроконтролерами. Виявлено, що одним з найбільш зрілих і доступних

рішень створення граничних обчислень для інтелектуального будинку є мікроконтролери сімейства STM32, що мають підтримку розгортання моделей машинного навчання STM32Cube.AI. Огляд мікрокомп'ютерів продемонстрував, що оптимальним пристроєм для роботи з моделями глибокого навчання є мікрокомп'ютер Nvidia Jetson Nano, що має підтримку технології CUDA та графічний прискорювач з 128 ядрами.

Для вибору і оцінки поточного стану програмного забезпечення системи інтелектуального будинку, проведено огляд публікацій, де розглянуті підходи до а) аналізу даних; б) побудови регресійних моделей; в) розв'язування задачі дезагрегації даних споживання енергії та г) створення прогностичних нейромережових моделей. Також, проаналізована схема опрацювання IoT-даних, визначено та описано її складові. Для розгортання створених нейромережових моделей, опрацьований MLOPS підхід та відповідні програмні засоби.

У доповнення до цього, розглянуті підходи до теплофізичного моделювання інтелектуального будинку, а саме до оцінки ефективних теплофізичних параметрів інтелектуального будинку (коефіцієнтів теплопровідності будівлі та теплоємності внутрішнього вмісту будівлі) та джерел обігріву, що створюють підґрунтя розробки програмного забезпечення для оптимізації функціонування інтелектуального будинку.

Виходячи з аналізу матеріалів можна підсумувати, що на даному етапі приділяється недостатньо уваги а) розвитку розподілених технологій граничних обчислень для інтелектуального будинку; б) побудові та аналізу прогностичних нейромережових моделей для кліматичних даних інтелектуального будинку; в) теплофізичному моделюванню інтелектуального будинку та його поєднанню з алгоритмами машинного навчання. Тому, в дисертаційній роботі розглянуті задачі нейромережового моделювання, створення прототипу апаратно-програмного комплексу для оптимізації функціонування інтелектуального будинку з метою забезпечення енергоефективності й обробки відповідних даних та побудови теплофізичних моделей для

оцінки ефективних теплових параметрів інтелектуального будинку та джерел обігріву.

РОЗДІЛ 2. РОЗРОБКА СТАТИСТИЧНИХ ТА НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ

Проектування IoT систем в контексті граничних обчислень та МН потребує проведення статистичного аналізу даних, які використовуються у розроблених моделях. Одні з найбільш поширених видів даних ІБ є кліматичні (температури повітря та обігрівальних елементів, вологості) і значення споживання енергії обігріву та електроенергії, що використовується приладами.

Цей розділ присвячений розробці статистичних моделей для: а) кліматичних даних та значень споживання енергії обігріву, отриманих з відкритої бази даних проекту *REFIT Smart Homes* [154], метою чого є покроковий дизайн та відлагодження процесу моделювання як даних ІБ, так і дослідження нових моделей на базі реальних вимірювань; б) температурних даних, накопичених у лабораторії інтелектуальних автономних систем факультету електроніки та комп'ютерних технологій Львівського національного університету імені Івана Франка за період з 1 лютого 2021 року по 1 вересня 2021 року.

Для нейромережевого моделювання, а саме для побудови прогностичних моделей для температур використовуються лабораторні вимірювання. На їх основі спроектовано та досліджено прості рекурентні нейронні моделі, які можна розгортати на малопотужному граничному пристрої — мікроконтролері (розділ 3).

В межах статистичного моделювання представлені регресійні моделі для оцінки зв'язків між даними та запропоновано підхід до розв'язування задачі дезагрегації даних споживання газу, що ґрунтується на алгоритмах кластерного аналізу. Ці моделі становлять частину розробленого програмного комплексу для ІБ.

2.1. Опис та характеристика даних

Оскільки накопичення достатньо великої кількості власних сенсорних даних, що вичерпно описують поведінку ІБ у розумінні динаміки розподілу кімнатної температури у комплексі зі споживанням енергії обігріву, не є завжди можливим внаслідок відсутності усіх необхідних сенсорних приладів (наприклад, для вимірювання точної кількості енергії споживання обігріву) чи можливості їх встановлення, у даній роботі було використані сторонні відкриті дані.

REFIT – це бази даних різноманітних вимірювань, проведених у двадцяти будинках в межах наукового проекту *REFIT: Smart Homes and Energy Demand Reduction* [155]. Проект *REFIT* (www.refitsmarthomes.org) проводив дослідження з 2013 по 2015 рік, у якому 20 будинків у Великобританії були модернізовані до розумних будинків шляхом встановлення пристроїв, включаючи розумні лічильники, програмовані термостати, програмовані радіаторні клапани, датчики руху, датчики дверей та віконні датчики.

Серед доступних наборів даних у даній роботі використовується *REFIT Smart Home dataset*, зібраний університетом м. Лофборо, Англія [96]. Цей набір даних включає дані про будівлі, споживання газу, внутрішню температуру повітря, місцеві кліматичні дані та інші вимірювання датчиків для 20 будинків *REFIT*.

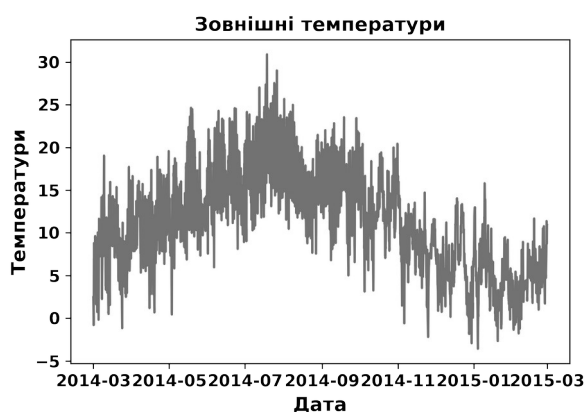


Рис. 2.1. Зовнішні температури

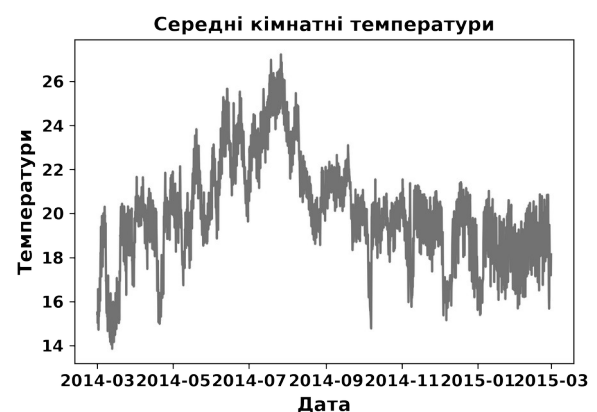


Рис. 2.2. Середні кімнатні температури



Рис. 2.3. Середні температури на обігрівальних елементах

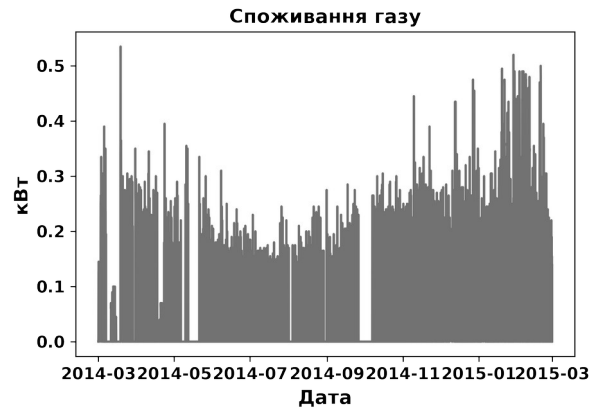


Рис. 2.4. Сумарне споживання газу

У роботі з метою побудови і дослідження статистичних моделей було розглянуто часові еволюції на протязі одного року (з 1 березня 2014 р. по 1 березня 2015 р.) таких параметрів: а) зовнішні температури повітря; б) середні кімнатні температури у будинках; в) температури на обігрівальних елементах у кімнатах; г) значення споживання газу у кВт. Вимірювання цих параметрів проводились через одногодичний часовий інтервал. На рис. 2.1-2.4 наведено графіки для даних з бази *REFIT*, які відповідають першому будинку.

В свою чергу, на рис. 2.5-2.6 зображені температурні дані, виміряні у лабораторії інтелектуальних автономних систем факультету електроніки та комп'ютерних технологій Львівського національного університету імені Івана Франка за період з 1 лютого 2021 року по 1 вересня 2021 року. Оскільки розробка моделей машинного навчання для нейроконтролерної системи (розділ 3) передбачає взаємодію з сенсорними даними, які можуть отримуватися у режимі реального часу, самі ці дані використовуються як для побудови нейромережових моделей, так і для їх тестування. Ці дані були накопичені окремою апаратною системою на базі мікрокомп'ютера Raspberry Pi3 Model B та температурних датчиків DS18B20.



Рис. 2.5. Температури у лабораторії



Рис. 2.6. Зовнішні температури поза лабораторією

2.2. Очищення та підготовка даних

Процес обробки сенсорних даних та їх статистичний аналіз необхідно здійснювати після етапу очищення, який передбачає: а) виявлення та заповнення пропущених значень й інтервалів; б) виявлення точкових викидів, аномалій чи аномальних підпоследовностей [156].

Відомі підходи до проблеми а), в яких для заповнення точкових пропусків використовується підхід вставки попереднього значення, а для інтервалу пропусків — лінійна інтерполяція [157] чи ковзне середнє [158]. Наприклад, на рис. 2.7 показаний результат застосування ковзного середнього для заповнення 5-ти значень температури у лабораторії, де ширина ковзного вікна $w = 6$. Недоліками таких підходів є потреба у підборі оптимальної ширини ковзного вікна, що залежить від кількості пропусків, а при застосуванні лінійної інтерполяції — не враховується часові закономірності у даних на попередніх відрізках часу.

Для виявлення точкових викидів та аномалій у сенсорних даних застосуємо підходи на базі *STL*-декомпозиції [159] та ізольованого лісу [160].

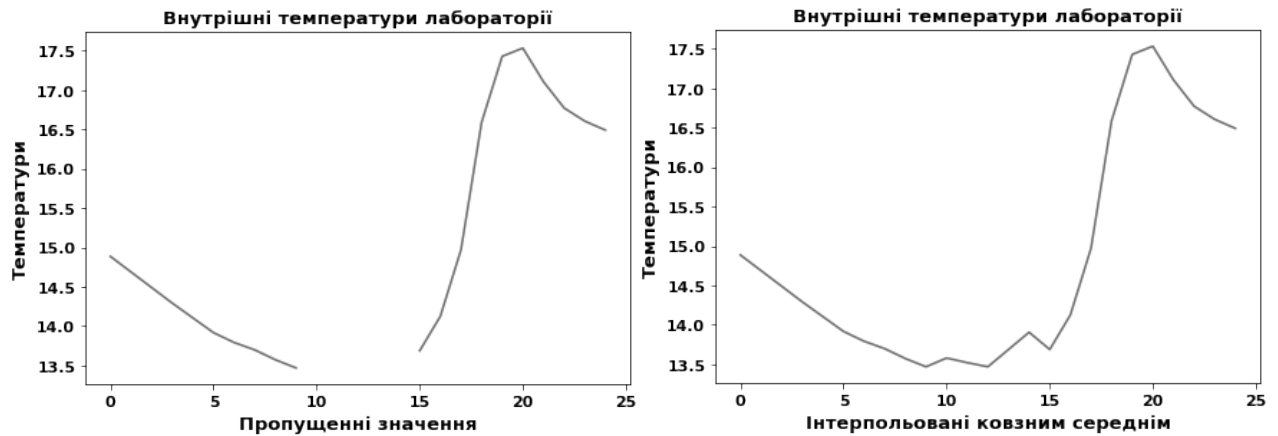


Рис. 2.7. Інтерполяція ковзним середнім

Для демонстрації ефективності цих методів додамо 1.5% аномальних точок (викидів) шляхом збільшення на деяку величину $\delta\tau = \pm 10$ випадкових температур у лабораторії за період з 1 березня по 1 квітня 2021 року (рис. 2.8). На рис. 2.8 (1) зображені додані аномалії, а на рис. 2.8 (2) результати адитивної *STL*-декомпозиції — графік виділеного тренду t_{rt} , сезонна компонента s_{rt} та різниці r_t , які визначаються як $r_t = y_t - t_{rt} - s_{rt}$, де y_t — температурний ряд. За розподілом різниць r_t та визначеним пороговим значенням $\tau \in r_t$ можна виділяти аномальні значення, проте оцінка порогу потребує візуалізації експериментів.

Результати застосування ансамблевого методу ізольованого лісу з параметром відсотку аномалій 1.5% наведені на рис. 2.9. Алгоритм ізольованого лісу є методом навчання без учителя та ґрунтується на побудові дерев рішень. У ізольованому лісі дані випадкової вибірки обробляються в деревоподібній структурі на основі випадково вибраних ознак. Зразки, які розташовуються глибше у дереві, є аномаліями з меншою ймовірністю, оскільки для їх ізоляції потрібно більше розділень. Аналогічно, зразки, які в кінцевому підсумку мають коротші гілки, вказують на аномалії, оскільки дереву було легше відокремити їх від інших спостережень.

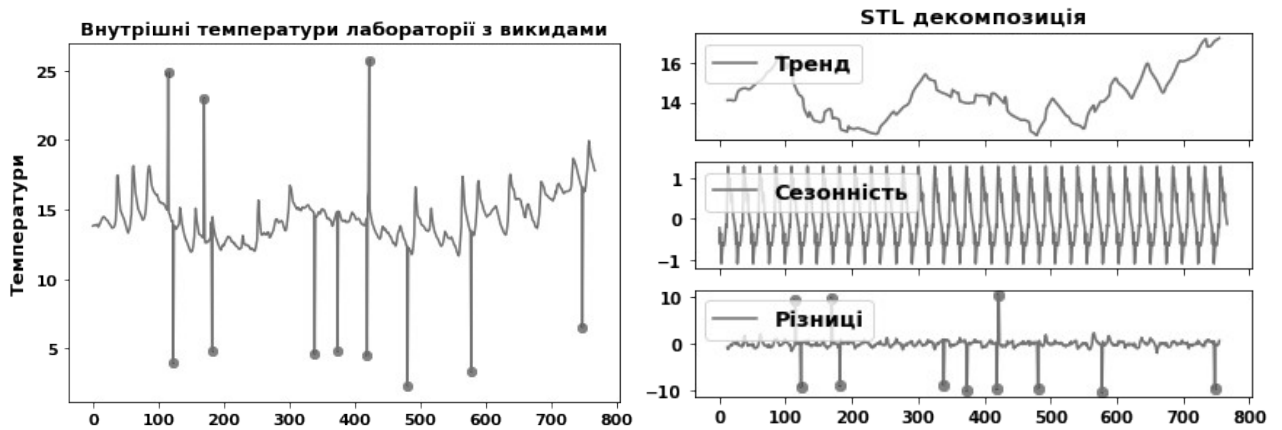


Рис. 2.8. Виявлення аномалій за *STL*-декомпозицією

Обидва з використаних методів в даному випадку дають приблизно однакову точність за винятком неправильно визначеної додаткової точки у методі ізольованого лісу.



Рис. 2.9. Виявлення аномалій за методом ізольованого лісу

Хоча існують підходи, що ґрунтуються на прогностичних моделях [161], кластерному аналізі [162] та автокодувальниках [163], вони потребують додаткових досліджень для конкретних даних та є більш обчислювано складними.

2.3. Регресійні моделі та статистичний аналіз

Кліматичні дані ІБ, а також зв'язані з ними вимірювання споживання енергії обігріву зумовлюють необхідність виявлення структури взаємозалежностей, які відіграють важливу роль у побудові прогностичних моделей. Автоматизація такого процесу дозволяє надавати користувачу актуальну інформацію як про кліматичний стан ІБ, так і доповнювати загальну статистику щодо використання пристроїв обігріву, або охолодження приміщень. З цією метою розглянемо процес побудови статистичних регресійних моделей даних на прикладі *REFIT* та розподілів температур у лабораторії.

На рис. 2.10 показані діаграми розмаху даних (відповідні рис. 2.1-2.4) ІБ *REFIT*: зовнішні температури, кімнатні температури, температури на обігрівальних елементах та споживання газу.

Як можна побачити, зовнішні температури демонструють передбачувану річну поведінку з виділеним піком на теплу пору року, динаміку якої можна використовувати у статистичному моделюванні.

Розподіл середніх кімнатних температур свідчить про наявність позитивної кореляції з зовнішніми температурами, хоча для більш точного аналізу можна розглядати такі залежності покімнатно. Оскільки протягом літнього періоду обігрів є відсутній, регресійна модель між незалежними зовнішніми та залежними внутрішніми температурами дозволяє встановити наближену оцінку теплового характеру будинку в сенсі його теплозберігаючих властивостей. Розподіл значень температур на обігрівальних елементах протягом теплого сезону дає підстави знехтувати ними, оскільки температури на батареях наближено рівні з внутрішньокімнатними. Проте, протягом холодної пори року такі виміри можна використовувати у регресійних моделях з урахуванням споживання газу.

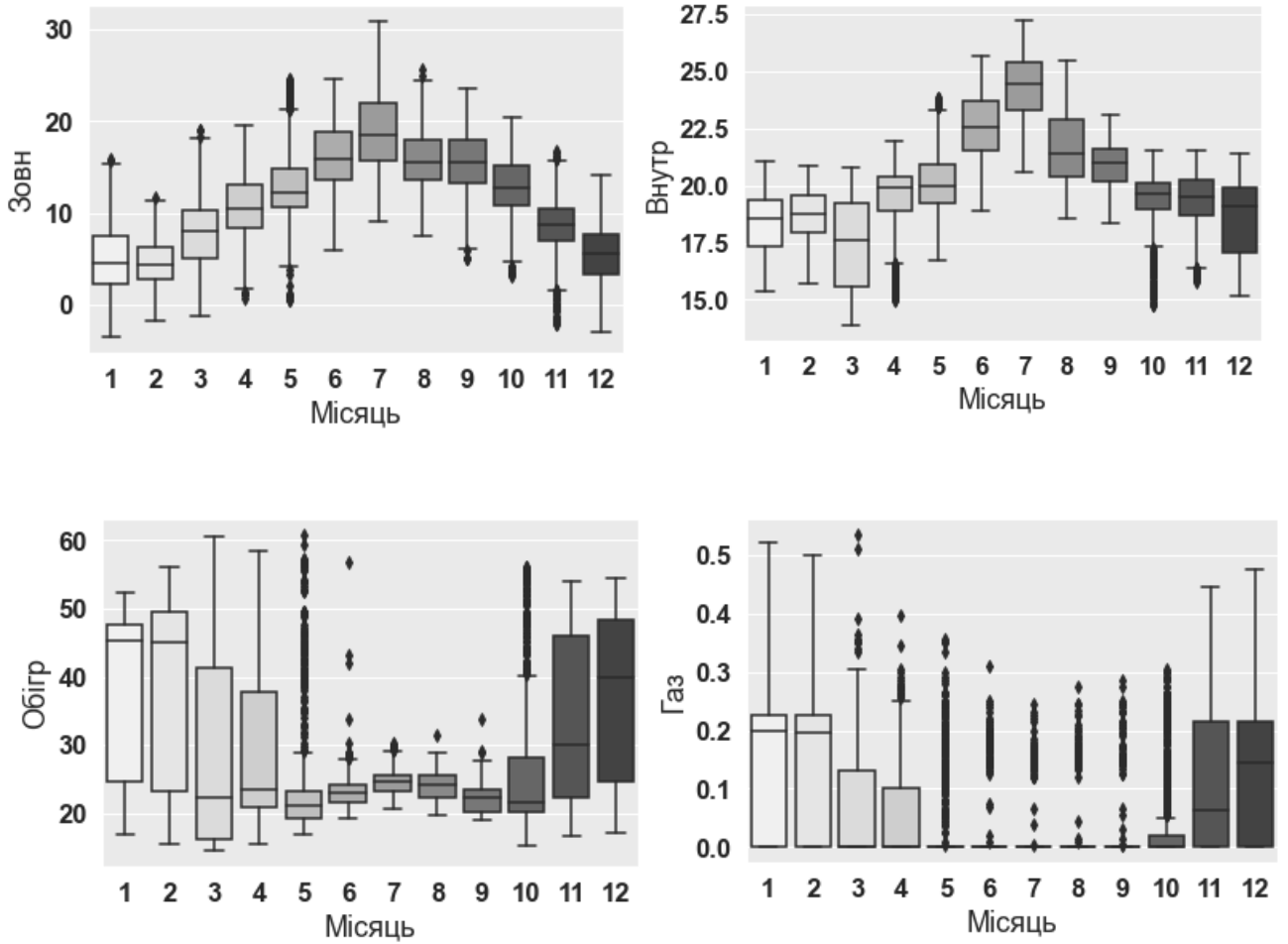
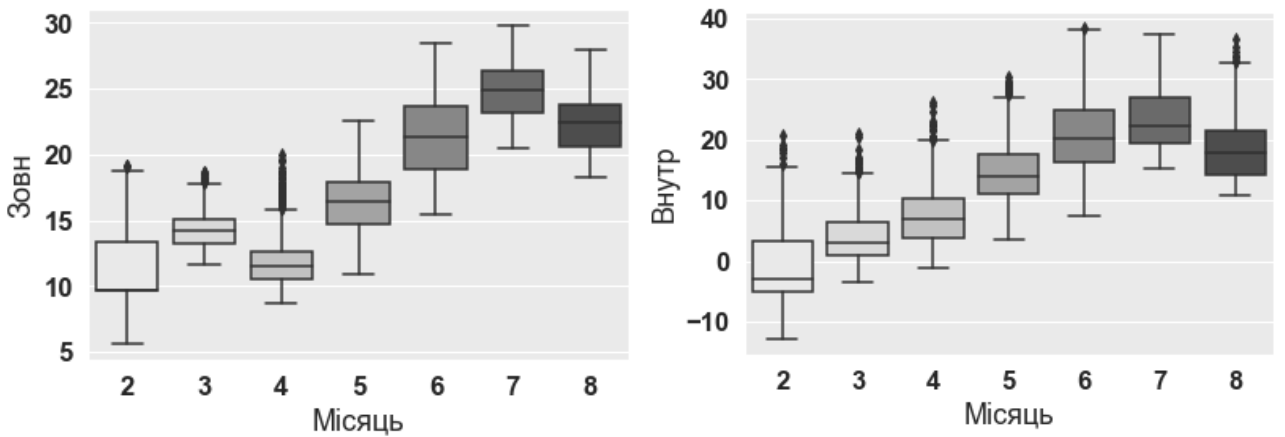
Рис. 2.10. Діаграми розмаху *REFIT*

Рис. 2.11. Діаграми розмаху даних лабораторії

Розподіл значень температур на обігрівальних елементах протягом теплого сезону дає підстави знехтувати ними, оскільки температури на

батареях наближено рівні з внутрішньокімнатними. Проте, протягом холодної пори року такі виміри можна використовувати у регресійних моделях з урахуванням споживання газу. В свою чергу, місячні розподіли використання газу протягом літнього періоду надають інформацію для задачі дезагрегації (визначення конкретних споживачів газу на базі сумарних значень та виявлених ознак), що є корисним для аналізу економії та оцінки теплової поведінки ІБ.

На рис. 2.11 зображені місячні статистики температур, виміряних у лабораторії. Динаміка даних протягом літнього періоду також може використовуватися для оцінки теплової інерції будівлі.

Відтак, розглянемо процес побудови та відбору регресійних моделей для встановлення залежностей між даними.

2.3.1. Регресійні моделі для зовнішньої та внутрішньої температур

Розглянемо побудову лінійної регресійної моделі виду $t_{ind} = f(t_{out}; \beta) + \varepsilon_t$, де t_{out} — зовнішні температури протягом літнього періоду, t_{ind} — середні кімнатні температури, $f(\cdot)$ — регресійна функція, β — параметри моделі, а ε_t — випадкова похибка. Така регресійна модель не лише дозволяє встановити залежність між температурами, а й наближено оцінити теплову інерцію будинку без наявності обігріву.

На рис. 2.12 показані діаграма розсіювання та кореляційна матриця, на основі чого можна зробити висновок про наявність лінійного зв'язку між незалежною t_{out} та залежною t_{ind} змінними. Графіки були отримані для даних *REFIT*.

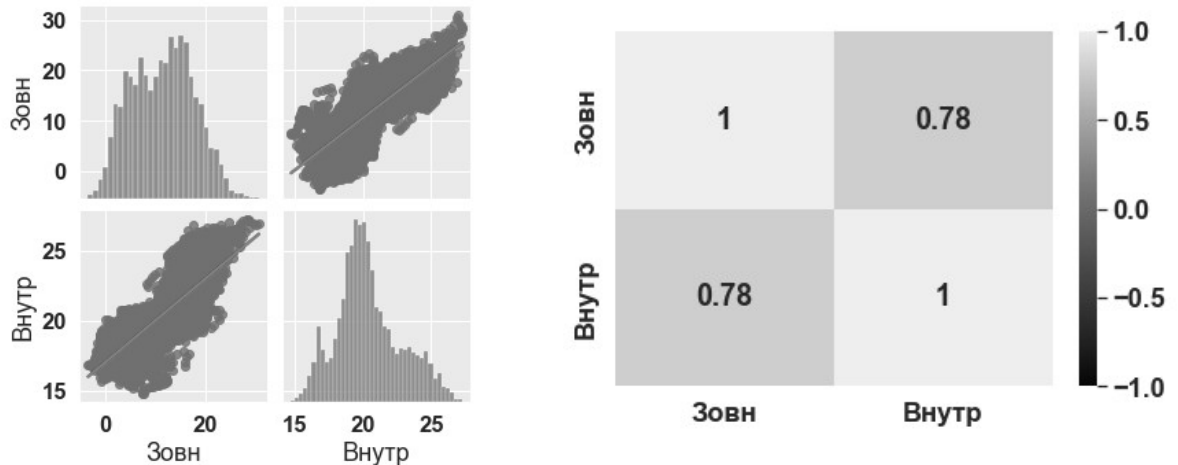


Рис. 2.12. Діаграма розсіювання зовнішніх та внутрішніх температур *REFIT*

Вибір типу лінійної регресійної моделі (простої лінійної, LASSO, Ridge, ElasticNet) [164] можна автоматизувати шляхом ітеративного підбору з використанням методу перехресного затвердження [165]. Наприклад, встановивши критерій відбору на основі скоректованого коефіцієнту детермінації $R^2 - adjusted$, інформаційних критеріїв AIC , BIC [166] та використавши k -кратне перехресне затвердження алгоритм вибору моделі складатиметься з таких кроків: 1) ініціалізація моделей $model[i] = model_init()$; 2) для кожної моделі $model[i]$: а) розбиття вибірки $D = [t_{out}, t_{ind}]$ на k підвбірок для реалізації перехресного затвердження; б) оцінка моделі на k -підвбірках за критерієм $R^2 - adjusted$, або AIC , BIC та усереднення результату; в) перевірка припущень для фінальної моделі лінійної регресії: тестування на гетероскедастичність (тест Бройша-Пагана), нормальний розподіл похибок ε_t (тест Андерсона-Дарлінга), автокореляція похибок ε_t (тест Дурбіна-Ватсона), лінійність (тест Гарві-Кольера).

Здійснивши чисельні експерименти за вищенаведеним алгоритмом, в результаті була отримана проста лінійна регресійна модель виду (рис. 2.13):

$$\ln(t_{ind}) = 2.84 + 0.014 * t_{out} + \varepsilon_t, \quad (2.1)$$

де розподіл похибок ε_t показаний на гістограмі рис. 2.13 (2), а коефіцієнт детермінації $R^2 - adjusted = 0.6$. Логарифмування залежної змінної було проведене задля виправлення гетероскедастичності похибок ε_t , виявлених під час експериментів.

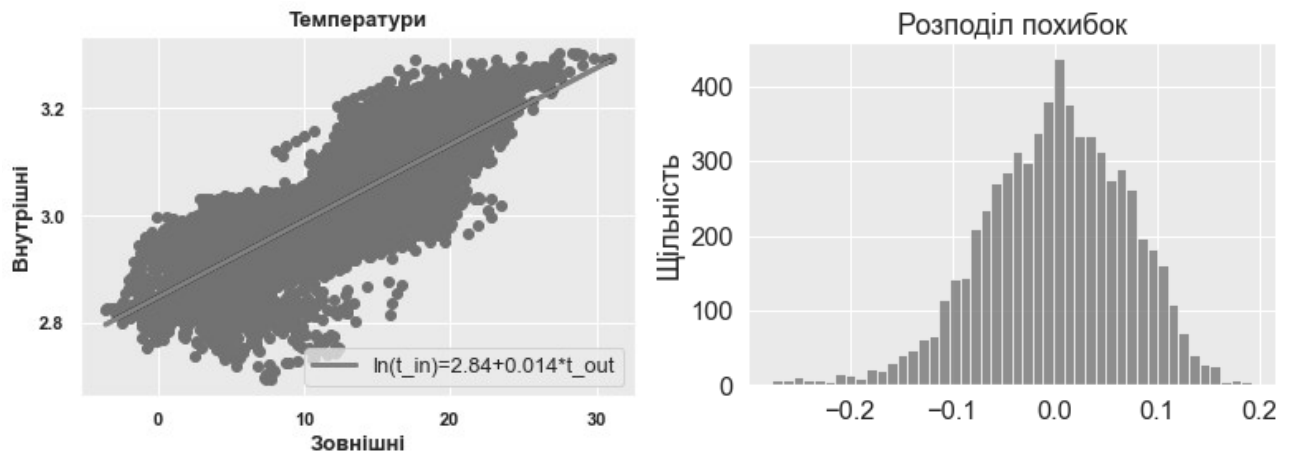


Рис. 2.13. Регресійна пряма для температур *REFIT* та розподіл похибок

Значення $R^2 - adjusted$ свідчить про те, що залежна величина t_{ind} кімнатних температур не в повній мірі залежить від зовнішньої температури, тобто присутні інші фактори (явища конвекції тощо). Зокрема, величина $R^2 - adjusted$ залежить від теплофізичних параметрів будинку. Очевидно, чим менший кут нахилу регресійної прямої, тим більшою є теплова інерція будинку.

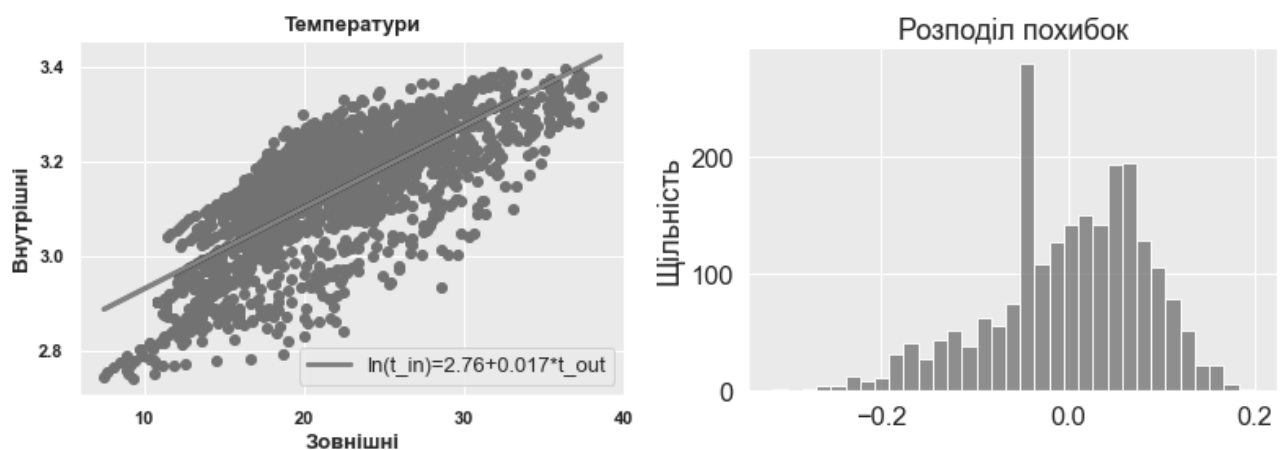


Рис. 2.14. Регресійна пряма для температур у лабораторії та розподіл похибок

Для температур, виміряних у лабораторії (рис. 2.5-2.6), лінійна регресійна модель матиме вигляд (рис. 2.14 (1)):

$$\ln(t_{ind}) = 2.76 + 0.017 * t_{out} + \varepsilon_t, \quad (2.2)$$

з розподілом похибок (рис. 2.14 (2)) та коефіцієнтом детермінації $R^2 - adjusted = 0.6$. Тут похибки ε_t не є нормально розподіленими, що може пояснюватися наявністю аномалій та викидів, не виявлених під час очищення даних. Позаяк, у моделі забезпечена скедастичність, лінійність та відсутність автокореляцій. Крім лінійних регресійних моделей можна застосовувати більш складні непараметричні аналоги, такі як регресія з градієнтним підсиленням [167]. Проте, інтерпретація такої моделі є складною [168], в ній відсутня можливість екстраполяції та необхідно проводити оптимізацію гіпер-параметрів.

2.3.2. Регресійна модель для споживання газу та температури обігрівальних елементів

Наступна регресійна модель — моделювання взаємозв'язку між сумарним споживанням газу *gas* та температур на обігрівальних елементах *rad* протягом зимового періоду. Ті пари значень, де споживання газу рівне нулю, були видалені з вхідного масиву, оскільки вони не є інформативними для моделювання.

Як можна побачити з діаграми розсіювання рис. 2.15, споживання газу та температури не демонструють чіткої лінійної взаємозалежності, тому в даному випадку можна використати непараметричний регресор на основі градієнтного підсилювання. Також, таку залежність можна моделювати поліноміальною лінійною регресією, що потребує оптимального підбору степенів поліному.

Кореляційна матриця (коефіцієнти Пірсона) показує позитивну кореляцію між цими змінними (збільшення кількості споживання газу відповідає збільшенню температури на батареї).

На рис. 2.16 продемонстрований результат побудови регресійних моделей на базі полінома другого ступеня та градієнтного підсилення з використанням бібліотеки *XGBoost* [169].

Лінійна регресійна модель на базі поліному другого степеня вигляд (рис. 2.16 (1)):

$$rad = 22.1 + 164.9 * gas + -237.8 * gas^2 + \varepsilon_t, \quad (2.3)$$

де коефіцієнт детермінації $R^2 - adjusted = 0.6012$, а значення середньої абсолютної відсоткової помилки (*MAPE*) — 0.087.

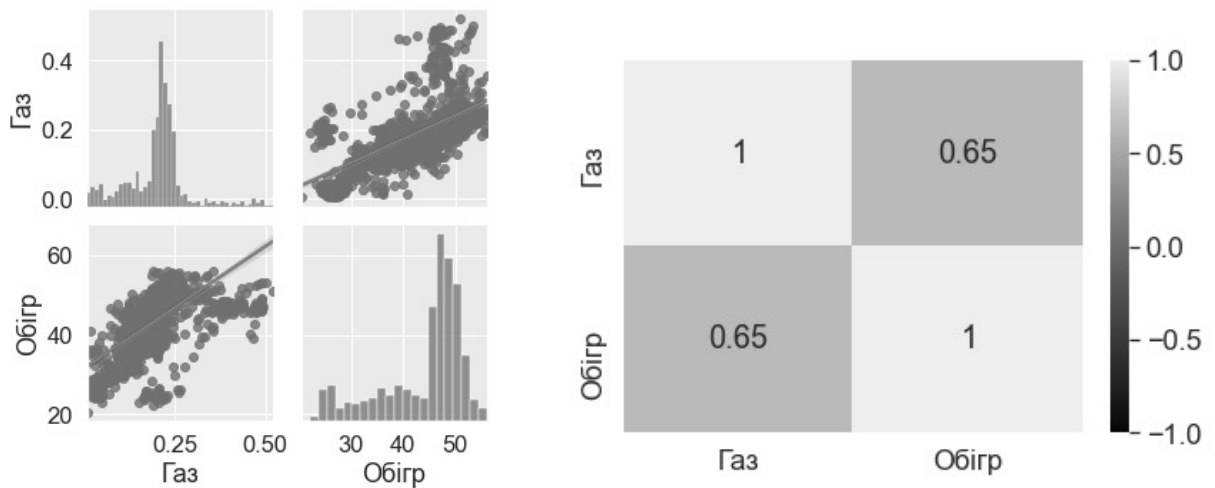


Рис. 2.15. Діаграма розсіювання даних споживання газу та температур на батареях *REFIT*

Регресійна модель на основі градієнтного підсилення (рис. 2.16 (2)) має такі параметри: максимальна глибина побудованого дерева — 5, темп навчання — 0.3, значення середньої абсолютної відсоткової помилки

($MAPE$) — 0.079. Остаточне дерево градієнтного підсилення зображене на рис. 2.17. Тут параметр f_0 відповідає за змінну gas .

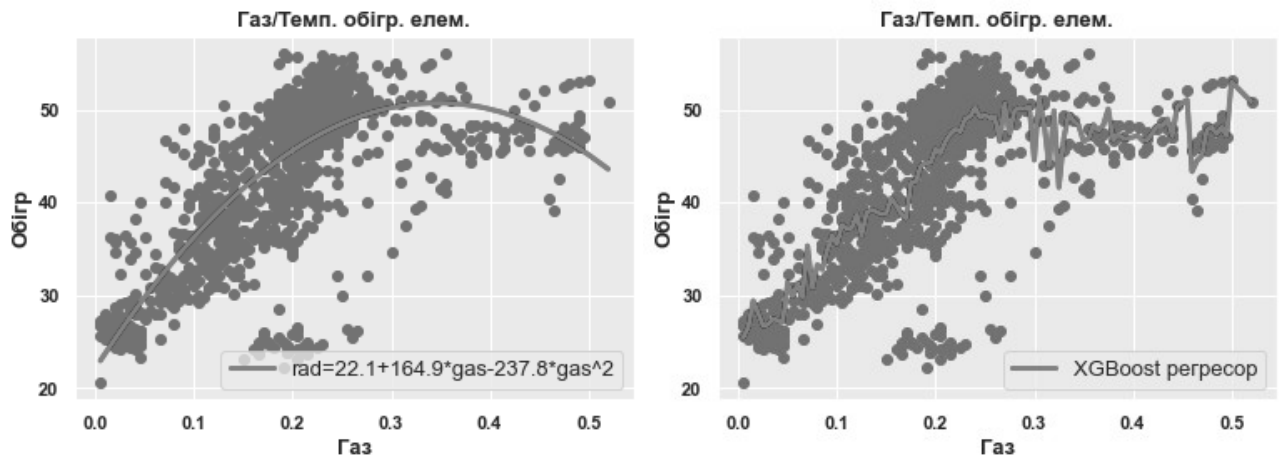


Рис. 2.16. Регресійні криві: поліноміальна та на основі градієнтного підсилення

Грунтуючись на показнику $MAPE$, регресійна модель градієнтного підсилення є точнішою та не вимагає поліноміальної трансформації як у випадку простої лінійної регресії. Позаяк, регресор на основі градієнтного підсилення містить значну кількість гіперпараметрів, які необхідно додатково оптимізувати.

Дану модель можна використовуватися для прогнозування температури на батареях за відомим сумарним споживанням газу у будинку, проте її інтерпретація, на відміну від лінійної регресії, є складною, що є недоліком.

В якості остаточної моделі для розгортання в контексті API для системи ІБ варто обрати регресор з градієнтним підсиленням, оскільки: а) процес підбору гіперпараметрів можна автоматизувати застосуванням AutoML [170] підходу; б) дана модель не вимагає чіткої лінійної залежності між змінними на відміну від лінійної регресії.

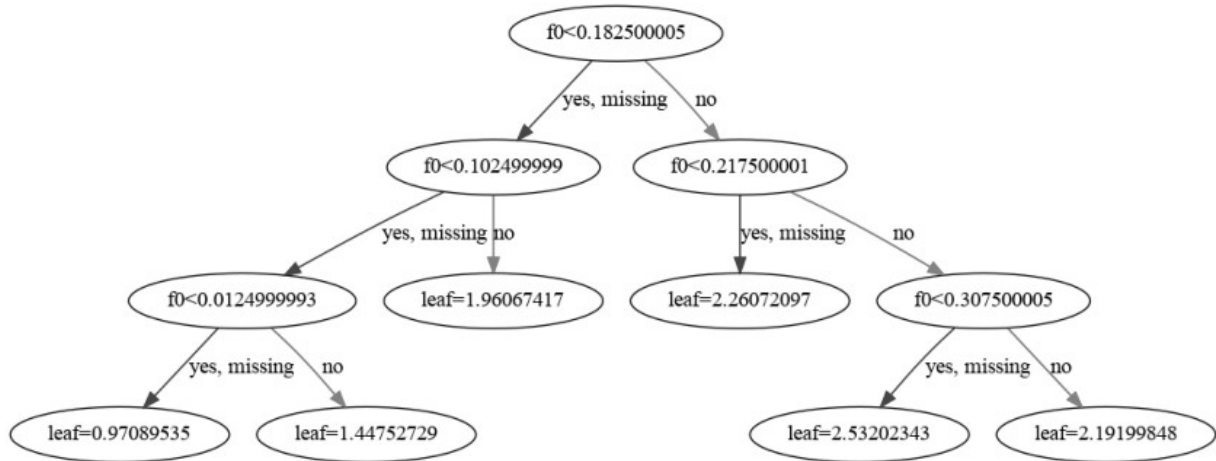


Рис. 2.17. Дерево моделі градієнтного підсилення

2.3.3. Регресійна модель для внутрішньої, зовнішньої температури та споживанням газу

Розглянемо регресійну модель виду $t_{ind} = f(t_{out}, gas; \beta) + \varepsilon_t$, що описує зв'язок між незалежними параметрами споживання газу gas , зовнішніми температурами t_{out} та залежними середніми внутрішніми температурами t_{ind} протягом зимового періоду. Попередньо триплети даних з нульовим споживанням газу були виключені з аналізу. Оскільки у моделі присутні дві змінні, їх можна протестувати (оцінка коефіцієнту інфляції дисперсії [171]) на наявність мульти-колінеарності (табл. 2.1). Значення цього коефіцієнту є менші за 10, що говорить про відсутність мультиколінеарності між сумарним споживанням газу та зовнішніми температурами.

На рис. 2.18 наведені діаграма розсіювання та кореляційна матриця, що свідчать про незначні кореляції та відсутність чітко вираженої лінійної залежності між даними.

Відтак, для побудови прогностичної регресійної моделі використаємо алгоритм градієнтного підсилення (максимальна глибина побудованого дерева — 5, кількість дерев — 50), дерево якого зображене

на рис 2.19 (параметри f_0 та f_1 відповідають t_{out} та gas), а значення середньої абсолютної відсоткової помилки ($MAPE$) — 0.032.

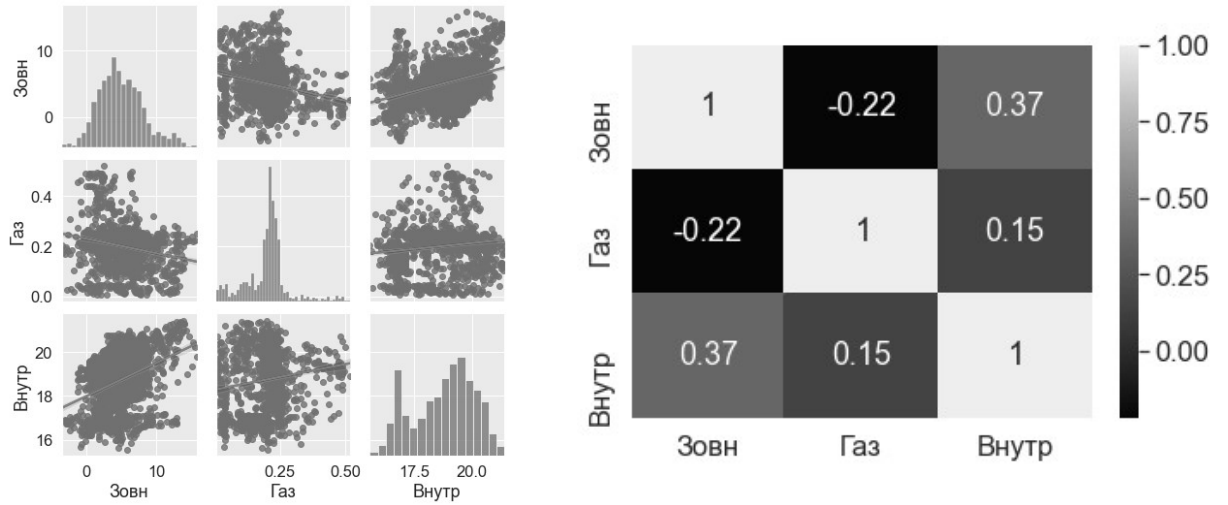


Рис. 2.18. Діаграма розсіювання даних споживання газу та температур *REFIT*

Змінна	Коефіцієнт інфляції дисперсії
t_{out}	3,99
gas	7,55

Табл.2.1. Оцінка коефіцієнту інфляції дисперсії

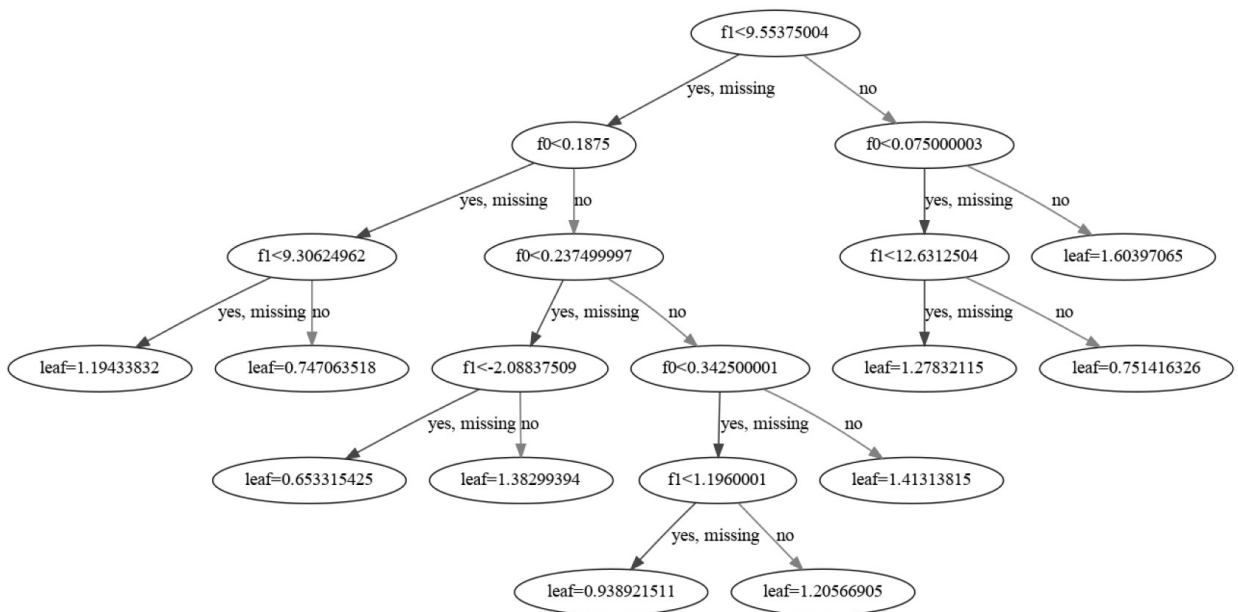


Рис. 2.19. Дерево моделі градієнтного підсилення

На відміну від регресійної моделі зовнішніх і внутрішніх температур протягом літа, зимовий період характеризується відсутністю вираженої прямої кореляції, що пов'язано з обігрівом приміщень. Врахування параметра сумарного споживання газу дозволяє більш точно змодельовати цей зв'язок, проте потребує окремого дослідження в контексті розв'язування задачі дезагрегації.

2.4. Модель дезагрегації газу та кластерний аналіз

У розглянутих вище регресійних моделях фігурує сумарне споживання газу, що містить декілька додатних складових, що відповідають не тільки обігріву приміщень, а й використанню на інші побутові потреби. Наприклад, серед них є складові, які відповідають використанню газу для приготування їжі, душу тощо. З метою виділення складових компонент з сумарної величини, був розроблений простий підхід на базі кластерного аналізу даних [10].

Нехай $G = \{g_t\}$ — сумарне споживання газу у будинку, де g_t — вимірне значення у час t , а G_s — споживання газу протягом сезону s . На основі цих даних можна виразити та проаналізувати споживчу поведінку за допомогою кластерного аналізу, тобто кожен центр c_s^k обчисленого кластеру k визначається групою ознак $\gamma_s^{(k)}$, виділеними з G_s і об'єднаними деяким спільним патерном. Масив G_s можна представити як множину $G_s = \{g_1, g_2, \dots, g_n\}$, де g_i , $i = 1, \dots, n$ — добові розподіли споживання газу. Також, групу ознак $\gamma^{(k)}$ можна виділити з усього часового ряду G . Кластерний аналіз можна застосувати до матриці подібності, побудованої за виділеними ознаками часового ряду [172].

Процедура кластеризації спрощує розробку методу дезагрегації газу: дослідження виявлених патернів у G_s для отримання наборів ознак, що відповідають часовим інтервалам t_s та пороговим значенням ε_s для виокремлення типу споживання. Відтак такий механізм дозволяє досліджувати поведінку споживачів.

Таким чином, задача дезагрегації складається з таких етапів: 1) обчислення кластерів ознак $\gamma_s^{(k)}$ з використанням методів кластерного аналізу (k -середніх, або нечіткої кластеризації [10]) та 2) на основі отриманих кластерів необхідно ідентифікувати тип споживання газу та виокремити відповідні складові з сумарного ряду.

Визначимо ознаки як різниці між попереднім та наступним значенням [10]:

$$\gamma_t = g_{t+1} - g_t \quad (2.4)$$

як додатні та від'ємні магнітуди у зміні споживання. Цей набір ознак можна доповнювати послідовними значеннями магнітуд та їх градієнтами тощо.

Позначимо масиви з додатними ознаками як $\boldsymbol{\gamma}_+ = [\gamma_1, \gamma_2, \dots, \gamma_m]^+$, а з від'ємними ознаками — $\boldsymbol{\gamma}_- = [\gamma_1, \gamma_2, \dots, \gamma_m]^-$, що отримані на базі деякого середнього добового профілю

$$\mathbf{P}_s = \frac{1}{n} \sum_i \omega_i g_i \quad (2.5)$$

де ω_i — вагові коефіцієнти в межах $(0, 1]$, що посилюють ефект більш типового ряду серед усіх масивів. Обчислення вагових коефіцієнтів ω_i можна здійснювати, наприклад, на базі Евклідової дистанції між відповідними рядами з G_s та їх поелементно усередненим рядом $\tilde{g}_s = \frac{\sum_{i=1}^n g_i}{n}$:

$$d_{s,i} = \|g_s - g_i\|_{l_2}, \quad \omega_i = 1 - \frac{d_{s,i} - \min(d_{s,i})}{\max(d_{s,i}) - \min(d_{s,i})}. \quad (2.6)$$

Кожен масив ознак $\boldsymbol{\gamma}_- = [\gamma_1, \gamma_2, \dots, \gamma_m]^-$ та $\boldsymbol{\gamma}_+ = [\gamma_1, \gamma_2, \dots, \gamma_m]^+$ підлягає кластеризації за допомогою одного з відомих алгоритмів

(агломеративна ієрархічна кластеризація, метод k-середніх, нечітка кластеризація тощо) [173].

Розглянемо середньодобовий, максимальний і мінімальний профілі P_s , обчислені з використанням (2.6) за зимовий та літній сезони (рис. 2.20). Застосувавши процес виділення ознак (2.4) до середньодобового розподілу за зимовий і літній період та ієрархічну кластеризацію, були отримані результати, зображені на рис. 2.21. Кількість кластерів $n_clusters = 2$, оскільки протягом літа було важливим виділити мінімум 2 споживача, для яких споживання відповідає побутовим потребам. Протягом зими $n_clusters = 2$, що відповідають споживанню газу для обігріву та іншим побутовим потребам (душ, приготування їжі тощо).

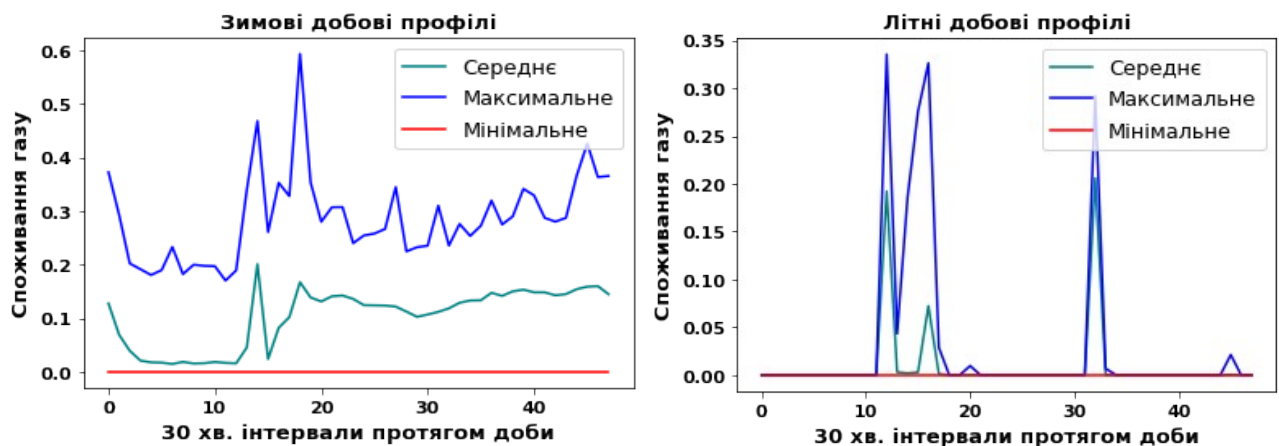


Рис. 2.20. Зважені середньо-добові профілі споживання газу

Взагалі кажучи, кожна додатна магнітуда з довільного кластера з центроїдом $+c_k$ наближено збігається з від'ємною магнітудою з кластера з центроїдом $-c_k$. Тут додатні та від'ємні кластери приблизно рівні за кількістю ознак; ці кластери визначають стани включення/вимкнення газу, або переходи між відповідними режимами.

В загальному, цей процес можна описати такою моделлю споживання:

$$X^k = \langle \gamma_+, \operatorname{argmin}_{\gamma_-} \|\gamma_+ - (-C)\|, t_s, \varepsilon_s \rangle, \quad (2.7)$$

де $\gamma_+ = \text{mean}(\gamma \in +C_k)$ — середня магнітуда з кластеру додатних ознак $+C_k$ з центроїдом $+c_k$, $\gamma_- \in -C$, $-C$ — множина всіх кластерів з негативними магнітудами, t_s — часові інтервали, коли відбувалось споживання газу X^k споживачем, $\|\cdot\|$ — деяка норма.

На рис. 2.21 кластер з найбільшими значеннями ознак (літо) визначає споживача X^1 (ймовірно душ/приготування їжі), X^2 — короткотерміновий споживач (миття посуду, рук тощо). На базі цих припущень, зимові кластери дозволяють виділити споживачів X^1 та X^2 , де X^2 — відповідає обігріву.

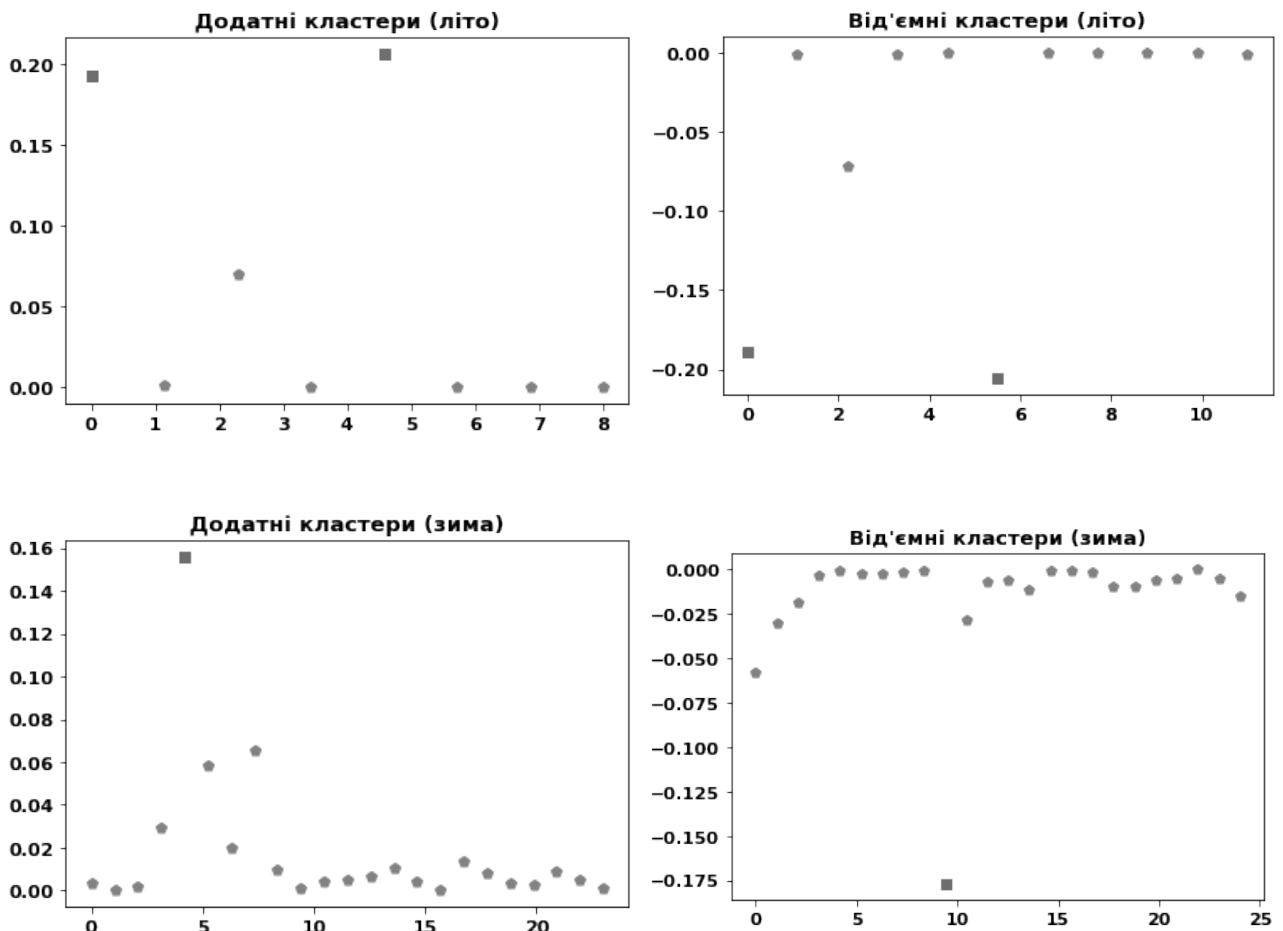


Рис. 2.21. Результати ієрархічної кластеризації

На жаль, у першому наближенні даний підхід, що ґрунтується на різницевих ознаках (2.4) дозволяє зробити лише грубу оцінку типу

споживання. Проте, за подальшого дослідження розглянутої ідеї можна доповнювати задля уточнення методу.

2.5. Розробка нейромережових прогностичних моделей для нейроконтролера

У третьому розділі роботи представлений апаратно-програмний комплекс для системи ІБ, що передбачає обробку сенсорних даних та можливість їх короткотермінового прогнозування нейронною мережею, розгорнутою на нейроконтролері. Для програмної реалізації останнього розглянемо процес побудови нейромережі на базі рішення TensorFlow 2.

2.5.1. Обробка та трансформація даних для нейромережі

Кліматичні дані інтелектуального будинку представляють собою масиви часових рядів $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m]$, або багатовимірний часовий ряд, де кожен $\mathbf{y}_k = [y_k^1, y_k^2, \dots, y_k^l]$ — часовий ряд зі значеннями конкретного кліматичного параметра, m — кількість кліматичних параметрів, l — довжина часового ряду.

Такий масив даних можна представити у матричному вигляді

$$\mathbf{Y} = \begin{bmatrix} y_1^1 & \cdots & y_1^l \\ \vdots & \ddots & \vdots \\ y_m^1 & \cdots & y_m^l \end{bmatrix}. \quad (2.8)$$

Проблема моделювання часових рядів може бути сформульована в рамках задачі навчання з учителем, що дає змогу використовувати таке переформулювання для застосування лінійних і нелінійних алгоритмів машинного навчання. Для цього кожен з векторів-рядків $\mathbf{y}_k = [y_k^1, y_k^2, \dots, y_k^l]$ у (2.8) реструктуризуємо до наступних послідовностей:

$$\begin{bmatrix} [y_k^1, \dots, y_k^v] & [y_k^{v+1}, \dots, y_k^{v+p}] \\ [y_k^2, \dots, y_k^{v+1}] & [y_k^{v+2}, \dots, y_k^{v+p+1}] \\ \dots & \dots \end{bmatrix}, \quad (2.9)$$

де v — довжина вектору предиктора, а p — кількість кроків для прогнозу, або довжина цільового вектора. В подальшому для спрощення, у роботі довжина зсуву по прогностичному вектору буде рівна одному часовому кроку.

Таким чином, представлення (2.9) можна подати у вигляді навчального набору даних \mathbf{X}_{train} та \mathbf{Y}_{train} , в якому перша колонка у (2.9) відповідає матриці ознак, а друга колонка — матриці міток (класів):

$$\mathbf{X}_{train} = \begin{bmatrix} [y_k^1, \dots, y_k^v] \\ [y_k^2, \dots, y_k^{v+1}] \\ \dots \end{bmatrix}, \quad \mathbf{Y}_{train} = \begin{bmatrix} [y_k^{v+1}, \dots, y_k^{v+p}] \\ [y_k^{v+2}, \dots, y_k^{v+p+1}] \\ \dots \end{bmatrix}. \quad (2.10)$$

Компоненти, або вектори-рядки у \mathbf{X}_{train} позначимо як x_k^v , що відповідають деякій v -ознаці k -того ряду. Для формування матриць (2.10) необхідно врахувати той факт, що перші v -значень у \mathbf{X}_{train} визначають наступні p -значень у \mathbf{Y}_{train} . Це означає, що при ініціалізації цих матриць на першому кроці

$$\begin{aligned} x_k^1 &\Leftarrow x_k^{v+1}, x_k^v &\Leftarrow x_k^{2v} \\ y_k^{v+1} &\Leftarrow y_k^{v+p}, y_k^{v+p} &\Leftarrow x_k^{2v+p}. \end{aligned} \quad (2.11)$$

Підготовка матриць (2.10) з використанням (2.11) реалізується застосуванням методу ковзного вікна.

Відтак, після перетворення набору часових рядів до матричних представлень (2.7) та (2.9)-(2.10) наступним кроком необхідно здійснити переформатування отриманих матриць у тривимірні тензори для побудови вхідних даних LSTM, GRU та CNN блоків.

2.5.2. Перетворення даних у тривимірні тензори

Оскільки вхідні блоки LSTM, GRU та CNN нейромереж на відміну від багат шарового перцептрона, де вхідні розмірності задані у вигляді матриці $A = (a_{ij})$ (рядки відповідають зразку, а стовпчики — значенням ознак), реалізовані для даних тривимірної розмірності $A = (a_{ijk})$ необхідно перетворити матриці (2.10) і (2.12) до відповідної структури. Тут розмірність задається параметрами [зразок, часовий крок, вектор ознак], а A — вхідний тензор даних.

Для моделювання часових рядів один зразок визначається однією часовою послідовністю, часовий крок — сталим кроком вимірювання показників, а вектор ознак — значеннями величин послідовності у кожній часовій мітці. В кожен момент часу багатовимірний ряд містить набір багатьох значень.

Для тренування рекурентних нейромереж на базі LSTM/GRU комірок, або гібридних нейромереж з першим згортковим шаром широко застосовується метод пакетного градієнтного спуску, коли на вхідний шар нейронів подається набір тривимірних тензорів A , при чому кількість нейронів у вхідному шарі не залежить від розмірності вхідних даних.

Таким чином, для перетворення (2.10) до тривимірного тензора необхідно додати ще один вимір:

$$\mathbf{X}_{train} = (y_{ij})_{r \times v} \Rightarrow \mathbf{X}_{train}^{3D} = (y_{kij})_{k \times r \times v}, \quad (2.12)$$

де r — кількість рядків, v — довжина ряду-предиктора, k — задає кількість зразків. Тензори \mathbf{X}_{train}^{3D} визначають структуру вхідних даних для LSTM/GRU та CNN комірок нейронної мережі.

2.5.3. Моделі на базі рекурентних нейронних мереж

Рекурентні нейронні мережі (РНМ) визначають широкий клас моделей для використання з даними, які містять складні часові залежності. Особливістю таких мереж є те, що обчислені під час тренування параметри залишаються сталими по мірі зміни вхідних даних в межах підпоследовностей. Вони встановлюють базову структуру та принципи навчання, спільні для різних видів РНМ: довгої короткотермінової пам'яті (LSTM), вентильного рекурентного вузла (GRU), гібридних архітектур (CNN-LSTM). Проектування моделей на базі РНМ для програмного забезпечення нейроконтролерних систем є одним з найбільш актуальним викликів у галузі створення IoT малопотужних пристроїв. З метою реалізації цього напрямку розглянемо складові компоненти і концепти РНМ.

Розглянемо задачу моделювання одновимірного ряду внутрішніх температур з використанням РНМ. Оскільки передбачається, що побудована у результаті такого моделювання РНМ буде розгортатися на продуктивному мікроконтролері сімейства STM32 F/H необхідно забезпечити мінімальну обчислювальну складність та розмір файлу моделі водночас досягнувши максимально можливої точності та уникнувши перенавчання моделі.

В якості даних для створення прототипу РНМ та її тестування в цьому випадку використовуємо частину внутрішніх температур у лабораторії (рис. 2.22(1) у проміжку з 1 березня 2021 року по 31 травня 2021 року, що містить 1488 вимірів, а на рис. 2.22(2) — гістограму розподілів значень. Ці дані становлять собою часовий ряд y .

В багатьох випадках, перед підготовкою нейромережевого моделювання ряд y варто перевірити на стаціонарність/нестаціонарність за допомогою розширеного тесту Дікі-Фулера і/або KPSS [174] тесту. При виявленні нестаціонарності ряду, одним з можливих рішень є диференціювання ряду $\Delta y = [\Delta y_2, \Delta y_3, \dots]$, де $\Delta y_i = y_{i+1} - y_i$ та проведення моделювання з Δy .

Після отримання прогнозованих значень $\Delta \tilde{\mathbf{y}}$ їх можна перетворити до $\tilde{\mathbf{y}}$ згідно перетворення $\tilde{y}_i = y_1 + \sum_i \Delta \tilde{y}_i$. Проте, недоліком такого підходу є проблема накопичення помилок прогнозування.

Перед нормалізацією та приведенням температурного ряду \mathbf{y} до матричної форми згідно (2.10) необхідно розділити \mathbf{y} на навчальну \mathbf{y}_{train} та тестувальну \mathbf{y}_{test} вибірки у деякій пропорції, наприклад 80% та 20%.

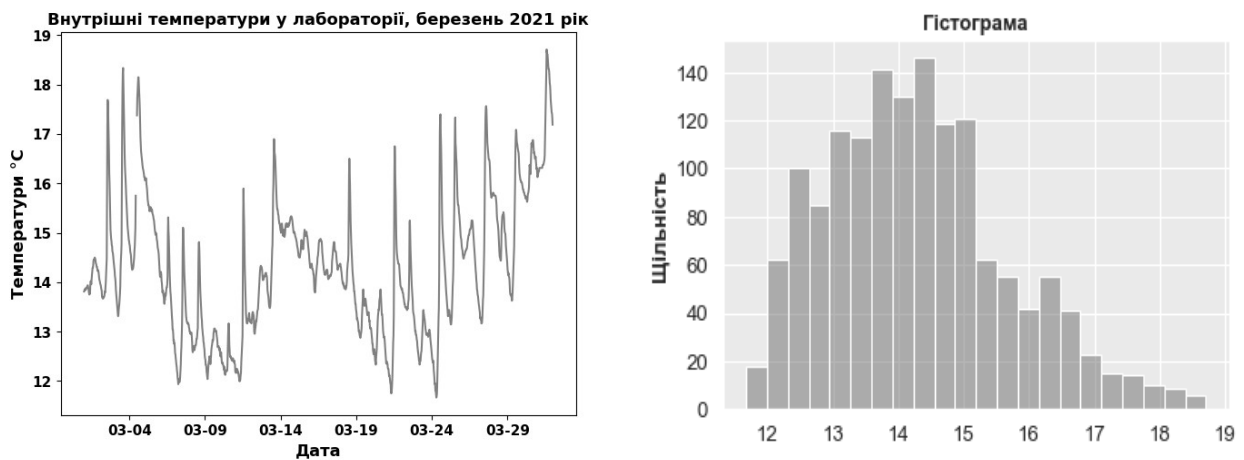


Рис. 2.22. Внутрішні температури за березень та відповідна гістограма

Для неймережевого моделювання з метою забезпечення коректного навчання РНМ, ряд \mathbf{y}_{train} необхідно нормалізувати, або стандартизувати. Тут застосуємо нормування ряду \mathbf{y}_{train} в межах діапазону 0..1 згідно формули:

$$\forall y^i \in \mathbf{y}_{train}, y^i = \frac{y^i - \min(\mathbf{y}_{train})}{\max(\mathbf{y}_{train}) - \min(\mathbf{y}_{train})}. \quad (2.13)$$

Нормування ряду \mathbf{y}_{train} навчальної вибірки відбувається перед нормуванням тестувальної вибірки \mathbf{y}_{test} , під час чого здійснюється визначення та значень $\min(\mathbf{y}_{train})$ і $\max(\mathbf{y}_{train})$, які потрібно

використовувати у (2.13) для нормування \mathbf{y}_{test} задля уникнення проблеми витоку даних.

Після нормування ряду за (2.13), нормалізовані набори температурних даних $\bar{\mathbf{y}}_{train}$ та $\bar{\mathbf{y}}_{test}$ підлягають перетворенню до матричних форм з використанням (2.10), де параметри довжини v вектора-предиктора та довжини p цільового вектора можна оцінювати на базі статистичного аналізу рядів, або динамічно в контексті підбору гіперпараметрів та конфігурацій нейронної мережі.

Після підготовки навчальної та тестувальної вибірок для РНМ встановимо гіпер-параметри та архітектури РНМ. Оскільки передбачається, що спроектована РНМ розгортається на мікроконтролері, обмежимося такими одношаровими архітектурами як LSTM, GRU та гібридною CNN-LSTM/GRU і дослідимо точність моделювання у межах задачі багатокрокового прогнозування температури.

Параметри РНМ для задачі навчання з учителем можна визначати за допомогою евристичних методів, або застосовуючи методи підбору гіперпараметрів, наприклад, з використанням генетичного алгоритму чи мета-евристичної оптимізації [175]. Такими параметрами можуть бути а) довжина вектора стану \mathbf{h}_i^i у i -тому шарі, б) розмір *batch* пакету даних для тренування, в) кількість епох тренування *epochs*, г) типи функції активації та помилки, г) розмірності вхідних тензорів (у тому числі значення p та v), д) коефіцієнт швидкості навчання α та алгоритм оптимізації для методу поширення помилки з часом.

У цьому розділі параметри конфігурації визначаються на базі генетичного алгоритму, де цільова функція задавалася як

$$\Phi(\mathbf{H}) = \frac{1}{N_v} \sum_{i=1}^{N_v} \|(model(\mathbf{y}^i; \mathbf{H}) - y^i)\|, \quad (2.17)$$

де N_v — кількість зразків у тестувальному наборі, $model(\mathbf{y}^i; \mathbf{H})$ — функція, яка здійснює прогноз на базі зразка \mathbf{y}^i за заданим набором \mathbf{H}

параметрів моделі РНМ. Для чисельного експерименту тут в якості таких параметрів було обрано довжину вектора стану h_t^i , розмір *batch* пакету даних для тренування та коефіцієнт швидкості навчання α , $\|\cdot\|$ — норма.

На рис. 2.23 зображена структура РНМ для моделювання температурного ряду y , де комірки (*cells*) у першому рекурентному шару RNN_1 можуть бути LSTM, або GRU типу.

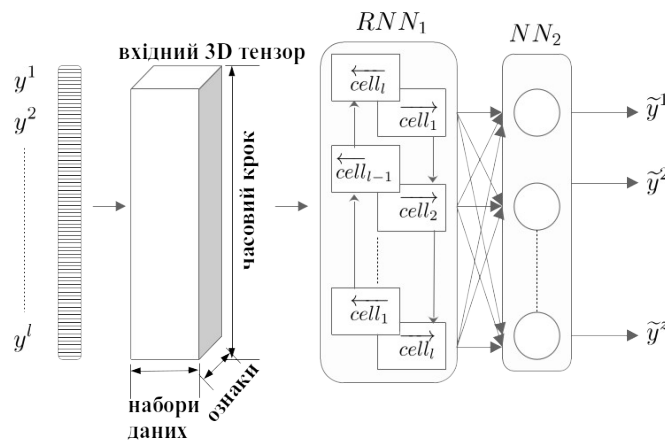


Рис. 2.23. Структура РНМ

Тут RNN_1 — рекурентний шар, на вхід якого подається тривимірні вхідні тензори X_{train}^{3D} та $X_{validation}^{3D}$, в межах якого реалізуються однонаправлені, або двонаправлені [176] зв'язки між комірками. Вихідні значення цього шару подаються на вхід до повнозв'язного нейромережевого шару NN_2 перцептронів з метою повернення вектора фіксованого розміру, оскільки розмірності вихідних даних шару RNN_1 є $(batch, l, output)$, де *output* визначається довжиною вихідного вектора, а \tilde{y}^i — компоненти прогнозованого вектора \tilde{y} .

2.5.3.1. Довга короткочасна пам'ять LSTM

Оскільки проста РНМ має вагомий недолік, а саме, проблеми з “вибуханням” та “зниканням” градієнтів, зокрема при тренування на

довгих послідовностях, у роботі [177] була запропонована покращена архітектура комірки РНМ — довга короткочасна пам'ять, перевагами якої є наявність вектора пам'яті та здатність краще обробляти довгі послідовності даних. Крім того, РНМ використовує гіперболічний тангенс як функцію активації та пов'язує вектори стану \mathbf{h}^i , \mathbf{h}^{i+1} і y^i , а LSTM використовує для такого зв'язку вектор пам'яті.

На рис. 2.24 зображена схема комірки LSTM.

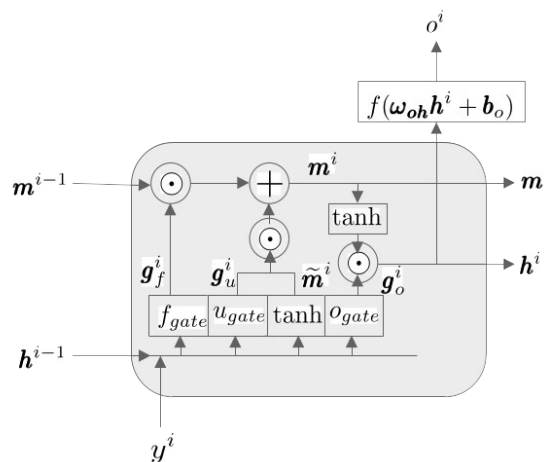


Рис. 2.24. Комірка довгої короткочасної пам'яті

Тут \mathbf{m}^i — вектор пам'яті комірки, f_{gate} , u_{gate} , o_{gate} — забувальний вентиль, вентиль оновлення стану і вихідний вентиль відповідно, g_f^i , g_u^i та g_o^i — вектори станів вентилів, $\tilde{\mathbf{m}}^i$ — проміжний вектор пам'яті, \odot — оператор добутку Адамара. Розмірності векторів пам'яті та стану у LSTM є однаковими та задаються в контексті гіперпараметрів нейромережі.

Основною відмінністю та водночас перевагою LSTM над простою РНМ є те, що LSTM мережі можуть вибірково запам'ятовувати, або забути частину інформації, що проходить через механізм пам'яті комірок. На кожному часовому кроці комірка отримує: а) інформацію, яка знаходиться у пам'яті з попереднього кроку; б) значення вектора стану з попереднього часового кроку та в) нові дані за поточний часовий крок.

Забувальний вентиль відповідає за видалення інформації з поточної комірки, беручи до уваги вектор стану попередньої комірки та застосовуючи сигмоїду, що повертає вектор зі значеннями у діапазоні $[0, 1]$. Вентиль оновлення стану комірки відповідальний за доповнення інформації та функціонує у декілька етапів: а) регулює, які значення слід додати до вектора пам'яті з використанням сигмоїда; б) створює вектор \tilde{m}^i , що містить всі можливі значення, якими можна доповнити вектор пам'яті з використанням функції гіперболічного тангенсу, яка нормує значення у діапазон $[-1, 1]$; в) множить регульовані значення на вектор \tilde{m}^i та додає нову інформацію до вектора пам'яті. Вихідний вентиль відповідає за виокремлення інформації з поточного стану комірки з використанням функцій сигмоїди і гіперболічного тангенса.

Алгоритм навчання LSTM детально описується у статті [178].

2.5.3.2. Вентильний рекурентний вузол GRU

Попри переваги *LSTM* над простою РНМ, складність архітектури її комірок можна зменшити без суттєвої втрати ефективності для простих задач, що було реалізовано у вентильному рекурентному вузлі GRU (рис. 2.25) [179]. На відміну від комірки LSTM, GRU не взаємодіє з вектором пам'яті, має більш спрощену архітектуру, в основі якої є вентиля скидання r_{gate} та оновлення u_{gate} .

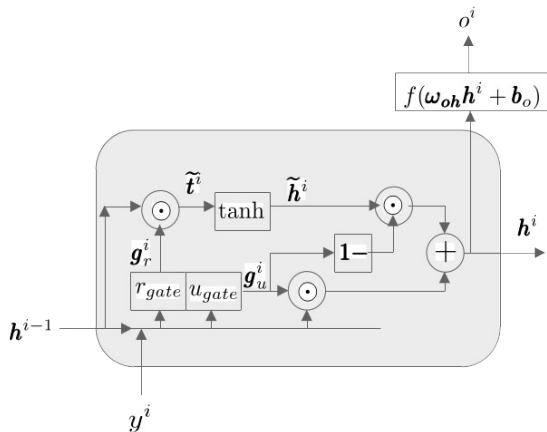


Рис. 2.25. Комірка вентильного рекурентного вузла

Застосування GRU нейромережі у створенні програмного забезпечення нейроконтролера має перевагу над LSTM, що проявляється у меншій обчислювальній складності водночас з співмірними показниками точності. GRU керує потоком інформації, як і блок LSTM, але без використання вектора пам'яті, що спрощує кількість обчислень та призводить до швидшого навчання нейромереж. Проте, для обробки достатньо довгих послідовностей, складність LSTM має перевагу над GRU, зокрема при обробці природної мови та генерації аудіо-послідовностей.

Незважаючи на зниження, або підвищення обчислюваної складності внаслідок використання *GRU*, або *LSTM* нейромереж на даний час не існує однозначної відповіді щодо переваженої ефективності однієї архітектури над іншою. В контексті проектування нейромереж для мікроконтролерних систем слід віддавати перевагу архітектурам з меншою кількістю арифметичних операцій та розміром файлу моделі, що буде розгортатися у пам'яті пристрою.

2.5.3.3. Гібридна згорткова рекурентна нейромережа (CNN-LSTM/GRU)

Окрім безпосередньо рекурентних нейромереж та їх видів (LSTM, GRU), для моделювання часових послідовностей та побудови прогностичних моделей можна використовувати гібридні архітектури. Одним з видів такої архітектури є згорткова рекурентна нейромережа CNN-LSTM/GRU, де в якості рекурентного шару може знаходитися шар LSTM, або GRU комірок. Під час роботи з одновимірними даними CNN-LSTM/GRU зчитує послідовність даних із певним лагом (затримкою) та вчиться виділяти ознаки, які мають значення для прогнозування.

Для таких операцій одношарова CNN може визначатися як набір n_κ одновимірних фільтрів κ довжиною l_κ , які зміщаються зі сталим кроком, що відповідає довжині зсуву по прогностичному вектору. Наприклад,

операція згортки між тензором $\mathbf{y} = [[y^1, y^2, \dots, y^\psi], 1]$ розмірності $\psi \times 1$ та фільтром κ описується як

$$\phi^{i,1} = \sum_{j=0}^{l_\kappa} \mathbf{y}^{i+j,1} \kappa^j, \quad (2.18)$$

де $\phi^{i,1}$ вихідний тензор після застосування одновимірної згортки, $i = 1, 2, \dots, \psi$.

Хоча у різних випадках, згортковий шар може знаходитися як до, так і після рекурентного, розглянемо тут таку архітектуру, де рекурентний шар слідує за згортковим шаром одновимірних фільтрів (рис. 2.26).

До вхідних даних застосовується набір одновимірних фільтрів, кількість яких n_κ та довжина l_κ задаються в якості гіперпараметрів нейромережі, які визначаються внаслідок оптимізації генетичним алгоритмом.

Вихідні тензори $\phi^{k,i,1}$ де $k \in [1, 2, \dots, n_\kappa]$ проходять крізь функцію активації, що забезпечує нелінійне перетворення у зрізаному лінійному вузлі $ReLU(\phi^{k,i,1}) = \max(0, \phi^{k,i,1})$ з подальшою обробкою у рекурентному та повнозв'язному шарах.

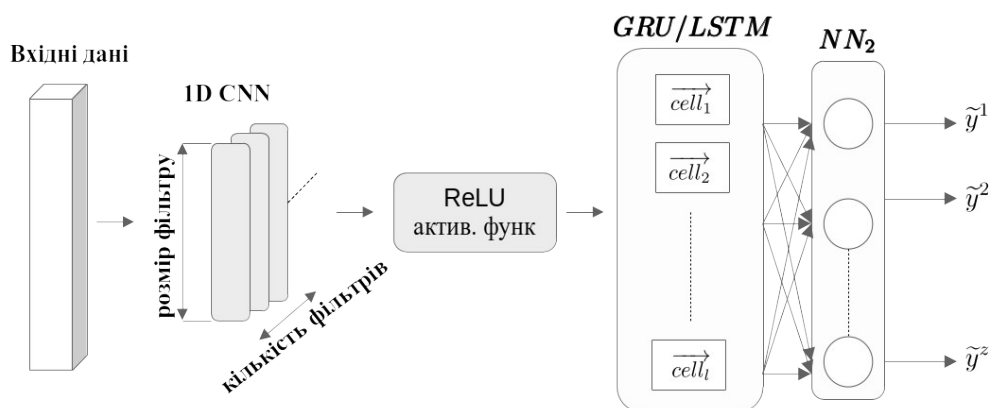


Рис 2.26. Структура гібридної згорткової рекурентної нейромережі

2.6. Оптимізація гіпер-параметрів

У методах глибокого навчання важливим етапом проектування та подальшого моделювання є якісний підбір гіперпараметрів, які визначають конфігурацію моделей та архітектури нейромереж. В контексті рекурентних нейромереж на основі LSTM, GRU чи CNN-LSTM/GRU до таких гіперпараметрів \mathbf{H} належать: а) довжина вектора прихованого стану h_l^i в i -му шарі; б) кількість рекурентних шарів i ; в) тип функції активації в i -му шарі; г) коефіцієнт швидкості навчання α ; г) розмір l_κ^i та кількість n_κ^i фільтрів для згорткового i -го шару РНН; д) розмір пакету даних для навчання $batch$; е) тип комірки рекурентного шару. Також, до цих гіперпараметрів можна долучити довжину вектора-предиктора v та кількість кроків для оптимального прогнозу p .

Серед відомих підходів, як пошук на сітці [180], випадковий пошук [181], Баєсова оптимізація [182], які широко використовуються у глибокому навчанні, еволюційні методи становлять окремий клас алгоритмів для оптимізації гіперпараметрів нейромереж. Представником такого класу методів є генетична оптимізація [175], схема якої показана на рис. 2.29.

На першому етапі відбувається генерація початкової популяції визначеного розміру випадкових розв'язків згідно правила неперервного рівномірного розподілу, які лежать у заданих межах: а) нижня і верхня межі довжини векторів h_l^i стану [50, 200] рекурентних комірок у i -тому шарі; б) нижня і верхня межі [3, 5] та [50, 150] для розмірів l_κ^i та кількості n_κ^i фільтрів для згорткового i -того шару у випадку CNN-LSTM/GRU відповідно; в) коефіцієнт темпу навчання $\alpha \in \{0.01, 0.05, 0.1\}$; г) розмір пакету $batch \in \{16, 32, 64\}$; г) довжина вектора-предиктора $v \in \{12, 15, 18\}$; д) кількість кроків для прогнозу $p \in \{2, 3, 5\}$; е) тип рекурентної комірки $cell \in \{LSTM, GRU\}$.

Для кодування тих гіперпараметрів, які набувають лише конкретних значень з множини обмежень (коефіцієнт темпу навчання, розмір пакету, довжина вектора-предиктора, кількість кроків для

прогнозу, тим рекурентної комірки) використовується цілочисельне представлення, наприклад $\alpha \in \{0.01, 0.05, 0.1\} \rightarrow [0, 2]$. Після обчислення функції (2.17), що передбачає процес перетворення закодованих розв'язків генетичного алгоритму у фактичні значення гіперпараметрів, навчання нейромережі для всієї популяції розв'язків та перевірки на досягнення максимальної кількості генерацій, відбувається сортування розв'язків у порядку зростання значення функції (2.17).

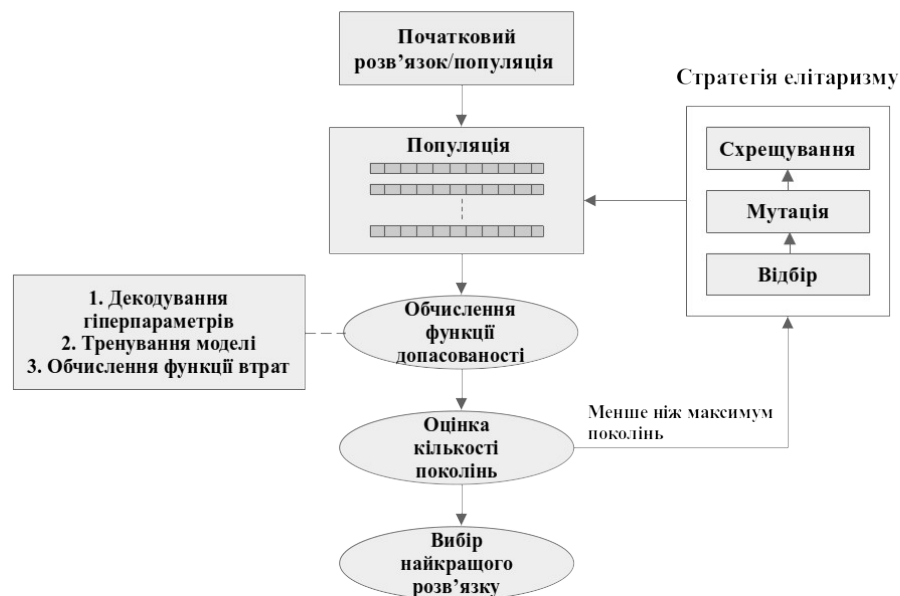


Рис. 2.27. Схема оптимізації гіперпараметрів з використанням ГО

На наступному кроці здійснюється відбір заздалегідь визначеної кількості розв'язків для побудови нової генерації шляхом турнірного відбору [183]. Незважаючи на те, що якість розв'язків у популяції зростає від покоління у покоління, існує ймовірність зникнення кращих індивідів у поточному поколінні, що пов'язано з змінами у розв'язок, що вносяться внаслідок схрещування і мутації. Для того, щоб найкращі індивіди обов'язково переходили у наступне покоління, необхідно застосувати стратегію елітаризму. Це означає, що n найкращих індивідів копіюються в наступне покоління, до того як всі місця будуть зайняті нащадками, отриманими в результаті відбору, схрещування та мутації. Скопійовані “елітні” розв'язки можуть використовуватися як батьки нових розв'язків.

Таким чином, задача обчислення гіперпараметрів для одношарової рекурентної нейромережі для нейроконтролера на базі ГО формулюється як:

- для заданого набору гіперпараметрів $\mathbf{H} = (h_l^1, \alpha, batch, v, p, cell)$ та обмежень $50 \leq h_l^1 \leq 150$, $\alpha \in \{0.01, 0.05, 0.1\}$, $batch \in \{16, 32, 64\}$, $v \in \{12, 15, 18, \}$, $p \in \{2, 3, 5\}$, $cell \in \{LSTM, GRU\}$ знайти оптимальний розв'язок \mathbf{H}^* такий, що

$$\mathbf{H}^* = \underset{\mathbf{H}}{\operatorname{argmin}} \frac{1}{N_v} \sum_{i=1}^{N_v} (\operatorname{model}(\mathbf{y}^i; \mathbf{H}) - y^i)^2. \quad (2.19)$$

Для обчислення параметрів гібридної *CNN-LSTM/GRU*, у функції (2.19) набір гіперпараметрів \mathbf{H} слід доповнити характеристиками фільтрів згорткового шару l_κ^1 та n_κ^1 .

2.7. Відбір нейромережі для нейроконтролера

Генетична оптимізація для обчислення гіперпараметрів нейромережі для прогностичного моделювання температурних даних ІБ дозволяє отримати оптимально сконфігуровану модель, придатну для розгортання на нейроконтролері з врахуванням обмежень на пам'ять пристрою (розмір нейромережі). Розглянемо процес відбору моделі для нейроконтролера, що складається з двох фаз: 1) обчислення гіперпараметрів та конфігурацію однієї моделі на базі LSTM/GRU рекурентних комірок для температурних даних з використанням (2.19); 2) здійснення аналогічного процесу (крок 1) для гібридної нейромережі при доповненні (2.19) додатковими наборами гіперпараметрів згорткового шару. Після отримання двох видів моделей на базі критеріїв точності, складності обчислень та розміру вихідного файлу відбувається відбір однієї моделі для розгортання на нейроконтролері.

Для чисельних експериментів були встановлені наступні параметри для налаштування ГА: розмір популяції — 12, ймовірність схрещування — 0.9, ймовірність мутації — 0.5, максимальна кількість поколінь — 4, кількість кращих індивідів-розв’язків у кожному поколінні — 3. ГО для задачі 2.39 реалізовувалась за допомогою бібліотеки еволюційних обчислень DEAP [184]. Обчислення, проведені для першої фази привели до результатів, які наведені у табл. 2.1, 2.2 та рис. 2.28.

h_l^1	α	$batch$	v	p	$cell$
163	0.01	64	12	3	<i>GRU</i>
Розмір		MSE		MAPE	
985.54 KB		0.05		0.0094	

Таблиця 2.1. Результати ГО для першої фази відбору

Тип шару	Розмірність вихідного тензора	Кількість параметрів
Вхідний шар	(•, 12, 1)	0
<i>GRU</i>	(•, 163)	81174
Повнозв’язний шар	(•, 3)	492
Сумарна кількість параметрів: 81666		
Кількість параметрів для визначення під час навчання: 81666		

Таблиця 2.2. Зведена характеристики моделі першої фази

h_l^1	α	$batch$	v	p	$cell$	l_κ^1	n_κ^1
188	0.01	16	15	2	<i>LSTM</i>	4	161
Розмір		MSE			MAPE		
3.06 MB		0.05			0.01		

Таблиця 2.3. Результати ГО для другої фази відбору

Тип шару	Розмірність вихідного тензора	Кількість параметрів
Вхідний шар	(•, 15, 1)	0
Згортковий шар	(•, 12, 161)	805
<i>LSTM</i>	(•, 188)	263200
Повнозв'язний шар	(•, 2)	378
Сумарна кількість параметрів: 264383		
Кількість параметрів для визначення під час навчання: 264383		

Таблиця 2.4. Зведена характеристики моделі другої фази

Як можна побачити з табл. 2.1, 2.2 оптимальним типом рекурентного шару було визначено GRU, використання якого для нейроконтролера є більш доцільним з точки зору обчислювальної складності у порівнянні з LSTM. Розмір отриманого файлу моделі не перевищує 1 МВ, що є сприятливим результатом для розгортання у пам'яті нейроконтролера навіть меншого об'єму внаслідок застосування процесу квантизації. Значення абсолютної середньо-відсоткової та середньоквадратичної помилок є достатньо малими. У табл. 2.3, 2.4 та рис. 2.29 показані результати підбору гіперпараметрів гібридної моделі, що на відміну від першої моделі містить більшу кількість гіперпараметрів.

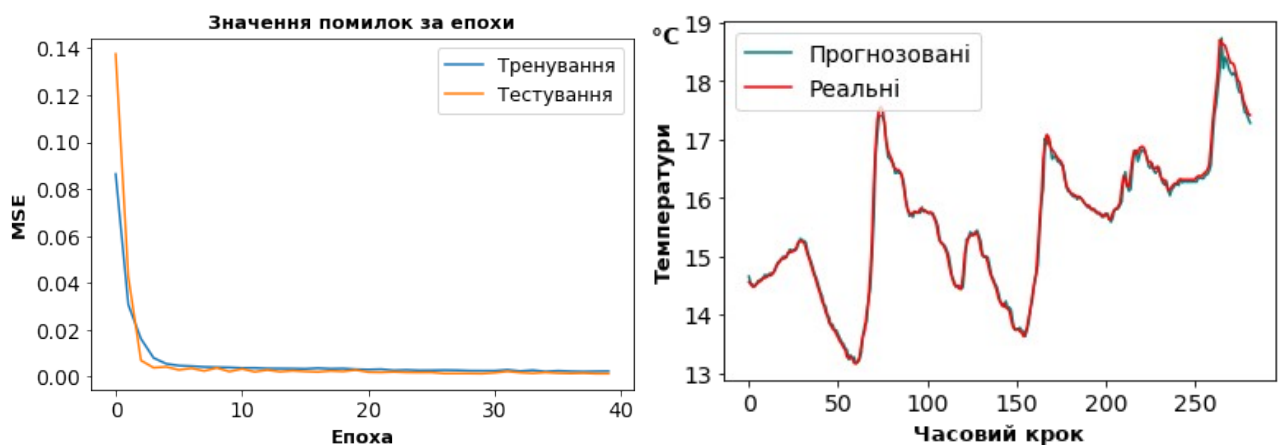


Рис. 2.28. Результати для моделі першої фази

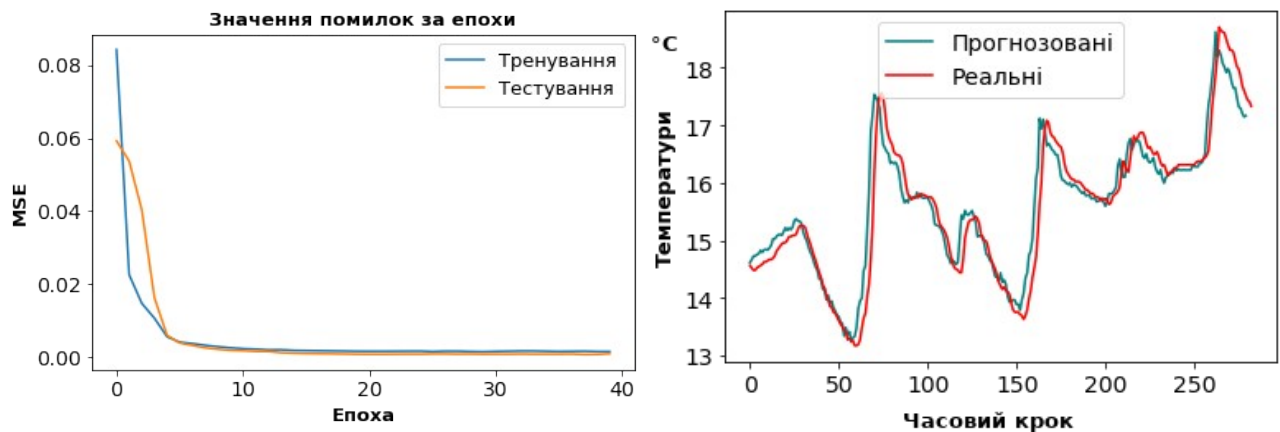


Рис. 2.29. Результати для моделі другої фази

Незважаючи на те, що модель другої фази досліджень є порівняно більшою — вихідний файл розміром 1.8 МВ, її точність у задачі прогнозування температур є практично такою ж, як і у моделі першої фази. Для нейроконтролера з об'ємом пам'яті менше 1 МВ (STM32 F407) розгортання такої моделі є неможливим без використання квантування ваг. Проте для пристрою з об'ємом пам'яті 2 МВ (STM32 H745) модель може розміщатися без попереднього квантування. Для використання моделі менш потужним пристроєм, наприклад, STM32 F407 доцільно обмежитися моделлю першої фази, а для більш потужного — модель другої фази може бути більш гнучкою та оптимальною з міркувань розширення і ускладнення задачі.

Висновки до розділу 2

У розділі побудовані та розглянуті статистичні і нейромережеві моделі для аналізу та обробки сенсорних даних температур та споживання газу в інтелектуальному будинку. Описаний процес підготовки та попередньої обробки даних (виявлення викидів та аномальних показників) з використанням методу ізольованого лісу та *STL*-декомпозиції. Розроблені регресійні моделі для:

а) температурних даних упродовж літнього періоду за відсутності обігріву приміщень, на основі чого можна 1) наближено оцінити теплозберігаючі властивості кімнати/будинку шляхом аналізу кута нахилу регресійної прямої та величини коефіцієнту детермінації $R^2 - adjusted$, що неявно вказують на здатність будинку підтримувати температурний режим під час втрачання тепла через зовнішні стіни через різницю температур; 2) здійснювати прогнозування внутрішніх температур за даними зовнішніх температур;

б) даних споживання газу і температур на батареях упродовж зимового періоду (лінійна модель з представленням незалежної змінної споживання газу у вигляді полінома другого степеня та модель на базі градієнтного підсилення), здійснено їх порівняння та виявлена перевага останньої, зокрема без обмежень, які типово накладаються на лінійну модель; така модель дозволяє пов'язати споживання газу та температури на батареях та у подальшому використовуватися для визначення ефективності обігріву;

в) даних зовнішніх та внутрішніх температур, а також споживання газу упродовж зими, що допомагає використовувати її для прогнозування внутрішніх температур за заданим споживанням газу і зовнішніми температурами; оскільки між даними немає чітко вираженої лінійної залежності, було використано метод на базі градієнтного підсилення, що повертає оптимальне регресійне дерево рішення.

Розроблені прогностичні регресійні моделі становлять частину створеного прототипу програмного комплексу для інтелектуального будинку та можуть використовуватися як для неявної оцінки теплозберігаючих характеристик інтелектуального будинку, так і для статистичного аналізу теплової поведінки інтелектуального будинку. Наприклад, значення коефіцієнтів детермінації, кутів нахилу регресійних прямих неявним чином свідчать як про ефективність опалення чи властивості інтелектуального будинку до збереження температур, так і можуть доповнювати більш складні моделі для оптимізації його функціонування.

Запропонований простий кластерний алгоритм для дезагрегації даних споживання газу, що ґрунтується на ознаках кривої добового профілю споживання. На основі відповідних результатів можна визначати споживачів газу. Наведена аналітично-числова модель споживача газу інтелектуального будинку. Для покращення запропонованих результатів можна доповнювати виділені ознаки (наприклад, температурними показниками) та впроваджувати більш складні моделі машинного навчання.

Розроблено рекурентні нейромережеві моделі для задачі прогнозування даних температури у вигляді одновимірного часового ряду на базі LSTM, GRU та гібридної CNN-LSTM/GRU архітектур. Метою розробки цих моделей є їх подальше розгортання на нейроконтролерній системі на базі мікроконтролерів STM32 F/H серії для побудови апаратно-програмного комплексу оптимізації функціонування інтелектуального будинку. Результати такого прогнозування можна використовувати для встановлення часу вмикання/вимикання опалення для оптимізації енергоспоживання, або для виявлення аномальних надходжень з сенсорів шляхом порівняння прогнозованих значень з реальними. Також на базі рекурентних нейромереж можливим є застосування автокодувальників (encoder-decoder) для зниження розмірності вхідних даних та збереження їх в кодованому представленні.

Запропоновано та застосовано підхід до оптимізації гіперпараметрів цих моделей на основі генетичного алгоритму, реалізацію якого можна впроваджувати для обчислювального хабу на базі мікрокомп'ютера Nvidia Jetson Nano. Оскільки розроблення моделей передбачає їх використання у програмному забезпеченні нейроконтролерної системи, під час розробки були накладені обмеження на складність і розмір вихідного файлу моделі; проаналізована їх ефективність; детально описаний процес підготовки та обробки температурних даних для рекурентних нейромереж.

РОЗДІЛ 3. ПРОТОТИП НЕЙРОКОНТРОЛЕРНОЇ СИСТЕМИ

У третьому розділі дисертаційної роботи запропоновано прототип апаратно-програмного комплексу нейроконтролерної системи для ІБ та деталізовано кроки її реалізації. Метою розробки такої системи є

1. дослідження, проектування та розгортання нейромережових моделей для обробки сенсорних температурних даних ІБ з використанням програмного каркасу TensorFlow 2, який має широке застосування у граничних обчисленнях;
2. створення *end-to-end* апаратної конфігурації мікрокомп'ютера Nvidia Jetson Nano з мікроконтролером STM32 H745/STM32 F767 як бази для програмного комплексу опрацювання сенсорних даних ІБ;
3. розробка та імплементація підходу MLOPS до побудови IoT рішень в контекст граничних обчислень;
4. дослідження процесу розгортання та параметрів нейромережових моделей на нейроконтролері;
5. доповнення існуючих підходів автоматизації ІБ.

3.1. Концепція прототипу нейроконтролерної системи

Першочерговою задачею запропонованої нейроконтролерної системи (мікроконтролер + нейронна мережа) є розвантаження центрального хабу та перенесення частини обчислювальних ресурсів (використання нейромережі для первинної обробки вхідних сенсорних даних) на мікроконтролерну платформу STM32 H745/F767. Для дослідження і тестування такої системи в контексті ГО на нейроконтролері були обрані нейромережові прогностичні моделі на базі рекурентних та гібридно-рекурентних архітектур. Розроблені моделі використовуються для здійснення короткотермінового прогнозування температурних даних,

ґрунтуючись на динаміці попередніх вимірювань. Оскільки ціллю роботи є проектування прототипу нейроконтролерної системи, для її дослідження була поставлена більш проста задача прогнозування одновимірних температурних рядів, що позаяк може бути розширена та доповнена складнішими випадками (багатовимірними часовими рядами з вимірними вологістю, освітлюваністю тощо).

Для реалізації прототипу нейроконтролерної системи необхідно забезпечити цілісність й інтеграцію усіх компонент та передбачити коректний потік даних в межах як апаратної, так і програмної структур.

Базова архітектура системи спроектованого апаратно-програмного комплексу (рис. 3.1) складається з таких вузлів:

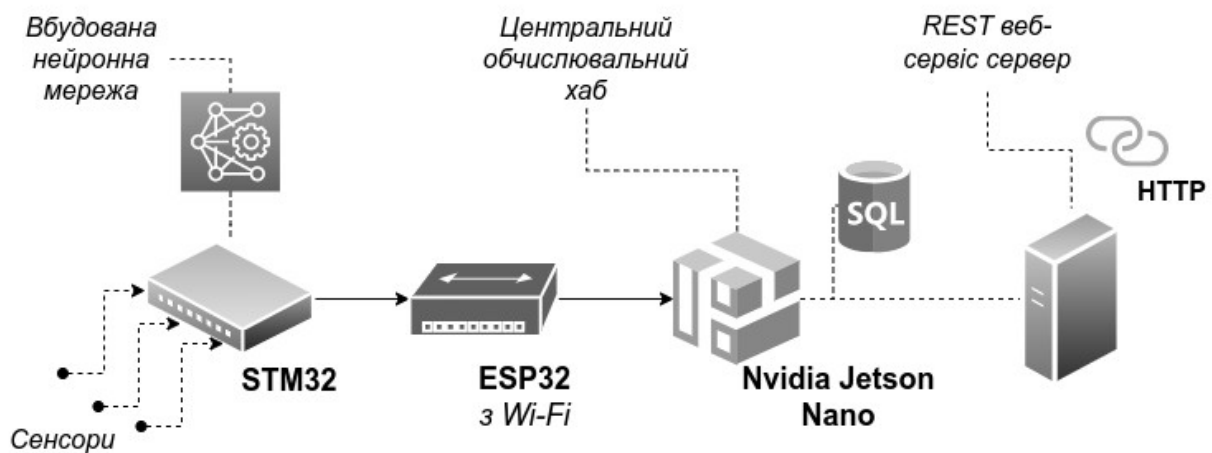


Рис. 3.1. Архітектура апаратно-програмного комплексу нейроконтролера

- набір сенсорів ІБ, наприклад давачів температури DS18B20, розташованих в приміщенні та на зовні (опційно);
- мікроконтролера сімейства STM32 з підтримкою пакету розширення X-CUBE-AI та розгорнутою на ньому нейромережею для короткотермінового прогнозування температури;
- мережевого з'єднання нейроконтролера з центральним хабом на базі Nvidia Jetson Nano з використанням TCP Sockets та Wi-Fi мережі за підтримки мікроконтролера ESP32;

- центральний хаб на основі мікрокомп'ютера Nvidia Jetson Nano під операційною системою Linux4Tegra, базованої на Ubuntu 20.04, який виступає основним обчислювальним вузлом для тренування нейронних мереж, розроблених аналітичних програмних модулів та розгорнутої бази даних SQLite;
- розгорнутим REST веб-сервером на хабі, доступ до якого здійснюється за HTTP-протоколом.

Концептуально, робота такої системи складається з таких етапів: а) температурні дані, які отримуються сенсорами ІБ надходять на нейроконтролер з розгорнутою нейронною мережею; б) нейроконтролер надсилає запит до хабу, отримує історичні дані температур та на їх основі здійснює короткотермінове прогнозування температури на період, визначений конфігурацією нейромережі; в) нейроконтролер порівнює прогнозовані дані з отриманими та може: 1) виявляти аномальну в контексті натренованої моделі поведінку отриманих температур; 2) доповнювати отримані дані прогнозованими значеннями за наявності втрачених чи пошкоджених вимірів; в) отримані від нейроконтролера значення зберігаються у базу даних на хабі та можуть використовуватися у подальших обчисленнях, або в якості бази для реалізації оптимального функціонування ІБ (сценарії оптимізації часу увімкнення/вимкнення опалення тощо).

3.2. Постановка задачі побудови нейроконтролерної системи

У роботі для нейромережевого забезпечення мікроконтролерів STM32H745 та STM32F767 та перетворення їх у нейроконтролер необхідно розв'язати такі задачі:

1) розробити нейромережеві моделі для обробки і прогнозування температури ІБ, здійснити їх аналіз та порівняння з метою побудови

оптимальної прогностичної моделі під супроводом програмного каркасу TensorFlow 2 (розділ 2);

2) здійснити оптимізацію гіперпараметрів моделей та їх порівняння з метою визначення найбільш ефективної архітектури з точки зору розгортання на мікроконтролерах (розділ 2);

3) спроектувати та реалізувати робочий цикл отримання та обробки даних, навчання моделей і їх розгортання на нейроконтролері з врахуванням практики MLOPS та сучасних засобів TensorFlow 2, Mlflow, Apache Airflow;

4) створити клієнт-серверну обгортку для робочого циклу п. 3 на базі FastAPI для забезпечення REST підходу до апаратно-програмного комплексу та моніторингу стану моделі.

Для пришвидшення реалізації нейроконтролерного комплексу підготовка, оптимізації архітектур та дослідження прототипів нейронних мереж здійснювалась на базі окремого сервера з дискретною відеоплатою Nvidia RTX 3080 Ti. Крім того, процес тренування та оптимізації гіперпараметрів нейромереж відтворювався на мікрокомп'ютері Nvidia Jetson Nano з під'єднаним нейроконтролером на базі STM32H745/STM32F767 для оцінки його ефективності як складової розробленої системи.

3.3. Неперервне розгортання моделей у системі нейроконтролера

В процесі надходження нових температурних даних до центрального хабу через нейроконтролер виникає потреба оновлення розгорнутої нейронної мережі шляхом її перетренування, або підбору нової топології. Зі зміною характеру часових залежностей (переходу на новий сезон року, встановлення нових систем опалення, вентиляції, утеплення об'єктів тощо) модель, підготовлена на базі попередніх даних втрачає точність та ефективність прогнозування. В цілому, для вирішення цієї проблеми застосовують такі підходи [185]: 1) онлайн-

навчання (модель дотреновується з надходженням кожного нового зразка даних); 2) офлайн-навчання (модель перетреновується на новому навчальному наборі даних, що складається зі старих й отриманих нових зразків) та 3) пакетне навчання (модель дотреновується на деякому наборі нових даних визначеного розміру $n \geq 1$).

В контексті даного комплексу можна використовувати підхід офлайн-навчання, за якого після отримання прогнозованих $\hat{\mathbf{y}}_k$ та реальних \mathbf{y}_k температур, де k відповідає номеру деякого типу температури (внутрішньої, або зовнішньої), а \mathbf{y}_k та $\hat{\mathbf{y}}_k$ — часові температурні ряди, протягом заздалегідь визначеного часового інтервалу I_f здійснюється перетренування нейромережі на новій навчальній вибірці $\mathbf{Y} = \cup_k (\mathbf{Y}_k^{t-1} \cup \mathbf{y}_k^t)$, де \mathbf{Y}_k^{t-1} навчальний набір рядів до поточного моменту прогнозування $t - 1$, а \mathbf{y}_k^t — отриманий ряд сенсорних даних від моменту прогнозування t довжиною I_f . Такий процес можна проводити циклічно доки результати прогнозування моделі є прийнятними у відповідності до критерію точності, визначеного за деякою нормою

$$L = \|\mathbf{y}_k - \hat{\mathbf{y}}_k\|. \quad (3.1)$$

У більшості випадків метрики для порівняння часових рядів (параметрів $\hat{\mathbf{y}}_k$ та \mathbf{y}_k) (3.1) визначаються як середнє абсолютне відхилення (3.2) та середньоквадратичні відхилення (3.3) та (3.4):

$$L = \frac{\sum_{i=1}^n |y_k^i - \hat{y}_k^i|}{n}, \quad (3.2)$$

$$L = \frac{\sum_{i=1}^n (y_k^i - \hat{y}_k^i)^2}{n}, \quad (3.3)$$

$$L = \sqrt{\frac{\sum_{i=1}^n (y_k^i - \hat{y}_k^i)^2}{n}}. \quad (3.4)$$

Тут y_k^i та \hat{y}_k^i компоненти векторів \mathbf{y}_k та $\hat{\mathbf{y}}_k$ відповідно, а n — довжина часових рядів. Отримане значення метрики за (3.2)-(3.4) можна

порівнювати з деякою допустимою пороговою величиною τ , визначеною на основі статистичного аналізу таких часових рядів.

На рис. 3.2 наведена схема процесу перерахунку гіперпараметрів H моделі задля покращення прогнозування, коли модель демонструє втрату точності згідно метрики (3.1).

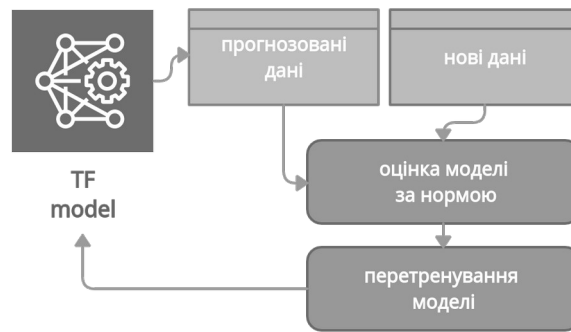


Рис. 3.2. Процес перетренування моделі

Модель реалізує прогнозування часового ряду \hat{y}_k на базі історичних даних з Y , які порівнюються з новими сенсорними параметрами y_k . Після цього, відбувається перетренування моделі з врахуванням нових даних y_k . Безпосередньо, тренування здійснюється на хабі Nvidia Jetson Nano з подальшим автономним оновленням файлів моделі на мікроконтролері.

3.4. Апаратна архітектура нейроконтролерної системи

Мікрокомп'ютер Nvidia Jetson Nano (рис. 3.3), на базі якого реалізовано хаб має такі обчислювальні характеристики: графічний прискорювач 128 ядер NVIDIA Maxwell™, центральний процесор quad-core ARM® A57 1479 MHz та 4 GB 64-bit LPDDR4 оперативної пам'яті зі швидкістю читання 25.6 гігабайт/секунду. Це забезпечує достатні обчислювальні ресурси як для запуску веб-серверу, так і для тренування нескладних (одношарових) рекурентних нейромереж завдяки наявності 128 ядерного графічного модуля.

З'єднання Nvidia Jetson Nano з нейроконтролером реалізується через мікроконтролер ESP32-WROOM-32 (Xtensa LX6 240 МГц, 4 МВ RAM), який обладнаний Wi-Fi модулем та виступає проміжним вузлом для підтримки операцій передавання файлу моделі нейромережі та отримання даних з нейроконтролера.

Розгортання та оновлення моделі на нейроконтролері здійснюється наступним чином: а) спочатку відбувається ініціалізація та “прошивка” нейроконтролера підготовленою моделлю (з заданою топологією нейромережі) на хабі через USB порт; б) оновлення моделі після її перетренування відбувається шляхом заміни вагових коефіцієнтів і не передбачає зміни її топології через неможливість заміни мікрокоду на нейроконтролері.

В свою чергу, з'єднання мікроконтролера ESP32 з нейроконтролером здійснено за протоколом UART.

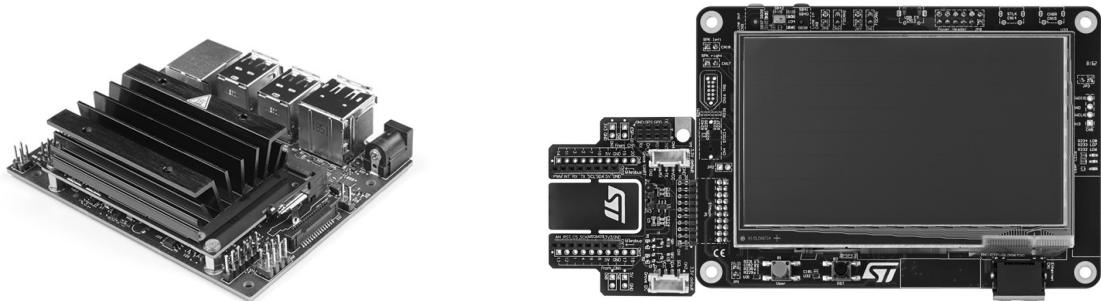


Рис. 3.3. Nvidia Jetson Nano та плата з мікроконтролером STM32 H745

Для використання запропонованого прототипу в якості системи реального часу ІБ сенсори температури, наприклад DS18B20, під'єднуються до нейроконтролера через інтерфейс GPIO.

Завдяки модульності, запропонований прототип може бути легко розширений додатковим мікроконтролерним устаткуванням. Крім цього, доступ до глобальної мережі Інтернет дає змогу залучати мобільні сервіси

для керування роботою даного комплексу, а саме внаслідок створення мобільних додатків та інтеграції з іншими віддаленими IoT системами.

Окремим питанням даної розробки є забезпечення кіберзахисту: каналів зв'язку, доступу до приватної інформації тощо. Впровадження останнього потребує окремого дослідження, проте не виноситься в обсяг даної дисертаційної роботи.

3.5. Програмна архітектура нейроконтролерної системи

Програмна реалізація застосунку керування нейроконтролерної системи ґрунтується на двох-компонентній архітектурі: а) серверної частини для обробки запитів та моніторингу стану розгорнутої нейромережевої моделі та б) взаємодії клієнта-мікроконтролера STM32 H/F з запущеним сервером за допомогою технології TCP сокетів для передавання даних та оновлення ваг моделі (рис. 3.4). Основною мовою програмування даної системи є Python версії 3.8, оскільки саме ця версія підтримується варіантом операційної системою Ubuntu 20.04 до Nvidia Jetson Nano.

Серверна частина системи створена за допомогою мікропрограмного каркасу FastAPI, який підтримує швидкий асинхронний веб-сервер Uvicorn. Застосунок на базі FastAPI пов'язує клієнтські запити, що можуть надходити на визначені маршрутні функції з відповідними обробниками подій, які визначаються логікою процесу функціонування системи (запуск тренування моделі, здійснення прогнозування та розгортання). З використанням TCP сокетів виконується надсилання запитів з боку як серверної, так і клієнтської (мікроконтролер STM32+ESP32) частин в межах локальної мережі Ethernet.

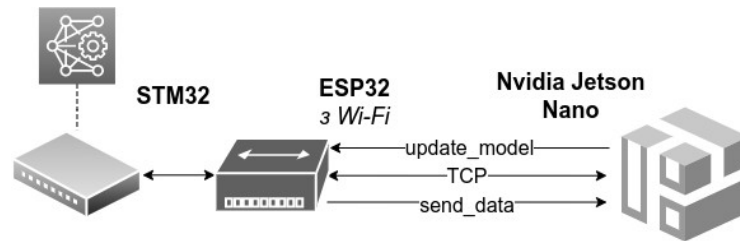


Рис. 3.4. Клієнт-серверна схема

При виявленні дрейфу даних, або втрати заданої точності моделі, сервер може оновлювати розгорнуту на мікроконтролері нейромережеву модель (*update_model*). У свою чергу, після здійснення обробки даних температури, клієнт-мікроконтролер надсилає ці дані до серверу (*send_data*) для їх подальшої обробки і збереження у базу даних. Окрім клієнта-мікроконтролера така архітектура дозволяє розширювати користувачів серверу в контексті долучення інших пристроїв (надсилання сповіщень про стан системи чи даних на смартфон, додавання інших сенсорів та обробників тощо) і масштабувати запропоновану систему у бік її вдосконалення.

3.5.1. Структура серверної частини

Для розроблення серверної частини нейроконтролерної системи перевагу було надано мікропрограмного каркасу FastAPI. Вибір оптимального каркасу серед таких популярних рішень веб-серверного дизайну як Django, Flask, FastAPI, Pyramid був здійснений з таких міркувань:

- розгортання серверної частини на пристрої з обмеженими обчислювальними ресурсами;
- наявність якісної та вичерпної документації;
- підтримка асинхронного програмування, ASGI та концепції RESTful.

На поточний момент тестування швидкодії цих програмних каркасів демонструє перевагу FastAPI над конкурентами [186], що є вагомим аргументом з точки зору граничних обчислень. Також, оскільки задачі машинного навчання у контексті граничних обчислень можуть потребувати значного обчислювального часу, необхідно забезпечити можливість асинхронної обробки інших запитів, що надходять на сервер. Це повністю реалізовується базовим каркасом FastAPI, що дозволяє обробляти як синхронні, так і асинхронні запити і має вбудовану підтримку перевірки даних, серіалізації JSON, аутентифікації та авторизації, а також підтримку OpenAPI.

Основним архітектурним шаблоном проектування серверної частини виступає Модель-Вид-Шаблон (MVT) (рис 3.5).

На рис. 3.6. показана концептуальна MVT-діаграма застосунку. Модель (MODEL) — це абстракція, яка визначає об'єктне представлення даних, що зберігаються у базі даних SQLite. Такими даними є температурні показники, отримані внаслідок вимірювань. З метою створення швидкого та робочого прототипу нейроконтролерної системи модель даних була обмежена до однієї таблиці температур (Temperature), проте за потребою може бути легко доповнена об'єктами-представленнями, наприклад, класами користувача й іншими кліматичними показниками.

Вид (VIEW) — це реалізація взаємодії користувацьких, або внутрішніх запитів з базою даних, що передбачає виконання CRUD (створити/прочитати/оновити/видалити) дій з даними температур та містить виклики основних обчислювальних операцій з потокової схеми рис. 3.7.

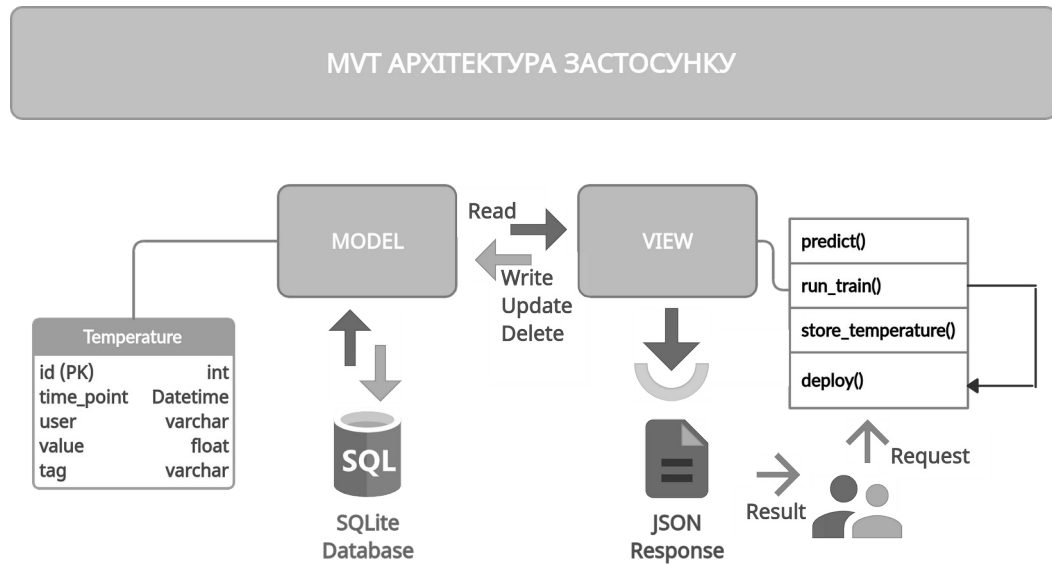


Рис. 3.5. MVT архітектура для FastAPI

Шаблон (TEMPLATE) не був реалізований у даній роботі, але може бути легко доданий для візуального представлення функціоналу прототипу засобами HTML/CSS/javascript.

Основні маршрутні функції, визначені у цьому блоці відповідають за такі операції:

- *predict()* — здійснює прогнозування температури розгорнутою на сервері моделлю на задану кількість кроків, що визначається конфігурацією моделі;
- *run_train()* — запускає процес, визначений DAG-схемою, який передбачає отримання даних, тренування нейромережевої моделі з/без автоматичного підбору гіперпараметрів шляхом застосування генетичного алгоритму та викликає метод *deploy()*;
- *store_temperature()* — агрегує роботу з базою даних через методи CRUD;

- *deploy()* — реалізує вбудовування підготовленої нейромережевої моделі у нейроконтролер через виклик функції *update_model* на базі TCP сокетів.

3.5.2. Програмні компоненти та конвеєр розгортанням

Для здійснення синхронізованого трикрокового процесу обробки даних, навчання моделі та її розгортання на сервері використано підходи сучасної концепції MLOPS, метою якої є гнучке впровадження експериментальних моделей машинного навчання у виробничу систему. Потреба реалізації такого процесу для нейроконтролерної системи спричинюється постійним накопиченням та зміною температурних даних, від яких залежать натренована модель, що розгортається. Відтак нейромережеву модель необхідно перетренувати та заново адаптовувати до нових даних під час роботи системи без її зупинки.

На рис. 3.6 наведений цикл робочого процесу розгортання, який складається з логічних модулів, визначених у коді. Перший компонент обробки і підготовки даних містить такі модулі:

- *stat_analysis* — містить реалізацію статистичної обробки температурних даних (регресійні моделі та попередній аналіз), а також додаткові можливості розширеного статистичного аналізу даних температур та споживання енергії обігріву (розділ 2);
- *phys_formulation* — код тепло-фізичного моделювання та розв'язання оберненої задачі визначення ефективних тепло-фізичних параметрів на прикладі даних REFIT (розділ 4);
- *collect_data* — реалізує збереження температурних вимірів у SQLite базі даних;

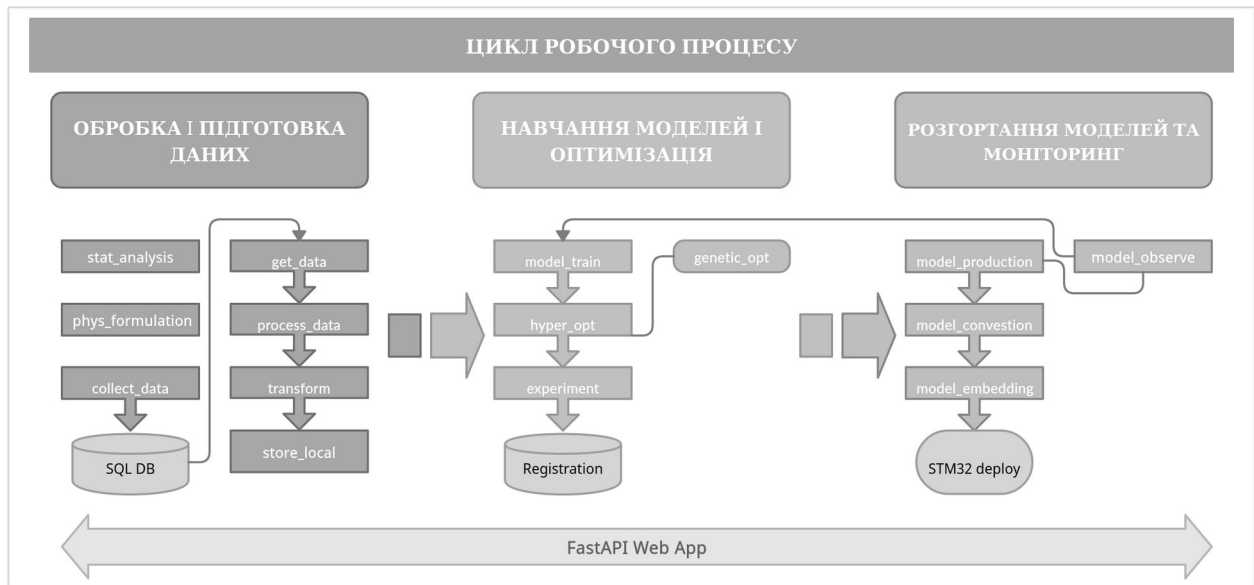


Рис. 3.6. Поточкова схема

- *get_data* — містить функції для отримання даних з SQLite, збереження проміжних оброблених масивів з температурними даними у локальній директорії проекту, які використовуються для тренування нейромережових моделей;
- *process_data* — визначає клас-обробник для нормування, зведення до тривимірних тензорів (*transform*) й розбиття масиву даних температури на навчальну і тестувальну вибірки;
- *store_local* — зберігає проміжні тривимірні масиви температури (навчальні і тестувальні вибірки) у локальній директорії, що визначається у файлі конфігурації проекту.

Конфігурація проекту та налаштування локального середовища задаються у відповідному yaml-файлі `params.yaml`.

Другий компонент поточної схеми 3.7 — навчання та оптимізація моделей складається з наступних модулів:

- *model_train* — містить функції для старту тренування нейромережових моделей і оцінки точності, задання моделей з

використанням програмного каркасу TensorFlow 2 та запуск платформи Mlflow для реєстрації (Registration), збереження у SQLite (за замовчуванням) та відстежування версій та параметрів моделей;

- *hyper_opt* — здійснює оптимізацію гіперпараметрів нейромережових моделей, містить створений клас *GenOneLayerSequenceModel* для обчислення гіперпараметрів одношарових рекурентних та гібридних нейронних мереж на основі генетичного алгоритму;
- *experiment* — допоміжний блок, що складається з Jupyter Notebook файлів для здійснення чисельних експериментів та дослідження моделей в інтерактивному режимі.

Третій блок — розгортання моделей та моніторинг відповідає за безпосередній запуск натренованих і сконфігурованих моделей на рівні сервера, готових для використання за користувацькими запитами. Також, може здійснювати відстежування стану моделей та отриманих нових даних, на основі чого може відбуватися перетренування з подальшим повторним розгортанням. Містить такі модулі:

- *model_production* — використовує зареєстровану і збережену засобами Mlflow модель згідно запитів до серверу;
- *model_conversion* — конвертація моделі з формату HDF5 (h5) у C-масив ваг та її збереження;
- *model_embedding* — вбудовування моделі на мікроконтролер STM32;
- *model_observe* — додатковий (опційний) модуль, що здійснює моніторинг моделі, наприклад, за допомогою інструменту Evidently AI, відстежує нові дані на наявність проблеми дрейфу даних та оцінює точність прогнозування моделі.

Для послідовного запуску кожного з компонентів системи використовується платформа Apache Airflow для управління потоками

робіт [187], які задаються орієнтованими ациклічними графами (DAG). Це дає змогу відстежувати виконання кожного завдання: обробки та підготовки даних, тренування нейромережових моделей з наступною реєстрацією у Mlflow та отримання найкращої моделі для її розгортання як на локальному сервері, так і у системі нейроконтролера.

Для досягнення послідовності процесу тренування нейромережових моделей створений відповідний DAG файл, що запускає компоненти та відповідні їм модулі з потокової схеми 3.6.

На рис. 3.7 показана схема ациклічного графа для тренування моделей та їх розгортання на нейроконтролері. Кожне з завдань визначає вузли Airflow графа *Ingest*, *Train_tune*, *Extract_transform*, *Train* та *Deploy*, які відповідають за запуски складових програмних модулів. Виконання наступного завдання не може бути запущене, поки не завершиться попереднє. Таким чином, Airflow дбає про цілісність та синхронізацію робочого потоку.

Компонент *Ingest* на основі заданої конфігурації у файлі `params.yaml` завантажує дані температури за деякий період з бази даних SQLite у `pandas` [112] — таблицьку та зберігає її у форматі `.CSV`.

Далі, можливі два варіанти виклику послідовності компонент на підставі параметру `pipeline_option` у `config.yaml`.

Перший з них — запуск повного процесу навчання нейромережі та підбору її гіперпараметрів *Train_tune* з використанням генетичного алгоритму, під час чого здійснюється генерація тренувальної та тестувальної вибірок (*extract_transform*) зі збереженого CSV файлу (крок *Ingest*). Отримана внаслідок таких кроків модель реєструється в окремо запущеному сервісі Mlflow, а її параметри також вписуються в `config.yaml` у поле `model_config` (рис. 3.8).

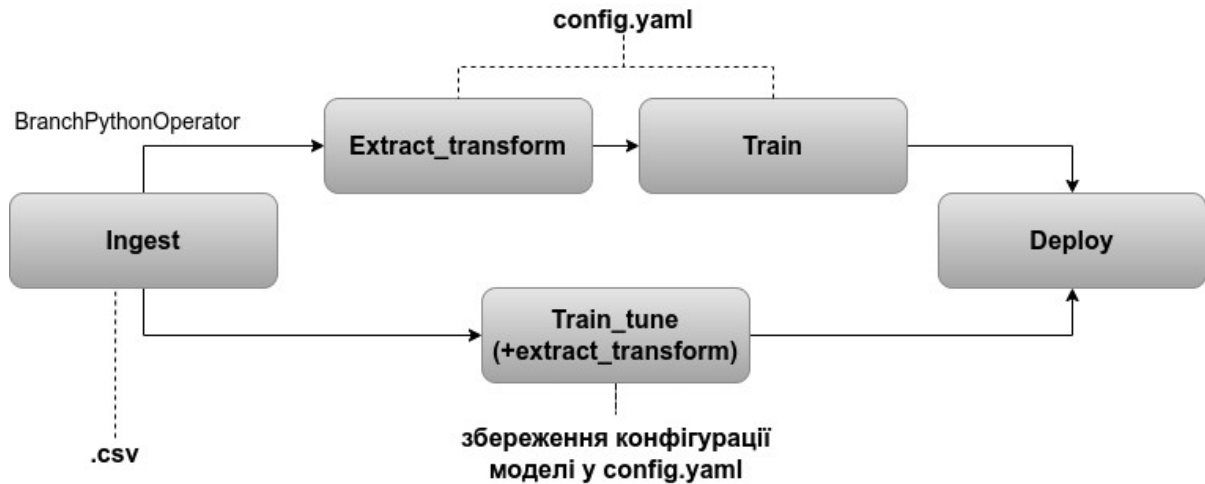


Рис. 3.7. DAG-схема тренування та розгортання моделі

Другий варіант виклику — це виконання двох кроків *Extract_transform* та *Train*, де перший генерує навчальну та тестувальну вибірки на базі конфігурації у полі *model_config*, а другий — здійснює навчання моделі та її реєстрацію через Mlflow.

Останній компонент схеми *Deploy* реалізує завантаження збереженої засобами Mlflow моделі у нейроконтролер через TCP-сокети.

```

temperature.py × ingest.py × train_tune.py × ext_transform.py ×
1  model_config:
2      activation: tanh
3      batch_size: 64
4      cell: GRU
5      hidden_vector: 142
6      learning_rate: 0.05
7      n_in: 48
8      n_out: 1
9      pipeline_option:
10     un_full: true
11

```

Рис. 3.8. Приклад вмісту config.yaml

Задля досягнення ізольованості та стабільності програмної частини прототипу можна використовувати Docker-контейнери, проте для розгортання системи на мікрокомп'ютері таке завдання є більш обчислювально складним.

3.6. Процес та результати роботи прототипу

Для демонстрації роботи нейроконтролерної системи спочатку розглянемо механізми та послідовність запусків потокової схеми 3.6, а після чого інтегровану роботу усього комплексу.

Обробка і підготовка даних. Для тестування розробленого прототипу апаратно-програмного комплексу були використані температурні дані, виміряні у лабораторії інтелектуальних автономних систем факультету електроніки та комп'ютерних технологій Львівського національного університету імені Івана Франка за період з 1 лютого 2021 року по 31 вересня 2021 року. Спершу, розглянемо роботу потокової схеми рис. 3.6. Для отримання температурних даних за деякий визначений період з *SQLite* виконується запит з параметром виду

$$\begin{aligned} \{start_date : "2021 - 02 - 02T14 : 51 : 01", \\ end_date : "2021 - 03 - 02T14 : 51 : 01"\}, \end{aligned} \quad (3.5)$$

який обробляються засобами *SQLAlchemy* (при зовнішньому HTTP доступі через маршрутну функції), або безпосередньо *sql*-запитом.

Отримані дані зберігаються у проміжному CSV файлі (рис. 3.9) у піддиректорії проекту *data/raw/*, яка перебуває в кореневому каталозі проекту (рис. 3.10).

```
stm_neural > data > raw > train.csv
1  ,time_point,user,value,tag,id
2  0,2021-02-02T14:51:01.026563,home,11.937,indoor,20
3  1,2021-02-02T15:01:02.932900,home,12.0,indoor,21
4  2,2021-02-02T15:11:04.852834,home,12.125,indoor,22
5  3,2021-02-02T15:21:06.772924,home,12.187,indoor,23
6  4,2021-02-02T15:31:08.532852,home,12.125,indoor,24
7  5,2021-02-02T15:41:10.372329,home,12.125,indoor,25
8  6,2021-02-02T15:51:12.212860,home,12.125,indoor,26
9  7,2021-02-02T16:01:13.972915,home,12.125,indoor,27
10 8,2021-02-02T16:11:15.892863,home,12.062,indoor,28
```

Рис. 3.9. Приклад даних у CSV-файлі

Для опрацювання цих даних використовуються обробники з модулів *extract_transform*, що відповідають за нормування і розбиття на навчальну і тестувальну вибірки та трансформацію у тривимірні тензори — вхідні структури даних для розроблених нейромереж. Під час обробки даних об'єкт, який здійснює нормування даних зберігається у бінарному форматі з розширенням *scaler.pkl* у директорії *data/processed/*. Його задача — здійснювати пряме та обернене нормування даних до реальних вимірів під час навчання та прогнозу нейромережею. Сформовані тензори для навчання нейромережі зберігаються у директорії *data/processed*.

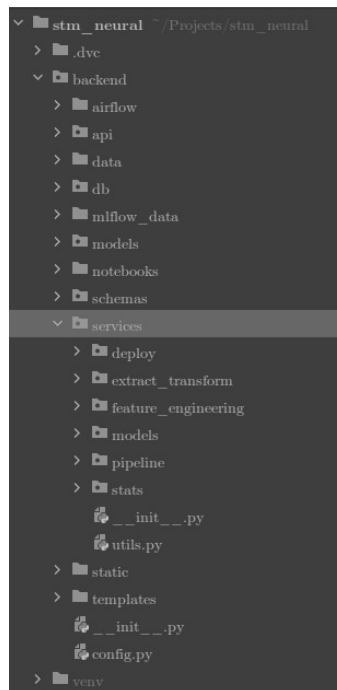


Рис. 3.10. Директорія проекту

Навчання, оптимізація та реєстрація моделей. Побудова і навчання нейромережевої моделі на базі LSTM, або GRU комірок (розділ 2) для прогнозування температури відбувається з використанням програмного каркасу TensorFlow 2. Оскільки передбачені два варіанти запуску навчання моделі: з автоматизованим підбором гіперпараметрів та заздалегідь визначеною її конфігурацією, ці процеси винесені в окремі файли з кодом.

Під час кожного тренування моделей проводиться їх реєстрація та збереження конфігурації засобами сервісу Mlflow (рис. 3.11), що має як веб-інтерфейс, так і можливість доступу через командний рядок. Запуск сервісу Mlflow відбувається командою `mlflow server --backend-store-uri sqlite:///mlflow.db --default-artifact-root ./artifacts --host 0.0.0.0 -p 5000`, де `mlflow.db` — база даних, куди Mlflow зберігає відомості про моделі, `/artifacts` — директорія з артефактами моделі. Доступ до моделі здійснюється через унікальний ідентифікатор, що забезпечує можливість відстежування версій, показників точності моделей тощо для можливості відтворюваності числових експериментів. Модель з найвищою точністю автоматично визначається як основна та надходить у категорію Production (фінальна версія). Збереження моделі відбувається у ієрархічному форматі даних HDF (h5).

На рис. 3.11 показані такі результати навчання і оптимізації гіперпараметрів одношарової рекурентної нейромережі, що були запуснені на мікрокомп'ютері Nvidia Jetson Nano: дата запуску, тривалість тренування, тип комірок та метрики точності.

Оптимізація гіперпараметрів виконується згідно (2.19) за допомогою бібліотеки генетичних алгоритмів DEAP (розділ 2), що інтегрується в код. Конфігурація генетичного алгоритму міститься у файлі `config.yaml`.

Як можна побачити, час затрачений на навчання моделі з одночасною оптимізацією гіперпараметрів сягає порядку 52 хвилин, що є достатньо значним показником. Для зниження часу такої оптимізації можна знижувати кількість популяцій генетичного алгоритму та розмірність простору пошуку. Також, оскільки FastAPI є асинхронним програмним каркасом, процес навчання моделі можна виокремити у фоновий режим засобами AsyncIO [188].

Запуск DAGs в Airflow, доступ через REST API. Процеси отримання, обробки даних та запуску навчання і обчислення гіперпараметрів моделей, що становлять складові DAG-схеми (рис. 3.7)

викликаються завдяки маршрутній функції `run_train()`. Ця та інші маршрутні функції, що відображають URL-адреси у відповідні методи FastAPI, доступні для HTTP-запитів, показані на рис. 3.12.

mlflow Experiments Models

model_iteration1 > recurrent_neural_network

Date: 2022-12-16 17:25:16 Source: airflow User: jetson

Duration: 53.2min Status: FINISHED

Notes [🔗](#)

None

Parameters

Name	Value
cell_type	GRU

Metrics

Name	Value
mape 🔗	0.008
mean_squared_error 🔗	0.074

Tags

Artifacts

- model
 - data
 - keras_module.txt
 - model.h5
 - MLmodel**
 - conda.yaml

Full Path: `./backend/mlflow_data/artifacts/1/1a3d6fdddb24b17b57156a64d502185/artifacts/model/MLmodel`
Size: 315B

```

artifact_path: model
flavors:
  keras:
    data: data
    keras_module: tensorflow.keras
    keras_version: 2.4.0
    python_function:
  
```

Рис. 3.11. Результат реєстрація моделі засобами Mlflow

temperature

- POST `/temperature/create_from_csv` Create Temperature
- POST `/temperature/` Create Temperature
- GET `/temperature/all` Read Temperature All
- GET `/temperature/{id}` Read Temperature All
- GET `/temperature/range/` Read Temperature Range
- GET `/temperature/time/` Read Temperature Time
- POST `/temperature/predict/` Make Prediction
- POST `/temperature/run_train/` Start Pipeline
- GET `/temperature/deploy/` Deploy

Рис. 3.12. Маршрутні функції

На рис. 3.13 наведене вікно сервісу Apache Airflow, який контролює виконання схеми 3.7, запущеної внаслідок надсилання HTTP POST-запиту на локальну URL адресу `http://127.0.0.1:8000/temperature/run_train/` та логує процес її виконання.

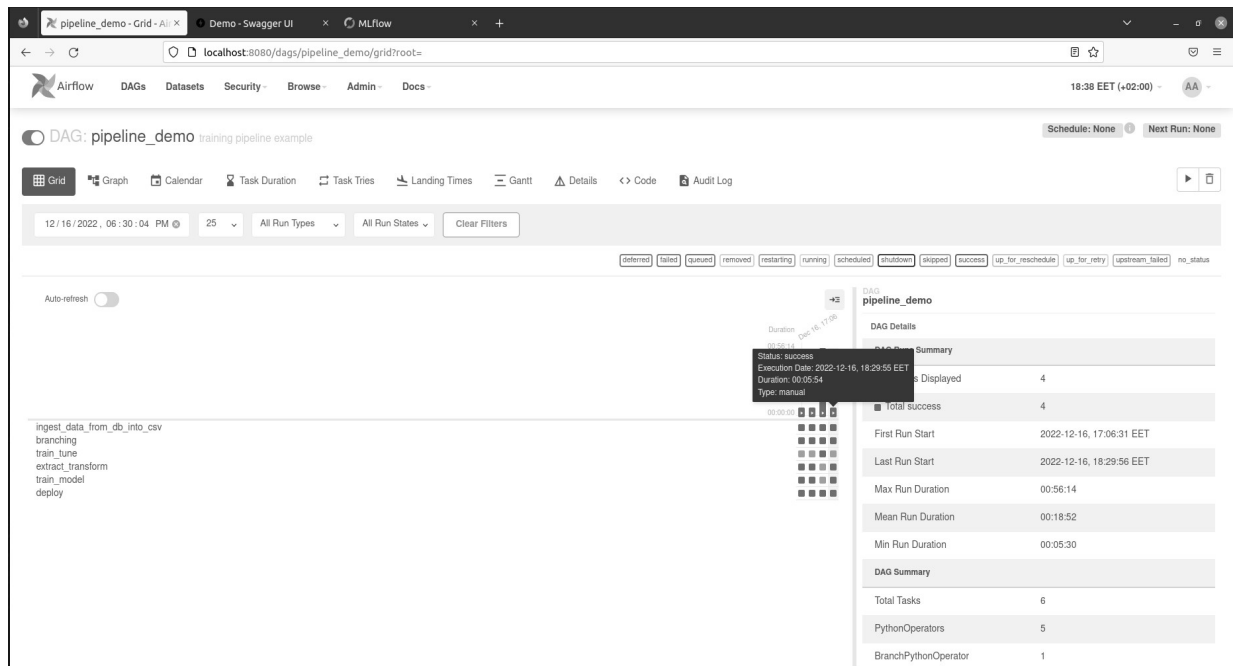


Рис. 3.13. DAG-процеси

Будь-які зміни у схемі 3.7 легко вносять у програмний код, що дозволяє без труднощів модифікувати процес виконання та відстежування його стану завдяки системі Airflow.

Таким чином, старт серверної частини прототипу складається з такої послідовності кроків

1. запуску веб-серверу `uvicorn`, що забезпечує функціонування REST API, створеного на базі асинхронного програмного каркасу FastAPI командою у Linux терміналі `uvicorn main: app -- reload`, де `main` — Python скрипт у кореневій директорії проекту, що містить об'єкт основного застосунку FastAPI;

2. запуску веб-сервісу Mflow для реєстрації моделей протягом їх навчання з наступним їх збереження у форматі h5 командою `mflow`

`server --backend-store-uri sqlite:///mlflow.db --default-artifact-root ./artifacts --host 0.0.0.0 -p 5000` в окремому вікні Linux терміналу;

3. запуску веб-сервісу Apache Airflow для злагодженого виконання схеми 3.8 командами `airflow webserver -p 8082` та `airflow scheduler`, де остання відповідає за старт відповідного планувальника.

Розгортання моделі на нейроконтролері. Перед об'єднанням нейроконтролера з хабом відбувається його стартова ініціалізація файлом моделі, що визначає її топологію. Для цього засобами STM32CubeIDE здійснюється налаштування мікроконтролера STM32 та, зокрема, визначення функції переривання `HAL_UART_RxCpltCallback()` (рис. 3.14), яка спрацьовує при надходженні каналом Rx заданої кількості байтів.

```

84 //=====
85=void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
86 {
87     //printf("Callback triggered\r\n", s);
88     if (((uint32_t*) (&arr)) == 101) || (((uint32_t*) (&arr)) == 102))
89     {
90         idx = ia = is = 0u;
91         code = *((uint32_t*)(&arr));
92         HAL_UART_Receive_IT(&huart2, (uint8_t*)(&arr), 4);
93         return;
94     }
95     switch (code)
96     {
97     case 101:
98         //printf("Arr = %s\r\n", s);
99         //printf("Rcvd = %f\r\n", arr);
100        test_data[idx++] = arr;
101        //HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_7);
102        HAL_UART_Receive_IT(&huart2, (uint8_t*)(&arr), 4);
103        break;
104    case 102:
105        arrays[ia][is++] = arr;
106        if (is == sizes[ia])
107        {
108            ++ia;
109            is = 0;
110        }
111        HAL_UART_Receive_IT(&huart2, (uint8_t*) (&arr), 4);
112        break;
113    }

```

Рис. 3.14. Код налаштування STM32

Після конфігурації STM32 налаштовується мікроконтролер ESP32, що слугує шлюзом для забезпечення мережевого з'єднання системи (рис. 3.15). Збережена засобами Mlflow нейромережева модель конвертується у C-масив ваг послідовно у відповідності кожному шару нейронів/комірок з використанням бібліотеки `keras2c` [189] та зберігається у локальній

директорії на хабі. Тепер конвертована модель може передаватися до ініціалізованого нейроконтролера з використанням сокетів (рис. 3.16).

```
const char* ssid = "KRKS-102";
const char* password = "kreks102";
// Set your Static IP address
IPAddress local_IP(192, 168, 103, 15);
// Set your Gateway IP address
IPAddress gateway(192, 168, 103, 1);

IPAddress subnet(255, 255, 255, 0);
IPAddress primaryDNS(192, 168, 103, 1); //optional
IPAddress secondaryDNS(8, 8, 4, 4); //optional

WiFiServer server(3535);
```

Рис. 3.15. Конфігурація ESP32

```
backend > services > deploy > convert_deploy.py > ...
31
32 arrays = ("gru_79_kernel_array", "gru_79_recurrent_kernel_array",
33 "gru_79_bias_array", "dense_93_kernel_array")
34 def send_array(data,s):
35     for x in data:
36         sent_data = struct.pack("f", x)
37         s.send(sent_data)
38
39
40 def send_code(code, s):
41     sent_code = struct.pack("I", code)
42     s.send(sent_code)
43
44
45 def update_model(config_path="params.yaml") -> str:
46     configs = read_params(config_path=config_path)
47     c_model_path = configs["deploy"]["c_model"]
48     with open(c_model_path, "rt") as f:
49         c_model = f.read()
50
51     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
52     s.connect((HOST, PORT))
53     send_code(102, s)
54     for layer in arrays:
55         n, ar = get_layer(c_model, layer)
56         send_array(ar, s)
57     send_code(103, s)
58     rcvd = s.recv(4)
59     print("Response : ", rcvd.decode())
60     s.close()
61     return rcvd.decode()
62
```

Рис. 3.16. Код передавання моделі за допомогою сокетів

Результат запуску розгортання моделі зображений на рис. 3.17.

The screenshot displays a web browser window with a REST client interface. The URL bar shows 'localhost:8000/docs#/temperature/deploy_temperature_deploy_get'. The main content area shows a GET request to '/temperature/deploy/'. The 'Execute' button is visible. Below, the 'Responses' section shows a 200 status code with a 'null' response body and headers including 'content-length: 4', 'content-type: application/json', 'date: Tue, 10 Jan 2023 13:00:53 GMT', and 'server: uvicorn'. An inset terminal window shows the command 'uvicorn main:app --reload' and its output, including logs for the application startup and successful response.

Рис. 3.17. Результат запуску процесу розгортання моделі на нейроконтролері

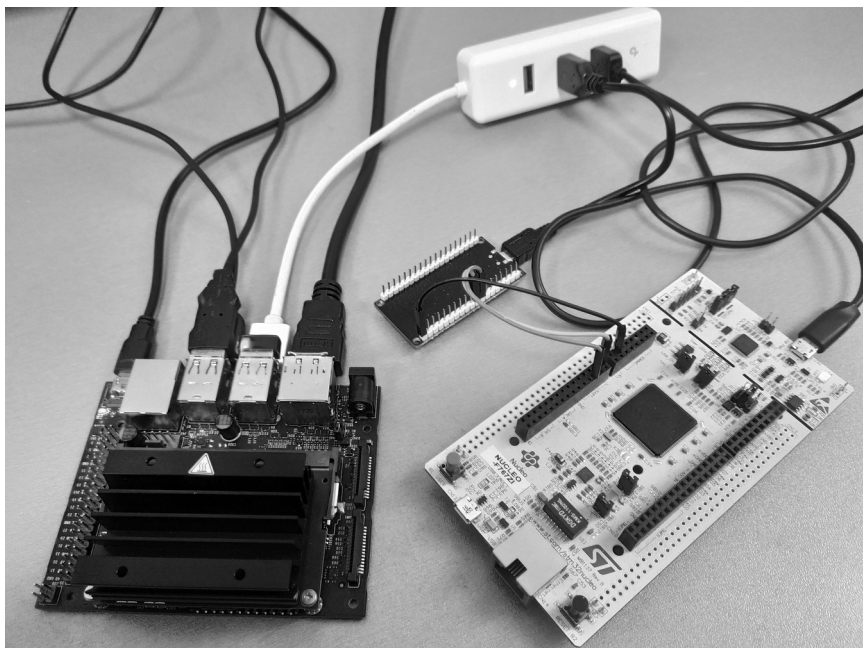


Рис. 3.18. Спроектований нейроконтролерний пристрій

У разі коректного надсилання масивів ваг у нейроконтролер, відбувається отримання статусу на боці хабу щодо успішного завершення

даного процесу у вигляді стрічки-повідомлення, яка обробляється у відповідній маршрутній функції *deploy()*. Після здійснення прогнозу моделлю нейроконтролера, оброблені дані надсилаються на хаб для подальшої обробки та, за необхідності, збереження у базі даних.

Спроектований апаратний прототип нейроконтролерної системи (хаб Nvidia Jetson Nano, ESP32 WROOM та STM32F767 Nucleo) наведений на рис. 3.18.

3.7. Переваги, обмеження та подальший розвиток прототипу

Реалізація ГО для IoT систем на основі інтеграції декількох апаратних компонент містить як переваги над хмарними сервісами, так і низку обмежень, які необхідно усунути в подальшому удосконаленню. До безумовних переваг спроектованого прототипу нейроконтролерної системи належать:

а) автономність — система хоч і ґрунтується на використанні веб-сервісів при її розробці, проте може легко функціювати без доступу до глобальної мережі Інтернет перебуваючи лише в межах локальної мережі;

б) зручність керування, налаштування та доступу — завдяки використанню REST архітектури, швидкого асинхронного програмного каркасу FastAPI та відкритого конфігураційного файлу можна отримати доступ до усіх ключових функцій системи в межах як локальної домашньої, так і глобальної мережі;

в) систему можна легко модифікувати та доповнювати новими службами, зберігаючи її базову архітектуру завдяки сервісу оркестрування процесами Airflow;

г) наявність сервісу реєстрації моделей Mlflow та використання підходів MLOPS дає змогу відстежувати різні моделі прогнозу температур та в подальшому і доповнювати їх багатовимірними даними, наприклад, показниками вологості та якості повітря без суттєвої зміни архітектури системи;

д) перенесення частини обчислень (виконання прогнозування) з хабу на нейроконтролер розвантажує та оптимізує обчислювальні ресурси та створює основу для додавання інших мікроконтролерних рішень з метою масштабування задач.

До обмежень даного прототипу варто віднести ціну — сумарна вартість мікроконтролера STM32 H745 та мікрокомп'ютера Nvidia Jetson Nano є порівняно значною, тому таке рішення у більшості випадків пасуватиме для організацій/домогосподарств з доступним для цього бюджетом; проте варто зазначити, що дана система налаштована на роботу з нейромережами і глибоким навчанням, що забезпечується апаратною конфігурацією Nvidia Jetson Nano; застосування дешевших мікрокомп'ютерів Raspberry Pi 3/4 можуть слугувати однією з альтернатив до даного рішення та в свою чергу потребує додаткового дослідження щодо порівняння ефективності.

Подальший розвиток запропонованого прототипу становить як нарощування апаратної конфігурації (додолучення нових мікроконтролерів, сенсорів для отримання даних температур) з порівнянням енергоефективності, так і внесення удосконалень у програмну частину — її огортання у Docker-контейнери з метою ізоляції коду, використання бібліотеки multiprocessing з AsyncIO для оптимізації часу виконання та оптимального задіяння усіх ядер центрального процесора, створення веб-інтерфейсу до серверної частини засобами HTML5/CSS/JavaScript. А інтеграція даного прототипу ГО з хмарними сервісами як основа до гібридного підходу створює підґрунтя для нового дослідження й ускладнення системи.

Висновки до розділу 3

У даному розділі розроблений прототип апаратно-програмного комплексу оптимізації функціонування інтелектуального будинку, спроектованого в межах парадигми граничних обчислень. Ідеєю створення такого комплексу є локалізація обчислень без залучення хмарних технологій, вивчення процесу розробки задля подальшого впровадження технологій машинного навчання у системи інтелектуального будинку. Для демонстрації концепції прототипу реалізовано та досліджено нейроконтролерну систему прогнозування внутрішніх температур інтелектуального будинку, що поєднує підходи граничних обчислень, машинного навчання і веб-серверну розробку з апаратною інтеграцією мікрокомп'ютер-мікроконтролери. Вперше розроблено апаратно-програмну інтеграцію високопродуктивного мікрокомп'ютера Nvidia Jetson Nano та нейроконтролера — SMT32 H745/F767 з розгорнутою на ньому нейронною мережею для оптимізації роботи інтелектуального будинку.

В апаратній схемі використовується мікроконтролер ESP32 з Wi-Fi модулем для забезпечення з'єднання між хабом та нейроконтролером, що також може використовуватись як додатковий обчислювальний вузол. Нейроконтролер розвантажує хаб й реалізує функцію розподілених обчислень — прогноз температур засобами розгорнутої рекурентної нейромережі та надсилає результати на хаб через UART та сокети.

Обчислювальний хаб виступає сервером з базою даних SQLite для збереження температур, REST архітектурою доступу до функцій системи: автоматизованим циклом навчання, оптимізації гіперпараметрів генетичним алгоритмом та розгортання нейромереж на нейроконтролері. Керування послідовністю цих процесів виконується програмною системною оркестрування Apache Airflow, а обробка і збереження навчених моделей нейромереж — засобами Mlflow. Використані програмні

компоненти визначають конвеєр MLOPS, який забезпечує гнучкість при роботі з проектами машинного навчання.

Для створення серверної частини було проаналізовано та використано швидкий асинхронний програмний каркас FastAPI, що огортає код системи та визначає доступ до функцій через HTTP-запити. Крім роботи з прогностичною моделлю, на сервері міститься код регресійних моделей (розділ 2), які можуть доповнювати результати обробки нейроконтролером.

Розроблений прототип демонструє перспективність локальних та автономних рішень граничних обчислень для інтелектуального будинку та окреслює кроки до розвитку системи: під'єднання сенсорів температури та доповнення їх іншими даними (вологістю, показниками якості повітря тощо) з відповідним подальшим проектуванням нових моделей машинного навчання.

РОЗДІЛ 4. РОЗРОБКА ТЕПЛОФІЗИЧНИХ МОДЕЛЕЙ ІНТЕЛЕКТУАЛЬНОГО БУДИНКУ

Проектування кліматичних систем ІБ зазвичай ґрунтується на накопиченні вимірювань (температур, вологості, споживання енергії тощо), їх аналізі засобами статистичних підходів та в подальшому використанні в якості навчальних вибірок для моделей машинного навчання, які розгортаються або на локальному пристрої, або на віддаленому хмарному сервісі. Не зважаючи на ефективність таких систем, часто є доцільним доповнювати розроблені моделі фізичними рівняннями, які не лише описують теплову поведінку в ІБ, а й можуть інформативно поєднуватися з алгоритмами статистичного аналізу даних у задачах теплопереносу [153]. Наприклад, знання про теплоємність та теплопровідність стін дозволяє коректувати параметри обігріву в окремому приміщенні та використовувати їх значення в проектуванні інтелектуального термостату.

Такі параметри можна обчислити шляхом розв'язування відповідного рівняння теплопровідності. Оскільки динамічне рівняння теплопровідності містить невідомі коефіцієнти теплопровідності та теплоємності, а також й розподіли температур та компоненту, що відповідає за джерело енергії в контексті ІБ, а їх точні значення є невідомими, постає необхідність у формулюванні обернених задач відносно невідомих параметрів [5].

У даному розділі запропонована методика для визначення таких характеристик на базі аналізу даних температур і споживання енергії та розв'язування оберненої задачі теплопровідності. Також, на базі цих даних і рівняння, розроблений підхід до оцінки ефективних теплових

параметрів індивідуальних джерел обігріву (радіаторів), які можуть використовуватися у задачах дезагрегації споживання газу.

4.1. Постановка задачі оцінки теплових характеристик ІВ

Розглянемо будівлю, повністю інтегровану із датчиками внутрішньої/зовнішньої температури, розумними термостатами та датчиками температури, встановленими на поверхнях радіаторів. Інтегральні (усереднені) теплофізичні параметри будівлі можна оцінити не лише за відомими фізичними властивостями стін і рівня ізоляції, але й за розв'язком динамічного рівняння теплопровідності. До таких параметрів слід віднести ефективні коефіцієнти теплопровідності та теплоємності, визначені для будівлі в цілому. В якості даних, необхідних для проведення моделювання, були вибрані наступні вимірювання, отримані з відкритої бази даних REFIT (рис. 2.1-2.4):

а) середні температури $\bar{T}(t)$ у будинку, які визначалися на основі відповідних кімнатних температур, де t — часова змінна;

б) зовнішня температура $T_e(t)$;

в) потужність джерел обігріву $Q(t)$, визначена на основі сумарного споживання газу у будинку. Оскільки доступні дані споживання газу вимірювалися в кубічних метрах, був виконаний перерахунок у кВт-год згідно [96].

Диференціальне рівняння теплопередачі, застосоване до процесу опалення будівлі (рис. 4.1, де Q_e — зовнішні потоки тепла від стін будинку), можна представити у наступному вигляді:

$$c(\mathbf{r})\rho(\mathbf{r})\frac{\partial T(\mathbf{r}, t)}{\partial t} = \text{div}(\kappa(\mathbf{r})\nabla T(\mathbf{r}, t)) + Q(\mathbf{r}, t), \quad (4.1)$$

де $c(\mathbf{r})$ — питома теплоємність, $\rho(\mathbf{r})$ — густина середовища, в якому поширюється тепло, $\mathbf{r} = \mathbf{r}(x_1, x_2, x_3)$ — радіус-вектор точки у просторі, $\kappa(\mathbf{r})$ — коефіцієнт теплопровідності, $T(\mathbf{r}, t)$ — температура у приміщенні, $Q(\mathbf{r}, t)$ — локальне значення теплової потужності, що виділяється в будинку.

Рівняння (4.1) необхідно доповнити граничними умовами, які описують потік тепла через поверхню виділеного об'єму.

$$-\kappa \frac{\partial T(\mathbf{r}, t)}{\partial \mathbf{n}} = \alpha(T_e - T), \quad (4.2)$$

де \mathbf{n} — зовнішня нормаль до поверхні, α — коефіцієнт теплопередачі, T_e — температура зовні виділеного об'єму.

Інтегруючи рівняння (4.1) по об'єму будинку V та використовуючи теорему Остроградського-Гаусса, отримуємо таке диференціальне рівняння для середньої температури в кімнаті $\bar{T}(t)$:

$$c^* \frac{\partial \bar{T}(t)}{\partial t} = -\kappa^* (\bar{T}(t) - T_e(t)) + \bar{Q}(t), \quad (4.3)$$

де $c^* [\text{Дж}/(\text{м}^3\text{К})]$ — ефективна (усереднена) об'ємна теплоємність, $\kappa^* [\text{Вт}/(\text{м}^3\text{К})]$ — нормований на V коефіцієнт теплопередачі зовнішніх стін будинку, який пропорційний їх площі та коефіцієнту теплопровідності і обернено пропорційний середній товщині стіни, $\bar{Q}(t)$ — середня потужність, яка виділяється в 1 м^3 будинку.

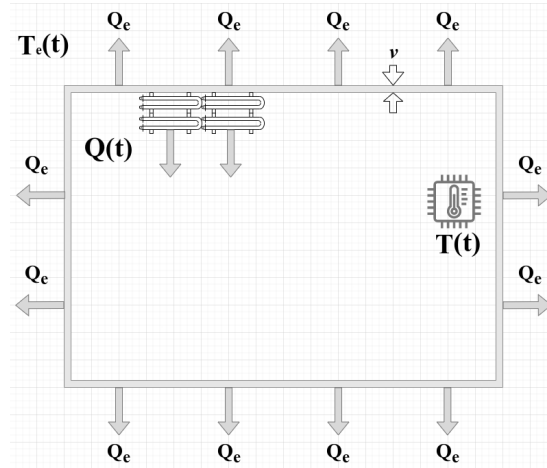


Рис. 4.1. Схема теплового процесу у будинку

Зазначимо, що рівняння (4.3) можна легко отримати, розглядаючи тепловий баланс у будинку.

Ввівши різницю температур $T^*(t) = \bar{T}(t) - T_e(t)$, рівняння (4.3) запишемо у вигляді, зручному для подальшого числового розв'язку

$$c^* \frac{\partial T^*(t)}{\partial t} = -\kappa^* T^*(t) + \bar{Q}(t) - c^* \frac{\partial T_e(t)}{\partial t}. \quad (4.4)$$

Рівняння (4.4) можна використовувати для розрахунку динаміки зміни температури у будинку, а також для знаходження ефективних коефіцієнтів c^* та κ^* (обернена задача).

Для розв'язування рівняння (4.4) здійснюємо його дискретизацію за часом t , що дозволяє використовувати реальні вимірювання температур та потужності, для дослідження діапазонів значень параметрів ефективної теплоємності c^* та теплопередачі κ^* .

Дискретне представлення рівняння (4.4), побудоване за допомогою методу скінченних різниць [5], має наступний вигляд:

$$c^* \frac{T^*(t_i) - T^*(t_{i-1})}{t_i - t_{i-1}} = -\kappa^* T^*(t_i) + \bar{Q}(t_i) - c^* \frac{T_e(t_i) - T_e(t_{i-1})}{t_i - t_{i-1}}, \quad (4.5)$$

де t_i та t_{i-1} — позначають послідовні моменти часу. Рівняння (4.5) доповнюється початковою умовою відомої температури в приміщенні t_0^* в моменту часу t_0 :

$$T^*(t_0) = t_0^*. \quad (4.6)$$

Тут ми припускаємо, що $\bar{Q}(t)$ і $T_e(t)$ є відомими функціями, визначеними для $t > t_0$.

Модель (4.5)-(4.6) визначає пряму задачу теплового стану будинку, яка буде надалі використовуватися для вивчення розподілів параметрів ефективної теплоємності c^* та теплопередачі κ^* .

4.2. Розв'язування та дослідження прямої задачі

Застосуємо рекурсивну різницеву схему для співвідношення між двома значеннями температури, розрахованими на поточному та попередньому кроках. Невідома температура $T^*(t)$, отримана з рівнянь (4.5)-(4.6) визначається за допомогою рекурентного співвідношення наступним чином:

$$T^*(t_i) = \frac{\bar{Q}(t_i) \Delta t_i - c^* \Delta T_{ei} + T^*(t_{i-1})}{c^* + k^* \Delta t_i}, \quad (4.7)$$

де $\Delta t_i = t_i - t_{i-1}$ — часовий крок, який відповідає моментів часу t_i , $\Delta T_{ei} = T_e(t_i) - T_e(t_{i-1})$.

У рівнянні (4.7) $\bar{Q}(t_i)$ — дискретне представлення функції споживання газу, яка залежить від використаного режиму обігріву (вмикання/вимкнення). Можливість розривів цієї функції, зумовлених вмиканням та вимиканням газу призводить до необхідності застосовувати модифіковані різницеві методи [190], які використовують процедури згладжування. При розрахунках було використано інформацію про мінімальну тривалість споживання газу для вибору найкращого кроку рекурсивної схеми:

$$\sum_{i=1}^{N_p} \frac{|T^*(t_i)_{\Delta t_i^j} - T^*(t_i)_{\Delta t_i^{j-1}}|}{|T^*(t_i)_{\Delta t_i^j}|} \leq \epsilon, \quad (4.8)$$

де $T^*(t_i)_{\Delta t_i^j}$ — значення температури в момент часу t_i , обчисленої за часовим кроком Δt_i^j , $\Delta t_i^j \in [0.1, 0.25, 0.5, 1]$, що відповідає 10, 15, 30 та 60 хвилинним крокам дискретизації, а $\epsilon > 0$ деяке достатньо мале значення, N_p — загальна тривалість вимірювань. Для обчислення значень в моментах часу, не заданих під час вимірювання необхідно використати механізм чисельної інтерполяції.

Незважаючи на те, що дані про енергоспоживання відомі протягом усього року, інформація про точну кількість енергії, використаної лише для процесу опалення, є недоступною. Щоб вирішити цю проблему, було застосовано наступний евристичний алгоритм:

1) протягом неопалюваного періоду функція споживання газу була проінтегрована, а відповідне середнє значення було встановлено як частка енергії, яка використовується в опалювальному періоді;

2) розраховане значення було віднято з даних за холодний період, а відповідні перераховані дані були підготовлені для використання в чисельних експериментах.

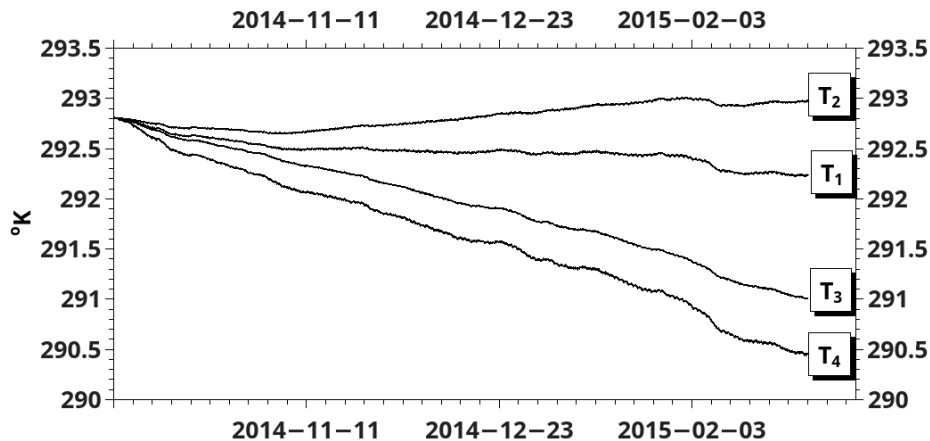


Рис. 4.2. Обчислені значення температури $T(t)$ для набору параметрів c^* та κ^*

На рис 4.2 наведені обчислені розподіли температури в приміщенні, розраховані для холодного періоду (жовтень 2014 р. – лютий 2015 р.). Ці розрахунки були зроблені для кількох пар параметрів $\{c^*, \kappa^*\}$: $\{5 \cdot 10^3, 0.5\}$, $\{6 \cdot 10^3, 0.4\}$, $\{8 \cdot 10^3, 0.8\}$, $\{4 \cdot 10^3, 0.7\}$, що відповідає температурам T_1 , T_2 , T_3 та T_4 . Вхідні дані для розрахунків (початкова внутрішня/зовнішня температура, дані про енергоспоживання) були вибрані з даних REFIT (рис. 2.1-2.4). Графіки на рис. 4.2 демонструють суттєву чутливість розв'язків відносно різних значень параметрів c^* та κ^* .

У свою чергу, на рис. 4.3 наведено приклади розв'язків рівняння (4.7), а саме $T^*(t_i)$, розрахованих для різних наборів c^* та κ^* , заданих у вигляді сітки значень: $c^* = [1.2 \cdot 10^3, 10^4]$, $\kappa^* = [10^{-1}, 1]$. Тут для комбінації параметрів c^* та κ^* розв'язки $T^*(t_{100})$ та $T^*(t_{500})$ були розраховані для моментів часу t_{100} та t_{500} протягом холодного періоду. Як можна побачити, кожна з побудованих поверхонь спадає до деякого мінімального значення, тому для визначення оптимальних параметрів c^* та κ^* можна застосувати алгоритм мінімізації в рамках розв'язання оберненої задачі.

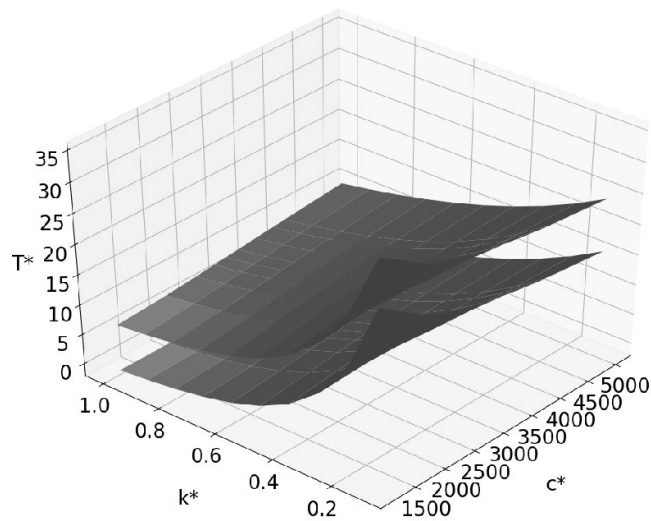


Рис. 4.3. Обчислені значення температури $T^*(t_{100})$ та $T^*(t_{500})$ для набору параметрів c^* та κ^*

4.3. Розв'язування та дослідження оберненої задачі

Розглянемо підхід до обчислення ефективних теплових параметрів c^* та κ^* ґрунтуючись на розв'язуванні оберненої задачі, сформульованої у вигляді мінімізації наступної функції

$$\Phi(c^*, \kappa^*) = \sum_{i=1}^{N_p} \left(\frac{T^*(t_i; c^*, \kappa^*) - T_{actual}^*(t_i)}{|T^*(t_i; c^*, \kappa^*)|} \right)^2 \rightarrow \min_{c^*, \kappa^*}, \quad (4.9)$$

де $T^*(t_i; c^*, \kappa^*)$ — розрахована температура як розв'язок відповідної прямої задачі (4.5-4.6) для заданих параметрів c^* та κ^* , а $T_{actual}^*(t_i)$ — реальне значення температури за час t_i . Тут здійснюється нормування значень різниці обчисленої та реальної температур для запобігання накопичення квадратів значень до надто великого значення через значну довжину вимірювань.

Ця задача вимагає застосування алгоритму мінімізації для пошуку глобального мінімуму. Для розв'язування використаємо алгоритм грубої

сили (brute force) для отримання початкового наближення розв'язку та квазі-ньютонівський метод з обмеженою пам'яттю BFGS [191] для отримання більш точного рішення.

Для застосування брутального перебору формуємо сітку \mathbf{G} значень параметрів c^* та κ^* : $c^* = [1.2 \cdot 10^3, 10^4]$, $\kappa^* = [10^{-1}, 1]$ з деяким кроком dc^* та $d\kappa^*$ для кожного з параметрів. Для кожної пари $(c^*, \kappa^*) \in \mathbf{G}$ обчислюємо визначаємо значення $T^*(t_i; c^*, \kappa^*)$ з урахуванням кроку згідно (4.8) та визначаємо значення функціоналу Φ . Найменше з отриманих значень Φ на заданій сітці визначає початкове наближення x^* розв'язку, яке використовується у процедурі уточнення.

Уточнення розв'язку реалізовується за допомогою наступного квазі-ньютонівського алгоритму, який базується на процедурі Бройдена — Флетчера — Гольдфарба — Шанно (BFGS). Переформулюємо задачу (4.9) у вигляді

$$\mathbf{x}^* = \operatorname{argmin}_{c^*, \kappa^*} \Phi(\mathbf{x}), \quad (4.10)$$

$\mathbf{x} = (c^*, \kappa^*)^T$ розв'язування якої здійснюємо на основі наступних перетворень

$$\mathbf{p}_i = -\mathbf{C}_i \nabla \Phi(\mathbf{x}_i), \quad (4.11)$$

де \mathbf{C}_i — матриця, яка представляє наближення до оберненої матриці Гессе у формулі Ньютона, $\nabla \Phi(\mathbf{x}_i)$ — градієнт функціоналу (4.9), компоненти якого можна наближено обчислювати методом скінченних різниць

$$\frac{\partial \Phi(\mathbf{x}_i)}{\partial x_i} \approx \frac{\Phi(\mathbf{x}_i + h) - \Phi(\mathbf{x}_i - h)}{2h}, \quad (4.12)$$

де $h > 0, h \ll x_k$ — деяке достатньо мале значення,

$$\mathbf{C}_{i+1} = (\mathbf{I} - \eta \Delta \mathbf{x}_i \mathbf{y}_i^T) \mathbf{C}_i (\mathbf{I} - \eta \mathbf{y}_i \Delta \mathbf{x}_i^T) + \eta \Delta \mathbf{x}_i \Delta \mathbf{x}_i^T, \quad (4.13)$$

де $\eta_i = (\mathbf{y}_i^T \Delta \mathbf{x}_i)^{-1}$, \mathbf{I} — одинична матриця, $\Delta \mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ — крок методу, $\mathbf{y}_i = \nabla \Phi(\mathbf{x}_{i+1}) - \nabla \Phi(\mathbf{x}_i)$.

Кожне $i + 1$ наближення розв'язку (4.9) обчислюється за наступною формулою, де $\alpha_i = \operatorname{argmin} \Phi(\mathbf{x}_i + \alpha \mathbf{p}_i)$, α — параметр методу,

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i. \quad (4.14)$$

Оскільки розв'язок оберненої задачі (4.9) досить чутливий щодо обраного опалювального періоду, розташування кімнат та інших особливостей будівлі, діапазон розрахункових значень c^* та κ^* може бути достатньо широким.

Чисельні експерименти, проведені для наявних даних виявили, що значення c^* змінюється в діапазоні від 1200 до 4200 Дж/(м³К), а значення κ^* від 0.6 до 0.65 Вт/(м³К) за періоди грудень 2014 — лютий 2015 та листопад 2014 — лютий 2015 відповідно.

Розв'язок оберненої задачі, c^* та κ^* , може бути використаний для прогнозування енергоспоживання та формулювання підходів до його зниження. Через чутливість такого розв'язку до вхідних даних, необхідно приділити увагу до вдосконалення методів попередньої обробки, наприклад, використавши алгоритм дезагрегації газу (розділ 2).

Також, запропонований підхід дає можливість наближено оцінювати ефективні теплофізичні характеристики архітектурних споруд, ґрунтуючись лише на вимірюванні температур та показників

енергоспоживання обігріву приміщень. На базі цього алгоритму отримано патент на корисну модель [7].

4.4. Оцінка теплової ефективності джерел обігріву

Дослідження теплової поведінки ІБ в умовах обігріву може бути розширене шляхом ідентифікації та оцінки теплових параметрів обігрівальних елементів, через які визначається як ефективність процесу обігріву, так і неявне обчислення кількості виділеного тепла конкретним обігрівальним елементом за невідомої їх потужності.

Аналізу параметрів системи опалення за допомогою непрямих сучасних підходів, таких як машинне навчання на основі зібраних даних, на даному етапі приділяється не достатньо уваги при проектуванні ІБ, оскільки більшість джерел тепла – це газові/електричні котли (особливо в Європі та Азії), і їх характеристики подекуди важко оцінити.

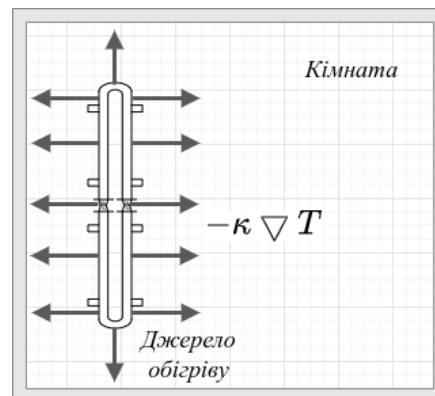


Рис. 4.4. Схема обігріву кімнати

У зв'язку з цим становить інтерес розширення знань про джерела опалення, встановлені в ІБ. У роботі запропоновано підхід до використання розподілів температури в приміщенні та на поверхнях обігрівальних елементів, а також сумарних значень витрат енергії як вхідних параметрів для розробленої моделі на базі рівняння теплопереносу, в якому враховано поведінку кожного джерела обігріву.

Для постановки задачі оцінки ефективних параметрів джерел обігріву (рис. 4.4), переформулюємо рівняння (4.3) до вигляду

$$c_r^* V_r \frac{\partial T_r(t)}{\partial t} = -\kappa S(\nabla T \cdot \mathbf{n}) + Q_r(t), \quad (4.15)$$

де $c_r^* = c\rho$ — ефективна об'ємна теплоємність джерела обігріву, $T_r(t) = \frac{1}{V_r} \int_{V_r} T_r(\mathbf{r}, t) dr$, V_r — об'єм джерела обігріву (батареї), $\kappa^* S(\nabla T \cdot \mathbf{n}) = \kappa^* S(T - T_r)$, κ^* — коефіцієнт теплопередачі (тепловіддачі) батареї, T_r — середня температура на джерелі обігріву, S — площа батареї, $Q_r(t)$ — потужність джерела обігріву.

Легко перевірити, що для кожного приміщення у будинку

$$\kappa^* S(T_r - T) = Q_r(t) - c_r V_r \frac{\partial T_r}{\partial t}, \quad (4.16)$$

де c_r — об'ємна теплоємність джерела опалення з теплоносієм.

Узагальнюючи рівняння (4.16) для усіх приміщень зі встановленими джерелами обігріву і беручи до уваги той факт, що доступна лише сукупна потужність споживання газу, тобто величина, що відповідає обігріву кожної кімнати є невідомою, ми можемо переформулювати (4.16) до наступного представлення:

$$\sum_i (T_{ri} - T_i) \omega_i = \sum_i (Q_i - \frac{c_r V_r}{\Delta t} (T_{ri+1} - T_{ri})) \quad (4.17)$$

де $\omega_i = \kappa^* S$ [Вт/К] — ефективний тепловий параметр джерела обігріву, Q_i — сумарна потужність опалення, Δt — тривалість часового інтервалу $[t_i, t_{i+1}]$.

В свою чергу, оцінка параметру ω_j для кожної j -тої кімнати можна обчислювати за наступним співвідношенням:

$$\omega_j = \frac{S_j \sum_{i=1..n} \omega_i}{n S_{total}}, \quad (4.18)$$

де S_j — площа поверхні батареї, S_{total} — сумарна площа поверхонь батарей у будинку, n — кількість кімнат зі встановленими батареями.

Використовуючи рівняння (4.17), задача ідентифікації ефективних теплових параметрів ω_j у формулюється наступним чином: враховуючи дані про температуру в приміщенні, виміряні в кожному приміщенні з радіатором, загальну теплову енергію, спожиту в будівлі, температуру на поверхнях обігрівальних елементах та площі поверхні батарей, необхідно обчислити параметр ω_j протягом часового інтервалу, відповідному отриманим даним.

4.4.1. Обчислення ефективних теплових параметрів

Для аналізу розподілів температур у приміщенні та на поверхнях батарей, а також даних про енергоспоживання, візуалізуємо ці показники, отримані з довільно обраного зимового дня, коли опалення було ввімкнено.

Як видно з рис. 4.5, температури у кожному з приміщень суттєво не відрізняються між собою, що також справедливо і для температур на поверхнях батарей. Цей факт пояснюється низькою тепловіддачею у будівлі. Сукупне використання газу демонструє чітку кореляцію з температурами на поверхнях радіаторів та дозволяє нам використовувати ці дані в рівняннях (4.17-4.18). Крім того, затримку між підвищенням

температури на поверхнях батареї та температурою приміщення можна розглядати як один з визначників коефіцієнта теплопровідності κ .

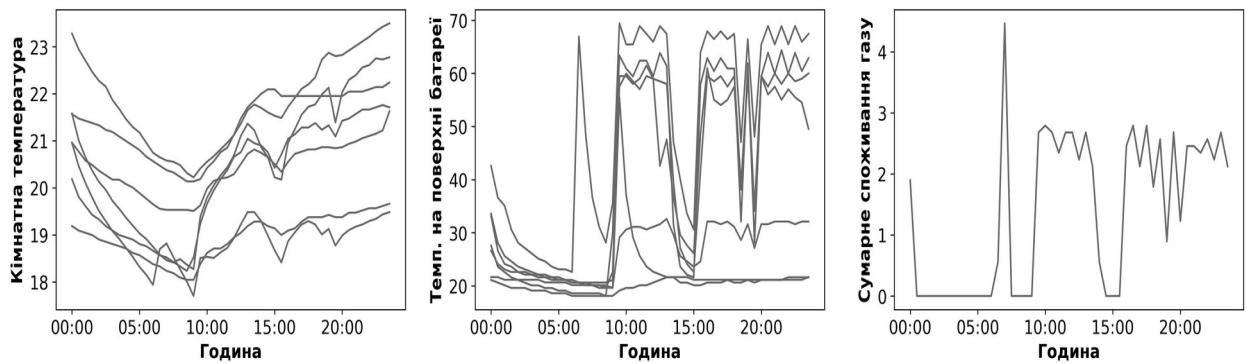


Рис. 4.5. Температури у кімнатах, на батареях та споживання газу

Під час аналізу даного набору даних було з'ясовано, що сукупне споживання газу зазвичай використовується багатьма споживачів, такими як обігрівальні елементи, плити, пральні машини тощо. Тому перед його використанням в рівняння (4.17) (параметр Q_i), необхідно виділити ту частину, яка стосується лише процесу опалення. Для цього застосовано алгоритм, описаний у п. 4.2 даного розділу. В результаті, отриманий розподіл потужностей обігріву показаний на рис. 4.6.

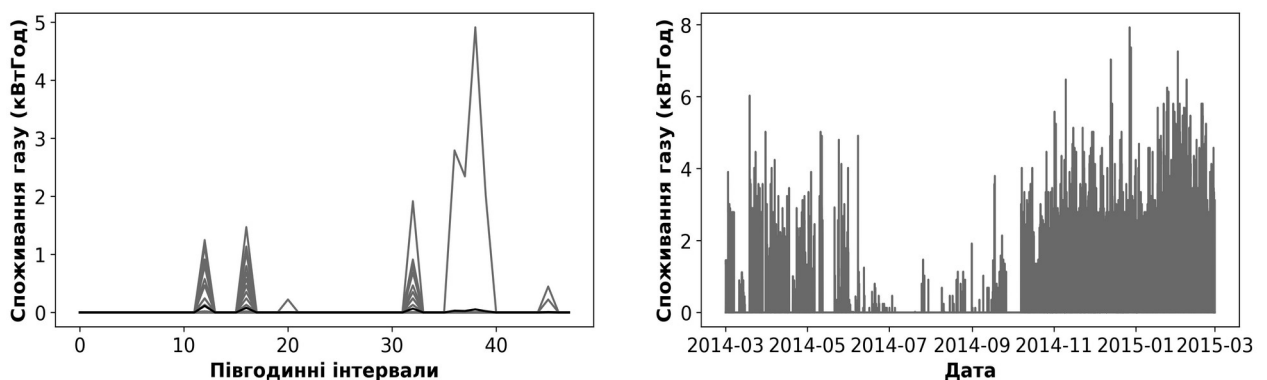


Рис. 4.6. Щоденне використання газу влітку (ліворуч) і річне використання газу для опалення (праворуч)

Оскільки в наявних даних присутні різні довжина часових рядів, що представляють собою вимірювання температур та споживання газу, а також й різні часові кроки, необхідно синхронізувати та об'єднати дані між собою.

Для оцінки ефективних теплових параметрів ω_j розглянемо в першу чергу періоди часу протягом холодної та перехідної пори року (осені та весни). Літні періоди не важливі в цьому дослідженні, оскільки опалення вимкнено, і можна досліджувати лише вплив першого доданку в правій частині рівняння (4.15). Також, розглядаємо лише ті інтервали, де виявляється достатня подібність між векторами, що відповідають поверхневій температурі на батареях та споживанню газу (рис. 4.7).

Для реалізації такої схеми запропоновано наступну схему: обчислення подібності з обов'язковою нормалізацією даних, оскільки кожен часовий ряд має різні діапазони значень.

Нормалізація даних може бути здійснена за допомогою відомої техніки мінімально-максимального масштабування (2.16):

$$y_i = \beta_1 + \frac{(y_i - \min(\mathbf{Y}))(\beta_2 - \beta_1)}{\max(\mathbf{Y}) - \min(\mathbf{Y})}, \quad (4.19)$$

де β_1 та β_2 — нижній та верхній діапазони нормування, $\mathbf{Y} \in \{Q, T_r\}$, $y_i \in \mathbf{Y}$.

Після завершення цього кроку потрібно застосувати етап перевірки подібності. Як тільки виконується умова $x \cdot y > 0$, $x = T_{ri+1} - T_{ri}$, $y = Q_{i+1} - Q_i$ для перевірки заданих інтервалів використовується міра косинусної подібності:

$$\cos_{similarity} = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \geq \varepsilon, \quad (4.20)$$

де x та y — вектори на кожному часовому інтервалі, $\varepsilon > 0$ — деяке достатньо мале значення.

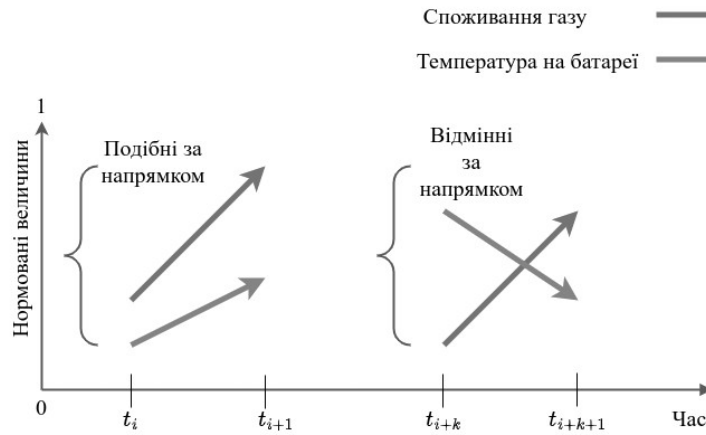


Рис. 4.7. Приклади подібності та відмінності векторів зміни температури та споживання газу

Тут дана схема визначена лише для однокрокового інтервалу, заданого двома послідовними точками, проте в загальному випадку цей підхід можна розширити для роботи з інтервалами різної довжини.

Застосування рівнянь (4.19) і (4.20) до синхронізованих даних $Y \in \{Q, T_r\}$ призводить до накопичення нових інтервалів з подібними векторами, що відповідають Q , T_r та T . Останні використовуються для розрахунку параметрів ω_i по кожному з отриманих інтервалів:

$$\omega_i = \frac{\int_{t_i}^{t_{i+1}} Q(x) - \frac{c_r V_r}{\Delta t} (T_r(y) - T_r(x)) dx}{\int_{t_i}^{t_{i+1}} T_r(x) - T(x) dx} \approx \frac{\Delta Q - \frac{c_r V_r}{\Delta t} \Delta T_r}{\Delta T}, \quad (4.21)$$

де $y \in [t_{i+1}, t_{i+2}]$, $\Delta Q = \frac{Q(t_{i+1}) + Q(t_i)}{2}$, $\Delta T_r = T_r(t_{i+1}) - T_r(t_i)$,
 $\Delta T = \frac{(T_r(t_{i+1}) - T(t_{i+1})) + (T_r(t_i) - T(t_i))}{2}$.

У рівнянні (4.21) права частина включає, за винятком компоненти ΔQ , доданок, пов'язаний з динамікою температури ΔT_r . На основі вищезазначеного рівняння (4.21) виконуємо такі кроки (рис. 4.8):

1) для зимового, весняного та осіннього сезонів розраховуємо набір значень ω_i ;

2) на основі цих значень обчислюємо середнє значення $\bar{\omega} = \frac{\sum_{i=1..n} \omega_i}{n}$;

3) використовуємо значення $\bar{\omega}$ для визначення конкретних значень ω_j для кожної кімнати в досліджуваній будівлі за допомогою множника $\frac{S_j}{S_{total}}$.

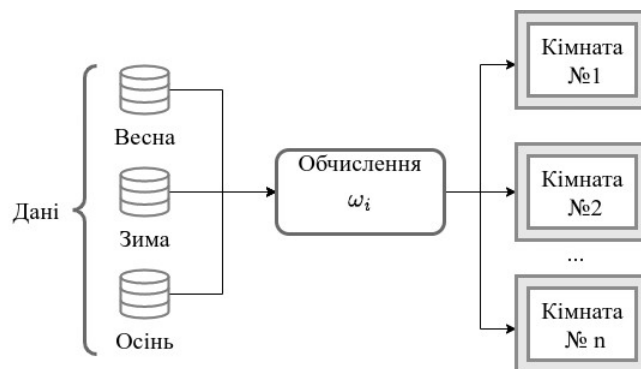


Рис. 4.8. Схема алгоритму обчислення ω_j

На рис. 4.9, 4.10 і 4.11 зображений розподіл ω_i протягом зими, осені та весни відповідно. Тут подані результати з урахуванням параметру ΔT_r (повний алгоритм) та без нього (спрощений алгоритм).

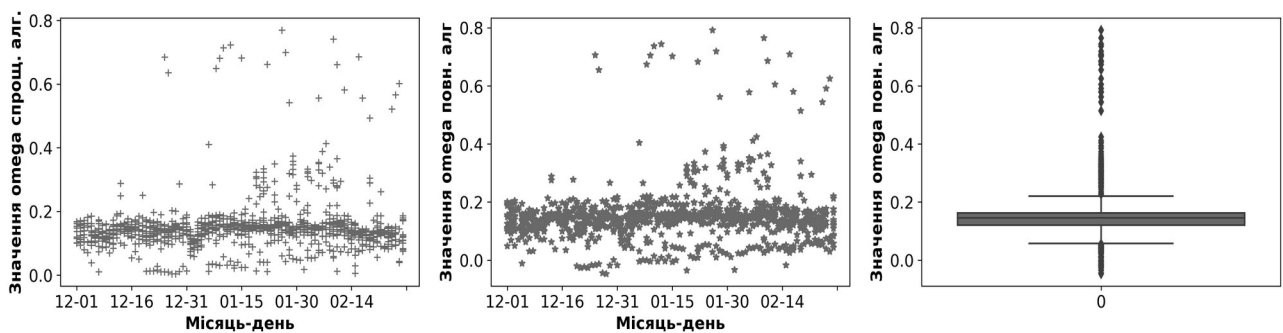


Рис. 4.9. Результати ω_i за зимовий період

Для обчислень приймаємо коефіцієнт теплоємності $c_r \approx 3 \cdot 10^{-2}$ кВт-год/К. Значення об'єму $V_r \approx 0.05$ м³ обчислено наближено, ґрунтуючись на розмірах батарей. Третій графік на кожному рисунку — це коробкова діаграма, яка вказує на можливі аномальні значення ω_i .

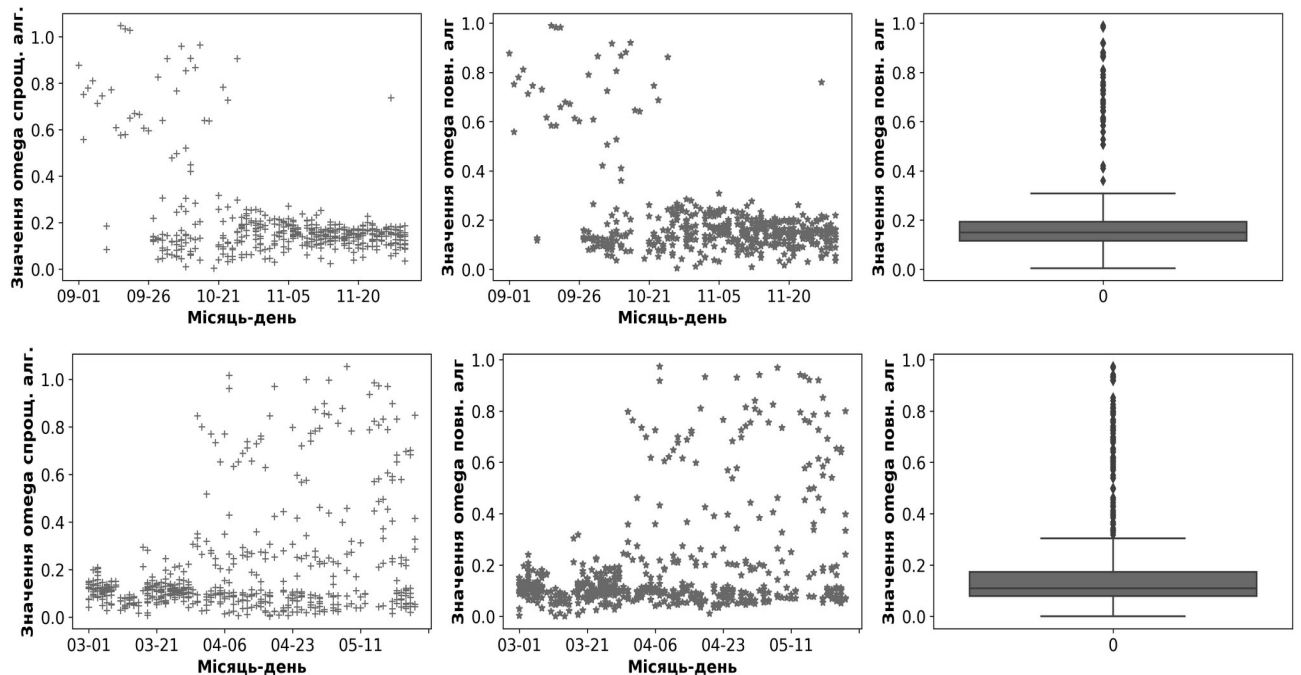


Рис. 4.10. Результати ω_i за весняний та осінній періоди

З коробкових діаграм видно, що для кожного досліджуваного сезону існує досить значний розкид значень ω_i . Така поведінка зумовлена як похибками розрахунків, так і помилками у вхідних даних, отриманими під час вимірювань. Тому було застосовано простий метод виявлення аномалій на основі інтерквартильного діапазону (IQR), щоб усунути небажані значення. Цей метод використовує перший і третій статистичні квартилі, щоб виділити ті значення, які лежать за межами згаданих інтервалів.

Зважаючи на вищезазначене, було отримано результати для $\bar{\omega} = \frac{\sum_{i=1..n} \omega_i}{n}$, розрахованих для кожного з сезонів, де n — кількість

часових інтервалів протягом кожного з сезонів (зима, осінь, весна). Таблиця 4.1 деталізує ці результати.

Сезон	Обчислення за спрощ. алг.	Обчислення за повн. алг.
Зима	0,0136	0,0138
Весна	0,0166	0,0167
Осінь	0,0179	0,0182

Табл. 4.1. Значення $\bar{\omega}$ для кожного сезону

Згідно з таблицею 1, при дослідженні значень параметрів $\bar{\omega}$, розрахованих для кожного з трьох сезонів, суттєвих відмінностей не виявлено. Крім того, ці висновки підтверджують пропозицію знехтувати другим членом правої частини в рівнянні (4.17), оскільки це не спричиняє жодних істотних змін кінцевих результатів. Відмінності ω_i протягом досліджуваних сезонів пов'язані з сумарними похибками обчислень та наявністю викидів даних.

Для визначення значень ω_j , що відповідають конкретному джерелу обігріву, використаємо формулу (4.18), в результаті чого отримаємо:

Кімн.№	1	2	3	4	5	6	7	8	9
ω_j	0.0016	0.001	0.003	0.0018	0.0018	0.0017	0.0018	0.0019	0.0012

Табл. 4.2. Значення ω_j для кожної батареї у кімнаті

Результати цього дослідження окреслюють ідею, що ефективні теплофізичні параметри джерел опалення можна оцінити за допомогою запропонованої моделі та даних, зібраних в ІБ за допомогою датчиків температури та лічильнику споживання енергії обігріву. Одним із можливих практичних застосувань, заснованих на запропонованій методиці, є рекомендаційна система щодо енергозбереження ІБ, яка дозволяє мешканцям виявляти несправність, або підготувати

оптимізований планувальник опалення/охолодження. Основним результатом застосування такої методики є використання обчислених параметрів ω_j для проблеми дезагрегації енергії, тобто розбиття агрегованих даних енергію на класи, що відповідають часовим інтервалам і конкретним споживачам енергії.

4.4.2. Метод уточнення ефективних теплових параметрів

Через похибки, що виникли під час обчислень та наявність некоректних значень у даних часових рядів, результати значень параметрів можуть бути уточнені. До прикладу, на рис. 4.11 показаний розподіл значень ω_i для кожного з виділених інтервалів за схемою (п. 4.3.1).

Розкид обчислених значень, зокрема спричинений не синхронною поведінкою між векторами сумарної різниці між температурою радіатора та кімнатною (табл. 4.3) відбувається через: 1) помилки зчитування; 2) наявність термостатичних радіаторних вентилів, тобто мешканці будинку можуть налаштувати потужність тепла в конкретному приміщенні. Також не було враховано електронагрівальні прилади, які можуть викликати підвищення температури в приміщенні.

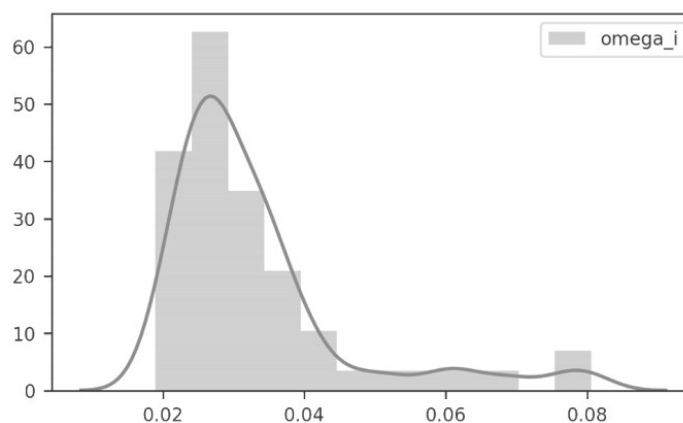


Рис. 4.11. Розподіл ω_i

Дата	Споживання газу	$\sum_j (T_{ri}^j - T_i^j)$
2015-01-04 05:00:00+00:00	1.786211	152.3355
2015-01-04 05:30:00+00:00	2.232764	200.3825
2015-01-04 06:30:00+00:00	2.344402	195.9065

Табл. 4.3. Приклади інтервалів з відповідними значеннями різниць температур

Для цього розглянемо підхід, який передбачає багатовимірну мінімізацію функції втрат за допомогою генетичного алгоритму [].

Формулу (4.18) можна доповнити деяким коефіцієнтом пропорційності λ_j :

$$\omega_j \sim \lambda_j \bar{\omega} \frac{S_j}{S_{total}}. \quad (4.22)$$

Під час обчислювальних експериментів було виявлено, що значення λ_j можна наближено встановити як $\lambda_j = 4$. Частина результатів після початкової корекції λ_j наведена в таблиці 4.4, де

$$e_i = \left| Q_i - \sum_j (T_{ri}^j - T_i^j) \lambda_j \omega_j \right|. \quad (4.23)$$

Для уточнення результатів (табл. 4.4), можна застосувати наступний метод, який ґрунтується на мінімізації функції:

$$f_e = \sum_{I \in \mathbf{I}} \int_I \left(Q_i - \sum_j (T_{ri}^j - T_i^j) \lambda_j \omega_j \right)^2 \rightarrow \min_{\lambda_i \in \Lambda}, \quad (4.24)$$

де I — множина всіх виділених інтервалів подій, $\Lambda = [1, 10] \times [1, 10] \dots [1, 10]$. Тут обмежуємось діапазоном зміни параметру λ_i в $[1, 10]$, проте він може змінюватися в залежності від різних умов.

Дата	Споживання газу	e_i
2014-12-09 3:30:00+00:00	2.12	0.58
2014-12-09 4:00:00+00:00	1.89	0.37
2014-12-09 4:30:00+00:00	2	0.22
2014-12-09 5:00:00+00:00	2.12	0.56
2014-12-09 5:30:00+00:00	2	0.3

Табл. 4.4. Приклади інтервалів з відповідними значеннями різниць температур

Мінімізація (4.24) є складною багатовимірною проблемою, яка вимагає значних обчислювальних ресурсів. Серед низки алгоритмів мінімізації (спряжені градієнти, методи Ньютона, Нелдера-Міда) було обрано еволюційний алгоритм, перевагами якого є різноманіття варіантів налаштування параметрів, відсутність розрахунків похідних; уникнення збіжності до локальних мінімумів; розпаралелювання; стійкість до шуму; здатність знаходити рішення у багатовимірному просторі тощо. Для конкретної проблеми було застосовано метод рою частинок (PSO) [192]. Суть цього методу полягає у генерації розв'язку на основі обчислення популяцій у всьому просторі пошуку та їх порівняння при виборі оптимального рішення на даному кроці ітерації.

Для обчислення були встановлені наступні параметри методу PSO: $N = 10$ (розмір популяції), $G = 100$ (кількість частинок у рої). Результати розрахунків значень λ_j наведено в таблиці 4.5.

λ_j	5.8	8.9	9.8	9	3.2	2.9	4.7
-------------	-----	-----	-----	---	-----	-----	-----

Табл. 4.5. Результати обчислень λ_j

Ґрунтуючись на результатах розв'язування задачі (4.24) було отримано наступне значення середньоквадратичної похибки

$$mse = \frac{1}{n} \sum_{i=1}^n (Q_i - \sum_j (T_{ri}^j - T_i^j) \lambda_j \omega_j)^2 = 0.66 \text{ та значення } f_e = 37.1.$$

Розраховані параметри ω_j можуть використовуватися при розв'язуванні задачі дезагрегації енергії, тобто виявлення частки енергії споживання для цілей, не пов'язаних з опаленням у відповідності до часової змінної. Знаючи розподіл температури на панелях батарей та у кімнатах можна визначити кількість спожитої теплової енергії конкретним нагрівальним елементом.

Знаючи таку кількість енергії, можна побудувати моделі теплової поведінки у кожній з кімнат на основі дерев рішень та оцінювати аномальні показники, наприклад, співставляти параметри тепловіддачі від обігрівального елемента з градієнтом температури у кімнаті тощо.

Висновки до розділу 4

У даному розділі дисертаційної роботи побудовані теплофізичні моделі ІБ на основі рівнянь теплопереносу у будівлі та у кімнатах. Метою такого моделювання є:

➤ оцінка ефективних теплофізичних параметрів будівлі: коефіцієнтів теплопровідності та теплоємності шляхом розв'язування прямої та відповідної оберненої задач за даними споживання енергії та усередненої температурної динаміки у будинку;

➤ оцінка ефективного теплофізичного параметра кімнатного джерела обігріву (теплова ефективність обігрівального елементу) шляхом розв'язування відповідної задачі теплопереносу, сформульованої з урахуванням виведеного параметру.

Обчислення ефективних теплофізичних параметрів будівлі відбувається на основі розподілів зовнішніх і внутрішніх температур упродовж холодної пори року та даних споживання енергії за наявності обігріву. З використанням методу скінченних різниць сформульована та розв'язана пряма задача визначення температури в ІБ, параметрами якої є відповідні теплофізичні характеристики; запропонована чисельна схема розрахунку кроку методу; показаний вплив теплофізичних параметрів на значення розрахованих температур. Базуючись на розв'язках прямої задачі, сформульована та розв'язана обернена задача ідентифікації ефективних теплових параметрів будівлі. Для її розв'язування були використані методи брутального перебору для отримання наближеного розв'язку та квазіньютонівський алгоритм BFGS для його уточнення. Запропонований підхід покладений в основу патенту на корисну модель.

Для визначення теплової ефективності джерел обігріву, запропонована методика на базі розв'язку задачі теплопровідності. Дана методика передбачає виділення інтервалів подій, під час яких

відбувається зростання споживання енергії і різниць температур на обігрівальними елементах та у кімнатах відповідно. Виділені інтервали використовуються для обчислень параметрів джерел обігріву із залученням різницевої схеми для розв'язування задачі теплопровідності. Аналіз результатів продемонстрував потребу в уточненні обчислених параметрів, з метою чого запропонований підхід, який ґрунтується на розв'язуванні додаткової задачі мінімізації похибки обчислень шляхом введення допоміжного коефіцієнту пропорційності. Процес мінімізації проводився еволюційним методом рою частинок. Отримані таким чином ефективні теплофізичні коефіцієнти індивідуальних джерел обігріву, що визначають їх теплову ефективність можна використовувати при обчисленні дезагрегації споживання енергії та реалізації моделей теплової поведінки в ІБ.

Результати та методики, викладені у цьому розділі, доповнюють алгоритми оптимізації функціонування ІБ.

ВИСНОВКИ

У дисертаційній роботі розроблені нові підходи до оптимізації функціонування інтелектуальних об'єктів на прикладі інтелектуального будинку. Такі підходи ґрунтуються на поєднанні методів машинного і глибокого навчання, статистичної обробки кліматичних даних, генетичної оптимізації, теплофізичного моделювання з реалізацією апаратно-програмного прототипу нейроконтролера. У межах здійсненого дослідження та розробки отримано наступні основні результати:

1) згідно з парадигмою граничних обчислень побудовано прототип апаратно-програмного комплексу для оптимізації енергоспоживання та прогностичного аналізу температурних даних інтелектуального будинку, що передбачає інтеграцію мікрокомп'ютера Nvidia Jetson Nano та нейроконтролера на базі мікроконтролера STM32;

2) розроблено програмне забезпечення нейроконтролерної системи з використанням мови Python 3, програмного каркасу TensorFlow 2, бібліотеки генетичної оптимізації DEAP в межах архітектури REST згідно підходів MLOPS з застосуванням сервісів Mlflow та Apache Airflow, яке забезпечує обробку даних інтелектуального будинку в реальному часі та їх використання для прогнозування температурних профілів;

3) побудовано модель прогнозування внутрішньої температури на базі рекурентних нейромереж, що дозволяє підвищувати ефективність роботи систем автоматизації для забезпечення теплових режимів в інтелектуальному будинку;

4) реалізовано автоматизований процес машинного навчання й обчислення гіперпараметрів нейромережевої моделі з подальшим її розгортанням на нейроконтролері, що забезпечує гнучкість системи щодо її масштабування та ускладнення;

5) розвинуто та апробовано моделі та методи дезагрегації даних споживання енергії, що базуються на навчанні без учителя та виділенні ознак і дають змогу кількісно визначати, яку частину з сумарної енергії використовує конкретний споживач;

6) розроблено регресійні моделі для зв'язку внутрішніх і зовнішніх температур та споживання газу в інтелектуальному будинку, які дозволяють прогнозувати часову динаміку зазначених показників;

7) запропоновано модель оцінки теплофізичних параметрів інтелектуального будинку, а саме ефективних коефіцієнтів теплопровідності та теплоємності на основі температурної динаміки та відповідних даних споживання енергії для обігріву, яка ґрунтується на розв'язуванні прямої та оберненої задач теплопереносу; отримано патент на корисну модель;

8) розроблено модель непрямої ідентифікації ефективного теплофізичного параметра нагрівального елемента з використанням рівняння теплопереносу та динаміки внутрішніх і зовнішніх температур повітря, а також температур на нагрівальних елементах, що дає змогу опосередковано оцінювати їх відносну ефективність та визначати кількісний вклад енергії обігріву обігрівальних елементів.

Результати, отримані у дисертаційній роботі та розроблений прототип апаратно-програмного комплексу інтелектуального будинку можуть бути використані у проектуванні та дослідженні пристроїв для оптимізації функціонування іншого класу інтелектуальних об'єктів як в рамках граничних та туманних обчислень, так і при їх подальшій інтеграції з хмарними сервісами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Exploring the predictability of temperatures in a scaled model of a smarthome / Thomas Burns [та ін.] // Sensors. – 2021. – Т. 21, №18. – С. 6052.
2. Jouini M. Surveying and analyzing security problems in cloud computing environments / Mouna Jouini, Latifa Ben Arfa Rabai // 2014 tenth international conference on computational intelligence and security (CIS), Kunming, 15–16 листоп. 2014 р. – [Б. м.], 2014.
3. Ketu S. Cloud, fog and mist computing in IoT: an indication of emerging opportunities / Shwet Ketu, Pramod Kumar Mishra // IETE technical review. – 2021. – С. 1–12.
4. Tandale U. An empirical study of application layer protocols for IoT [Електронний ресурс] / Upendra Tandale, Bashirahamad Momin, Deva P. Seetharam // 2017 international conference on energy, communication, data analytics and soft computing (ICECDS), Chennai, 1–2 серп. 2017 р. – [Б. м.], 2017.
5. Estimation of smart home thermophysical parameters using dynamic series of temperature and energy data / Oleh Sinkevych [та ін.] // 2019 IEEE 2nd ukraine conference on electrical and computer engineering (UKRCON), Lviv, Ukraine, 2–6 лип. 2019 р. – [Б. м.], 2019.
6. Sinkevych O. Estimation of effective thermal parameters of heating sources based on dynamic measurments in smart home / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Voyko // Вісник Національного університету "Львівська політехніка" "Комп'ютерні системи проектування. Теорія і практика" – 2019 – No. 1(1). – С.–58-66.
7. Пат. № 140565 Україна, МПК G01N 25/20, G01N 27/18, G06F 15/00. Спосіб визначення теплофізичних характеристик архітектурних

- споруд / Монастирський Л. С., Соколовський Б. С., Сінькевич О. О., Оленич І. Б. Заявник і власник – Львівський національний університет імені Івана Франка. – № u201905725; заявл. 27.05.2019; опубл. 10.03.2020 р. Бюл. № 5.
8. Sinkevych O. Statistical Analysis of the Thermal Parameters of Smart Homes / O. Sinkevych, L. Monastyrskii, B. Sokolovskyi // Електроніка та інформаційні технології. – 2018. – Вип. 10. – С. 99–108.
 9. Сінькевич О. Регресійні моделі для аналізу даних інтелектуального будинку / О. Сінькевич // Електроніка та інформаційні технології. – 2022. – Вип. 20. – С. 62–73.
 10. Sinkevych O. Gas Disaggregation Approach Based on Cluster Analysis / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Boyko // Computer Technologies of Printing. – 2019. – Вип.- 41 (1). – С. 23-33.
 11. Sinkevych O. Development of neuro-controller based on STM 32 / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Boyko // Електроніка та інформаційні технології. – 2020. – Вип. 13. – С. 118–125.
 12. Sinkevych O. Low complexity recurrent neural networks for edge computing / O. Sinkevych // Електроніка та інформаційні технології. – 2021. – Вип. 16. – С. 11–19.
 13. Sinkevych O. MLOPS prototype of AI system for edge computing / O. Sinkevych. Ya. Boyko, L. Monastyrskyy // Електроніка та інформаційні технології. – 2022. – Вип. 17. – С. 74–83.
 14. Sinkevych O. Determination of Regression Parameters for the Thermal and Energy Components of Smart Homes / O. Sinkevych, L. Monastyrskyy, B. Sokolovskyi // International Scientific and Practical Conference "Electronics and Information Technologies" (ELIT-2018). A-92 A-95. 2018.
 15. Сінькевич О. Встановлення кореляційних зв'язків між тепловими параметрами розумних будинків / О. Сінькевич, Л. Монастирський, Б. Соколовський // Матеріали Міжнародної наукової конференції

- студентів і молодих науковців з теоретичної та експериментальної фізики “Еврика-2018”. – [Б. м.], 2018. – Н13.
16. Cluster analysis of smart home energy time series / Олег Сінкевич [та ін.] // Матеріали IV Міжнародної науково-технічної конференції „теоретичні та прикладні аспекти радіотехніки, приладобудування і комп’ютерних технологій“ присвячена 80-ти річчю з дня народження професора Я. І. Проця, 2019. – [Б. м.]. – С. 237–240.
17. Sinkevych O. On the development of object detector based on capsule neural networks / Oleh Sinkevych, Daniil Berezhansky, Zenyk Matchyshyn // 2019 XIth international scientific and practical conference on electronics and information technologies (ELIT), Lviv, Ukraine, 16–18 верес. 2019 р. – [Б. м.], 2019.
18. Sinkevych O. Dynamic approach to identification of smart home thermophysical parameters / Oleh Sinkevych // Матеріали Міжнародної наукової конференції студентів і молодих науковців з теоретичної та експериментальної фізики “ЕВРИКА-2019”, 14-16 трав. 2019 р. – [Б. м.], 2019. – П1.
19. Sinkevych O. To the modeling of smart home heating behavior / Oleh Sinkevych // Матеріали Міжнародної наукової конференції студентів і молодих науковців з теоретичної та експериментальної фізики “ЕВРИКА-2020”, 6 жовт. 2020 р. – [Б. м.], 2020. – G7.
20. Algorithm of tuning heating source thermophysical parameters in smart home / Oleh Sinkevych [та ін.] // 2020 IEEE xvith international conference on the perspective technologies and methods in MEMS design (MEMSTECH), Lviv, Ukraine, 22–26 квіт. 2020 р. – [Б. м.], 2020.
21. Embedding sequence model in STM32 based neuro-controller / Oleh Sinkevych [та ін.] // 2021 IEEE 12th International conference on electronics and information technologies (ELIT), Lviv, Ukraine, 19–21 трав. 2021 р. – [Б. м.], 2021.

22. Sinkevych O. Genetic hyper-parameters optimization for STM32 deep learning model / Oleh Sinkevych, Oleksandr Rechynskiy // Матеріали Міжнародної наукової конференції студентів і молодих науковців з теоретичної та експериментальної фізики “ЕВРИКА-2021”, 19 трав. 2021 р. – [Б. м.], 2021. – G10.
23. Roe D. 7 Big Problems With the Internet of Things [Електронний ресурс] / David Roe // CMSWire.com. – Режим доступу: <https://www.cmswire.com/cms/internet-of-things/7-big-problems-with-the-internet-of-things-024571.php>.
24. Global IoT and non-IoT connections 2010-2025 | Statista [Електронний ресурс] // Statista. – Режим доступу: <https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/>.
25. Energy Efficiency Effectiveness of Smart Thermostat Based BEMS [Електронний ресурс] / Koushik Mandlem [та ін.] // Energy Engineering. – 2020. – Т. 117, № 4. – С. 165–183.
26. Kaur K. Swarm Intelligence and its applications towards Various Computing: A Systematic Review / Komalpreet Kaur, Yogesh Kumar // 2020 International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 17–19 черв. 2020 р. – [Б. м.], 2020.
27. Said O. Towards internet of things: Survey and future vision / Omar Said, Masud Mehedi // International Journal of Computer Networks. – 2013. – Т. 5, № 1. – С. 1–17.
28. Shaikh E. Internet of Things (IoT): Security and Privacy Threats / Eman Shaikh, Iman Mohiuddin, Ayisha Manzoor // 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 1–3 трав. 2019 р. – [Б. м.], 2019.

29. Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future / Sandro Nižetić [та ін.] // Journal of Cleaner Production. – 2020. – Т. 274. – С. 122877.
30. A Concise Review on Internet of Things (IoT) -Problems, Challenges and Opportunities / Laith Farhan [та ін.] // 2018 11th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Budapest, 18–20 лип. 2018 р. – [Б. м.], 2018.
31. The Smart Home Concept: our immediate future / Vincent Ricquebourg [та ін.] // 2006 1ST IEEE International Conference on E-Learning in Industrial Electronics, Hammamet, Tunisia, 18–20 груд. 2006 р. – [Б. м.], 2006.
32. Smart Home: Integrating Internet of Things with Web Services and Cloud Computing / Moataz Soliman [та ін.] // 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), Bristol, United Kingdom, 2–5 груд. 2013 р. – [Б. м.], 2013.
33. Teng W.-C. Design of MyServer: A Residential Server in Smart Home Systems / Wei-Chung Teng, Yu-Chun Pao, Sheng-Luen Chung // 2008 IEEE Asia-Pacific Services Computing Conference (APSCC), Yilan, Taiwan, 9–12 груд. 2008 р. – [Б. м.], 2008.
34. Wu J.-S. Smart Home Personalizing Services Based on Adaptive Cloud IoT Devices with Bigdata and Deep Learning Technique/ Jain-Shing Wu, Ming-Shen Jian // 2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), YILAN, Taiwan, 20–22 трав. 2019 р. – [Б. м.], 2019.
35. Smart home: integrating internet of things with web services and cloud computing / Moataz Soliman [та ін.] // 2013 IEEE 5th international conference on cloud computing technology and science (cloudcom), Bristol, United Kingdom, 2–5 груд. 2013 р. – [Б. м.], 2013.

36. Lin Y. Study of smart home system based on cloud computing and the key technologies / Yuan Lin // 2015 international conference on computational intelligence and communication networks (CICN), Jabalpur, 12–14 груд. 2015 р. – [Б. м.], 2015.
37. Marikyan D. A systematic review of the smart home literature: a user perspective / Davit Marikyan, Savvas Papagiannidis, Eleftherios Alamanos // Technological forecasting and social change. – 2019. – Т. 138. – С. 139–154.
38. An IoT based smart thermostat / Levent Ozgur [та ін.] // 2018 5th international conference on electrical and electronic engineering (ICEEE), Istanbul, 3–5 трав. 2018 р. – [Б. м.], 2018.
39. Xiaodong Z. Design and implementation of smart home control system based on STM32 / Zhang Xiaodong, Zhang Jie // 2018 chinese control and decision conference (CCDC), Shenyang, China, 9–11 черв. 2018 р. – [Б. м.], 2018.
40. Keşkişoğlu A. Challenges on smart thermostat systems in intelligent buildings / Ataberk Keşkişoğlu, Cihan Turhan // UEMK 2019 proceedings book, Gaziantep, 25 жовт. 2019 р. – Gaziantep University, 2019. – С. 264–274.
41. Kjørveziroski V. IoT serverless computing at the edge: a systematic mapping review / Vojdan Kjørveziroski, Sonja Filiposka, Vladimir Trajkovic // Computers. – 2021. – Т. 10, № 10. – С. 130.
42. Chithra Devi S. A comparative study on cloud computing and edge computing with its applications / S. Chithra Devi, D. Maheswari // Indian journal of natural sciences. – 2021. – Т. 12, № 66. – С. 32241–32247.
43. Singhal A. K. Cloud computing vs fog computing: a comparative study / Aviral Kumar Singhal, Niraj Singhal // International journal of advanced networking and applications. – 2021. – Т. 12, № 04. – С. 4627–4632.

44. Internet of everything / ред.: В. Di Martino [та ін.]. – Singapore : Springer Singapore, 2018.
45. Edge computing: a systematic mapping study / Huihui Xue [та ін.] // 2021 IEEE intl conf on dependable, autonomic and secure computing, intl conf on pervasive intelligence and computing, intl conf on cloud and big data computing, intl conf on cyber science and technology congress (dasc/picom/cbdcom/cyberscitech), AB, Canada, 25–28 жовт. 2021 р. – [Б. м.], 2021.
46. Chakraborty T. Home automation using edge computing and Internet of Things / Tanmay Chakraborty, Soumya Kanti Datta // 2017 IEEE international symposium on consumer electronics (ISCE), Kuala Lumpur, 14–15 листоп. 2017 р. – [Б. м.], 2017.
47. Piyare R. Internet of things: ubiquitous home control and monitoring system using android based smart phone / Rajeev Piyare // International journal of internet of things. – 2013. – Т. 2, № 1. – С. 5–11.
48. Designing and implementation of home automation system based on remote sensing technique with arduino uno microcontroller / Iman I. M. Abu Sulayman [та ін.] // 2017 9th IEEE-GCC Conference and Exhibition (GCCCE), Манана, 8–11 трав. 2017 р. – [Б. м.], 2017.
49. Pinter J. M. Development of speech-based interface for smart home systems / Judit Maria Pinter, Laszlo Czap, L. Kiss Marton // 2018 19th international carpathian control conference (ICCC), Szilvasvarad, 28–31 трав. 2018 р. – [Б. м.], 2018.
50. Akour M. Mobile voice recognition based for smart home automation control / Mohammed Akour // International journal of advanced trends in computer science and engineering. – 2020. – Т. 9, № 3. – С. 3788–3792.

51. IoT-Inspired framework of intruder detection for smart home security systems / Tariq Ahamed Ahanger [та ін.] // Electronics. – 2020. – Т. 9, № 9. – С. 1361.
52. IoT and machine learning based prediction of smart building indoor temperature / Debayan Paul [та ін.] // 2018 4th international conference on computer and information sciences (ICCOINS), Kuala Lumpur, 13–14 серп. 2018 р. – [Б. м.], 2018.
53. Self-Learning algorithm to predict indoor temperature and cooling demand from smart WiFi thermostat in a residential building / K. Huang та ін. Sustainability. 2020. Т. 12, № 17. С. 7110.
54. Deep reinforcement learning for smart home energy management / Liang Yu [та ін.] // IEEE internet of things journal. – 2020. – Т. 7, № 4. – С. 2751–2762.
55. Traboulsi S. Machine learning models for predicting indoor air temperature of smart building / Salam Traboulsi, Stefan Knauth // Communications in computer and information science. – Cham, 2022. – С. 586–595.
56. Towards smart home automation using IoT-enabled edge-computing paradigm / Hikmat Yar [та ін.] // Sensors. – 2021. – Т. 21, № 14. – С. 4932.
57. Node-RED [Електронний ресурс] // Node-RED. – Режим доступу: <https://nodered.org>.
58. A portable node of humidity and temperature sensor for indoor environment monitoring / Trio Adiono [та ін.] // 2018 3rd international conference on intelligent green building and smart grid (IGBSG), Yi-Lan, 22–25 квіт. 2018 р. – [Б. м.], 2018.
59. Smart campus: an experimental performance comparison of collaborative and cooperative schemes for wireless sensor network / Carolina Del-Valle-Soto [та ін.] // Energies. – 2019. – Т. 12, № 16. – С. 3135.

60. Gavra V.-D. Usage of ZigBee and LoRa wireless technologies in IoT systems / Vlad-Dacian Gavra, Ovidiu Aurel Pop // 2020 IEEE 26th international symposium for design and technology in electronic packaging (SIITME), Pitesti, Romania, 21–24 жовт. 2020 р. – [Б. м.], 2020.
61. Chen C.-H. Edge computing gateway of the industrial internet of things using multiple collaborative microcontrollers / Ching-Han Chen, Ming-Yi Lin, Chung-Chi Liu // IEEE network. – 2018. – Т. 32, № 1. – С. 24–32.
62. ESP32-S3 Wi-Fi & Bluetooth 5 (LE) MCU | Espressif Systems [Електронний ресурс] // Overview | Espressif Systems. – Режим доступу: <https://www.espressif.com/en/products/socs/esp32-s3> (дата звернення: 29.06.2022).
63. Dokic K. Analysis of ESP32 SoC for feed-forward neural network applications / Kristian Dokic, Dubravka Mandusic, Bojan Radisic // Learning and analytics in intelligent systems. – Cham, 2019. – С. 165–175.
64. STM32H745/755 - STMicroelectronics [Електронний ресурс] // STMicroelectronics. – Режим доступу: <https://www.st.com/en/microcontrollers-microprocessors/stm32h745-755.html> (дата звернення: 29.06.2022).
65. Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey / Giang Nguyen [та ін.] // Artificial intelligence review. – 2019. – Т. 52, № 1. – С. 77–124.
66. ONNX | Home [Електронний ресурс] // ONNX | Home. – Режим доступу: <https://onnx.ai/>.
67. Quantization and deployment of deep neural networks on microcontrollers / Pierre-Emmanuel Novac [та ін.] // Sensors. – 2021. – Т. 21, №9. – С. 2984.

68. Learning to quantize deep neural networks: a competitive-collaborative approach / Md Fahim Faysal Khan [та ін.] // 2020 57th ACM/IEEE design automation conference (DAC), San Francisco, CA, USA, 20–24 лип. 2020 р. – [Б. м.], 2020.
69. Mobilenet-SSDv2: an improved object detection model for embedded systems / Yu-Chen Chiu [та ін.] // 2020 international conference on system science and engineering (ICSSE), Kagawa, Japan, 31 серп. – 3 верес. 2020 р. – [Б. м.], 2020.
70. Improving the accuracy of SqueezeNet with negligible extra computational cost / Yi Liu [та ін.] // 2020 international conference on high performance big data and intelligent systems (HPBD&IS), Shenzhen, China, 23 трав. 2020 р. – [Б. м.], 2020.
71. Joshi S. R. HBONext: hbonet with flipped inverted residual / Sanket Ramesh Joshi, Mohamed El-Sharkawy // 2021 IEEE international conference on design & test of integrated micro & nano-systems (DTS), Sfax, Tunisia, 7–10 черв. 2021 р. – [Б. м.], 2021.
72. Knowledge distillation: a survey / Jianping Gou [та ін.] // International journal of computer vision. – 2021. – Т. 129, № 6. – С. 1789–1819.
73. Maragatham T. IoT based home automation system using Raspberry Pi 4 / T. Maragatham, P. Balasubramanie, M. Vivekanandhan // IOP conference series: materials science and engineering. – 2021. – Т. 1055, № 1. – С. 12081.
74. Saleem A. A. Smart homes powered by machine learning: a review / Ayad Abdulrahman Saleem, Masoud Mohammad Hassan, Ismael Ali Ali // 2022 international conference on computer science and software engineering (CSASE), Duhok. – [Б. м.], 2022. – С. 355–361.
75. Li Z. A networked smart home system based on recurrent neural networks and reinforcement learning / Zhongwang Li, Bin Deng //

- Systems science & control engineering. – 2021. – Т. 9, № 1. – С. 775–783.
76. Google cloud IoT - fully managed iot services [Электронный ресурс] // Google Cloud. – Режим доступа: <https://cloud.google.com/solutions/iot#section-7>.
77. Applicability of industrial IoT in diversified sectors: evolution, applications and challenges / Rohan Sethi [та ін.] // Studies in big data. – Singapore, 2020. – С. 45–67.
78. Xiaobei W. Design and implementation of intelligent PID control system of high-efficiency comber / Wang Xiaobei, Wang Haibo // Procedia engineering. – 2011. – Т. 24. – С. 355–359.
79. Gupta A. Edge intelligence makes smart homes truly intelligent - capgemini engineering united states [Электронный ресурс] / Arpna Gupta // Capgemini Engineering United States. – Режим доступа: <https://capgemini-engineering.com/us/en/insight/edge-intelligence-makes-smart-homes-truly-intelligent/>.
80. Теслюк В. М. Опрацювання даних у системах "розумного" будинку з використанням моделей на підставі мереж Петрі / В. М. Теслюк, А. Г. Казарян, І. Я. Казимира // Scientific Bulletin of UNFU. – 2021. – Т. 31, № 1. – С. 131–136.
81. An overview of IoT sensor data processing, fusion, and analysis techniques / Rajalakshmi Krishnamurthi [та ін.] // Sensors. – 2020. – Т. 20, № 21. – С. 6076.
82. Wavelet-Based filtration procedure for denoising the predicted CO2 waveforms in smart home within the internet of things / Jan Vanus [та ін.] // Sensors. – 2020. – Т. 20, № 3. – С. 620.
83. Missing value imputation based on gaussian mixture model for the internet of things / Xiaobo Yan [та ін.] // Mathematical problems in engineering. – 2015. – Т. 2015. – С. 1–8.

84. A missing sensor data estimation algorithm based on temporal and spatial correlation / Zhipeng Gao [та ін.] // International journal of distributed sensor networks. – 2015. – Т. 2015. – С. 1–10.
85. Mary I. P. S. Imputing the missing data in IoT based on the spatial and temporal correlation / I. Priya Stella Mary, L. Arockiam // 2017 IEEE international conference on current trends in advanced computing (ICCTAC), Bangalore, 2–3 берез. 2017 р. – [Б. м.], 2017.
86. Lin W.-C. Deep learning for missing value imputation of continuous data and the effect of data discretization / Wei-Chao Lin, Chih-Fong Tsai, Jia Rong Zhong // Knowledge-Based systems. – 2022. – Т. 239. – С. 108079.
87. Filling missing data using interpolation methods: study on the effect of fitting distribution / M. N. Noor [та ін.] // Key engineering materials. – 2013. – Т. 594-595. – С. 889–895.
88. Liu F. T. Isolation forest / Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou // 2008 eighth IEEE international conference on data mining (ICDM), Pisa, Italy, 15–19 груд. 2008 р. – [Б. м.], 2008.
89. Anomaly detection in time series via robust PCA / Yongjun Jin [та ін.] // 2017 2nd IEEE international conference on intelligent transportation engineering (ICITE), Singapore, Singapore, 1–3 верес. 2017 р. – [Б. м.], 2017.
90. Tuli S. TranAD / Shreshth Tuli, Giuliano Casale, Nicholas R. Jennings // Proceedings of the VLDB Endowment. – 2022. – Т. 15, № 6. – С. 1201–1214.
91. Hyndman R. 3.6 STL decomposition | Forecasting: Principles and Practice (3rd ed) [Електронний ресурс] / Rob Hyndman, George Athanasopoulos // OTexts. – Режим доступу: <https://otexts.com/fpp3/stl.html>
92. Hanbanchong A. SARIMA based network bandwidth anomaly detection / Aphichit Hanbanchong, Kerk Piromsopa // 2012 international joint

- conference on computer science and software engineering (JCSSE), Bangkok, Thailand, 30 трав. – 1 черв. 2012 р. – [Б. м.], 2012.
93. Machine learning for anomaly detection: a systematic review // IEEE access. – 2021. – С. 1-43.
94. A review on deep learning techniques for iot data / Kuruva Lakshmana [та ін.] // Electronics. – 2022. – Т. 11, № 10. – С. 1604.
95. Oleh Sinkevych, Liubomyr Monastyrskiy, Bohdan Sokolovskiy. Determination of regression parameters for the thermal and energy components of smart homes / International Scientific and Practical Conference "Electronics and Information Technologies" (ELIT-2018). A-92 A-95. 2018.
96. Heating behaviour in English homes: an assessment of indirect calculation methods / T. Kane [та ін.] // Energy and buildings. – 2017. – Т. 148. – С. 89–105.
97. Linear regression analysis of energy consumption data for smart homes / Pheny Phemelo Moletsane [та ін.] // 2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO), Opatija, 21–25 трав. 2018 р. – [Б. м.], 2018.
98. Energy usage prediction for smart home with regression based ensemble model / Mohammad Shamsul Hoque [та ін.] // 2020 8th international conference on information technology and multimedia (ICIMU), Selangor, Malaysia, 24–26 серп. 2020 р. – [Б. м.], 2020.
99. Shorfuzzaman M. Predictive analytics of energy usage by iot-based smart home appliances for green urban development / Mohammad Shorfuzzaman, M. Shamim Hossain // ACM transactions on internet technology. – 2022. – Т. 22, № 2. – С. 1–26.
100. Spencer B. Forecasting temperature in a smart home with segmented linear regression / Bruce Spencer, Omar Alfandi, Feras Al-Obeidat // Procedia computer science. – 2019. – Т. 155. – С. 511–518.

101. Ma X. Prediction of outdoor air temperature and humidity using Xgboost / Xiaoming Ma, Cong Fang, Junping Ji // IOP conference series: earth and environmental science. – 2020. – Т. 427. – С. 012013.
102. A regression-based framework to examine thermal loads of buildings / Mohammad K. Najjar [та ін.] // Journal of cleaner production. – 2021. – Т. 292. – С. 126021.
103. Armel C. Energy disaggregation [Электронний ресурс] / Carrie Armel // <https://web.stanford.edu>. – Режим доступу: <https://web.stanford.edu/group/peec/cgi-bin/docs/events/2011/becc/presentations/3%20Disaggregation%20The%20Holy%20Grail%20-%20Carrie%20Armel.pdf>.
104. Hart G. W. Residential energy monitoring and computerized surveillance via utility power flows / G. W. Hart // IEEE technology and society magazine. – 1989. – Т. 8, № 2. – С. 12–16.
105. Unsupervised disaggregation of low frequency power measurements / Hyungsul Kim [та ін.] // Proceedings of the 2011 SIAM international conference on data mining. – Philadelphia, PA, 2011.
106. Non-Intrusive load monitoring using prior models of general appliance types / Oliver Parson [та ін.] // Proceedings of the AAAI conference on artificial intelligence. – 2012. – Т. 26, № 1. – С. 356–362.
107. Non-intrusive appliance load monitoring using low-resolution smart meter data / Jing Liao [та ін.] // 2014 IEEE international conference on smart grid communications, Venice, Italy, 3–6 листоп. 2014 р. – [Б. м.], 2014.
108. Deep learning-based energy disaggregation and on/off detection of household appliances / Jie Jiang [та ін.] // ACM transactions on knowledge discovery from data. – 2021. – Т. 15, № 3. – С. 1–21.
109. Humala B. UniversalNILM / Bontor Humala, Akshay S. N. Uttama Nambi, Venkatesha R. Prasad // E-Energy '18: the ninth international

- conference on future energy systems, Karlsruhe Germany. – New York, NY, USA, 2018.
110. Disaggregating high-resolution gas metering data using pattern recognition / Ammar Alzaatreh [та ін.] // Energy and buildings. – 2018. – Т. 176. – С. 17–32.
111. Breschi V. Kalman filtering for energy disaggregation / V. Breschi, D. Piga, A. Vemporad // IFAC-PapersOnLine. – 2018. – Т. 51, № 5. – С. 108–113.
112. Pandas - python data analysis library [Електронний ресурс] // pandas - Python Data Analysis Library. – Режим доступу: <https://pandas.pydata.org/>.
113. Dask – Dask documentation [Електронний ресурс] // Dask – Dask documentation. – Режим доступу: <https://docs.dask.org/en/stable/>.
114. How to reshape input data for long short-term memory networks in keras [Електронний ресурс] // Machine Learning Mastery. – Режим доступу: <https://machinelearningmastery.com/reshape-input-data-long-short-term-memory-networks-keras/>.
115. ColumnML / Каан Кара [та ін.] // Proceedings of the VLDB Endowment. – 2018. – Т. 12, № 4. – С. 348–361.
116. Учасники проєктів Вікімедіа. Злиття даних – Вікіпедія [Електронний ресурс] / Учасники проєктів Вікімедіа // Вікіпедія. – Режим доступу: https://uk.wikipedia.org/wiki/Злиття_даних.
117. A survey of data fusion in smart city applications / Billy Pik Lik Lau [та ін.] // Information fusion. – 2019. – Т. 52. – С. 357–374.
118. Design and implementation of a smart home system using multisensor data fusion technology / Yu-Liang Hsu [та ін.] // Sensors. – 2017. – Т. 17, № 7. – С. 1631.
119. Multi-Sensor data fusion for smart home reliable pedestrian localization / Lina Achaji [та ін.] // 2021 IEEE 3rd international multidisciplinary

- conference on engineering technology (IMCET), Beirut, Lebanon, 8–10 груд. 2021 р. – [Б. м.], 2021.
120. Towards modified entropy mutual information feature selection to forecast medium-term load using a deep learning model in smart homes / Omaji Samuel [та ін.] // Entropy. – 2020. – Т. 22, № 1. – С. 68. –
121. Müller I. M. Feature selection for energy system modeling: identification of relevant time series information / Inga M. Müller // Energy and AI. – 2021. – Т. 4. – С. 100057.
122. A feature selection-based predictive-learning framework for optimal actuator control in smart homes / Sehrish Malik [та ін.] // Actuators. – 2021. – Т. 10, № 4. – С. 84.
123. Zenger A. Towards the intelligent home: using reinforcement-learning for optimal heating control / Alexander Zenger, Jochen Schmidt, Michael Krödel // KI 2013: advances in artificial intelligence. – Berlin, Heidelberg, 2013. – С. 304–307.
124. Nandi C. Automatic trigger generation for rule-based smart homes / Chandrakana Nandi, Michael D. Ernst // CCS'16: 2016 ACM SIGSAC conference on computer and communications security, Vienna Austria. – New York, NY, USA, 2016.
125. Indoor temperature prediction in an IoT scenario / Pedro Monteiro [та ін.] // Sensors. – 2018. – Т. 18, № 11. – С. 3610.
126. Temperature and load consumption forecast in smart building on foundation iot by ARIMA algorithm / Phi Chi Do [та ін.] // Journal of science and technology: issue on information and communications technology. – 2019. – Т. 17, № 12.2. – С. 55.
127. Siami-Namini S. A comparison of ARIMA and LSTM in forecasting time series / Sima Siami-Namini, Neda Tavakoli, Akbar Siami Namin // 2018 17th IEEE international conference on machine learning and applications (ICMLA), Orlando, FL, 17–20 груд. 2018 р. – [Б. м.], 2018.

128. Comparison of time series forecasting based on statistical ARIMA model and LSTM with attention mechanism / Kun Zhou [та ін.] // Journal of physics: conference series. – 2020. – Т. 1631. – С. 012141.
129. Banerjee S. A comparative study of seasonal-arima and RNN (LSTM) on time series temperature data forecasting / Sumanta Banerjee, Shyamapada Mukherjee // Pervasive computing and social networking. – Singapore, 2022. – С. 315–326.
130. Elsaraiti M. A comparative analysis of the ARIMA and LSTM predictive models and their effectiveness for predicting wind speed / Meftah Elsaraiti, Adel Merabet // Energies. – 2021. – Т. 14, № 20. – С. 6782.
131. Preeti. Financial and non-stationary time series forecasting using LSTM recurrent neural network for short and long horizon / Preeti, Rajni Bala, Ram Pal Singh // 2019 10th international conference on computing, communication and networking technologies (ICCCNT), Kanpur, India, 6–8 лип. 2019 р. – [Б. м.], 2019.
132. Hippert H. S. Combining neural networks and ARIMA models for hourly temperature forecast / H. S. Hippert, C. E. Pedreira, R. C. Souza // Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks. IJCNN 2000. neural computing: new challenges and perspectives for the new millennium, Como, Italy, 27 лип. 2000 р. – [Б. м.], 2000.
133. Comparison of ARIMA and LSTM for prediction of hemorrhagic fever at different time scales in China / Rui Zhang [та ін.] // Plos one. – 2022. – Т. 17, № 1. – С. 262.
134. Medium-term forecasting of cold, electric and gas load in multi-energy system based on VAR model / Li Yujie [та ін.] // 2018 13th IEEE conference on industrial electronics and applications (ICIEA), Wuhan, 31 трав. – 2 черв. 2018 р. – [Б. м.], 2018.

135. LSTM-based indoor air temperature prediction framework for HVAC systems in smart buildings / Fatma Mtibaa [та ін.] // Neural computing and applications. – 2020. – Т. 32, № 23. – С. 17569–17585.
136. Machine learning on mainstream microcontrollers / Fouad Sakr [та ін.] // Sensors. – 2020. – Т. 20, № 9. – С. 2638.
137. Artificial intelligence ecosystem for STM32 - STMicroelectronics [Электронний ресурс] // STMicroelectronics. – Режим доступу: https://www.st.com/content/st_com/en/ecosystems/artificial-intelligence-ecosystem-stm32.html.
138. TinyML platforms benchmarking / Anas Osman [та ін.] // Lecture notes in electrical engineering. – Cham, 2022. – С. 139–148.
139. AI:X-CUBE-AI documentation - stm32mcu [Электронний ресурс] // ST Wiki - STMicroelectronics. – Режим доступу: https://wiki.st.com/stm32mcu/wiki/AI:X-CUBE-AI_documentation
140. Mirza F. How to run neural network on STM32 [Электронний ресурс] / Fahad Mirza // Fahad Mirza - Fahad Mirza - Embedded Engineer. – Режим доступу: <https://mirzafahad.github.io/2020-06-16-tflite-stm32/>.
141. Jetson Nano Developer Kit [Электронний ресурс] // NVIDIA Developer. – Режим доступу: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>.
142. Performance evaluation of the nvidia jetson nano through a real-time machine learning application / Sebastián Valladares [та ін.] // Advances in intelligent systems and computing. – Cham, 2021. – С. 343–349.
143. X-CUBE-AI - STMicroelectronics [Электронний ресурс] // STMicroelectronics. – Режим доступу: <https://www.st.com/en/embedded-software/x-cube-ai.html>.

144. What is MLOps? - Databricks [Электронный ресурс] // Databricks. – Режим доступа: <https://www.databricks.com/glossary/mlops>.
145. Why Should You Use MLOps? - Amazon SageMaker [Электронный ресурс]. – Режим доступа: <https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-projects-why.html>
146. The best IoT databases for the edge – an overview and guide - AZ Big Media [Электронный ресурс] // AZ Big Media. – Режим доступа: <https://azbigmedia.com/business/the-best-iot-databases-for-the-edge-an-overview-and-guide/>.
147. Feature store for ML [Электронный ресурс] // Feature Store For ML. – Режим доступа: <https://www.featurestore.org/>.
148. Developments in MLflow / Andrew Chen [та ин.] // SIGMOD/PODS '20: international conference on management of data, Portland OR USA. – New York, NY, USA, 2020.
149. Ramírez S. FastAPI [Электронный ресурс] / Sebastián Ramírez // FastAPI. – Режим доступа: <https://fastapi.tiangolo.com/>.
150. Gift N. Practical MLOps / Noah Gift, Alfredo Deza. – [Б. м.] : O'Reilly Media, Inc., 2021. – 458 с.
151. Developing suitable thermal models for domestic buildings with Smart Home equipment / Vanda Dimitriou [та ин.] // Proceedings of the 2014 building simulation and optimization conference, London, 23 черв. 2014 р. – [Б. м.], 2014.
152. Himpe E. Characterisation of the thermal performance of a test house based on dynamic measurements / Eline Himpe, Arnold Janssens // Energy procedia. – 2015. – Т. 78. – С. 3294–3299.
153. A data analysis technique to estimate the thermal characteristics of a house / Seyed Tabatabaei [та ин.] // Energies. – 2017. – Т. 10, № 9. – С. 1358.

154. REFIT – REFIT: smart homes and energy demand reduction [Электронный ресурс] // REFIT – REFIT: Smart Homes and Energy Demand Reduction. – Режим доступа: <https://www.refitsmarthomes.org/>.
155. Hargreaves T. Learning to live in a smart home / Tom Hargreaves, Charlie Wilson, Richard Hauxwell-Baldwin // Building research & information. – 2017. – Т. 46, № 1. – С. 127–139.
156. García S. Data preprocessing in data mining / Salvador García, Julián Luengo, Francisco Herrera. – [Б. м.] : Springer International, 2015. – 320 с.
157. Heydt M. Learning pandas - Second Edition: High performance data manipulation and analysis using Python / Michael Heydt. – [Б. м.] : Packt Publishing - ebooks Account, 2017. – 446 с.
158. Droke C. Moving averages simplified / Clif Droke. – [Б. м.] : Marketplace Books, 2001. – 129 с.
159. Hyndman R. Forecasting: principles and practice [Электронный ресурс] / Rob Hyndman, George Athanasopoulos. – Режим доступа: <https://otexts.com/fpp3/>.
160. Liu F. T. Isolation forest / Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou // 2008 eighth IEEE international conference on data mining (ICDM), Pisa, Italy, 15–19 груд. 2008 р. – [Б. м.], 2008.
161. Pena E. H. M. Anomaly detection using forecasting methods ARIMA and HWDS [Электронный ресурс] / Eduardo H. M. Pena, Marcos V. O. de Assis, Mario Lemes Proenca // 2013 32nd international conference of the chilean computer science society (SCCC), Temuco, Cautin, Chile, 11–15 листоп. 2013 р. – [Б. м.], 2013.
162. Agrawal S. Survey on anomaly detection using data mining techniques [Электронный ресурс] / Shikha Agrawal, Jitendra Agrawal // Procedia computer science. – 2015. – Т. 60. – С. 708–713.

163. Zhou C. Anomaly detection with robust deep autoencoders [Электронный ресурс] / Chong Zhou, Randy C. Paffenroth // KDD '17: the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, Halifax NS Canada. – New York, NY, USA, 2017.
164. Bonaccorso G. Machine Learning Algorithms: a reference guide to popular algorithms for data science and machine learning / Giuseppe Bonaccorso. – [Б. м.] : Packt Publishing, 2017. – 360 с.
165. Dangeti P. Statistics for Machine Learning: Techniques for exploring supervised, unsupervised, and reinforcement learning models with Python and R / Pratap Dangeti. – [Б. м.] : Packt Publishing, 2017. – 442 с.
166. The basics of financial econometrics: tools, concepts, and asset management applications / Frank J. Fabozzi [та ин.]. – [Б. м.] : Wiley, 2014. – 448 с.
167. Chen T. XGBoost / Tianqi Chen, Carlos Guestrin // KDD '16: the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco California USA. – New York, NY, USA, 2016.
168. XGBoost-Based algorithm interpretation and application on post-fault transient stability status prediction of power system [Электронный ресурс] / Minghua Chen [та ин.] // IEEE access. – 2019. – Т. 7. – С. 13149–13158.
169. XGBoost Documentation – xgboost 1.6.1 documentation [Электронный ресурс] // XGBoost Documentation – xgboost 1.6.1 documentation. – Режим доступа: <https://xgboost.readthedocs.io/en/stable/>.
170. AutoML: Automatic Machine Learning – H2O 3.36.1.2 documentation. H2O.ai Documentation. URL: <https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>.

- 171.10.7 - Detecting Multicollinearity Using Variance Inflation Factors | STAT 462 [Электронный ресурс] // Statistics Online | STAT ONLINE. – Режим доступа: <https://online.stat.psu.edu/stat462/node/180/>.
172. Handbook of Cluster Analysis [Электронный ресурс] / ред.: С. Hennig [та ін.]. – [Б. м.] : Chapman and Hall/CRC, 2015.
173. Aggarwal C. C. Data Clustering: Algorithms and Applications / Charu C. Aggarwal, Chandan K. Reddy. – [Б. м.] : Taylor & Francis Group, 2018. – 652 с.
174. Bawdekar A. A. Selection of stationarity tests for time series forecasting using reliability analysis / Advait Amol Bawdekar, B. Rajanarayan Prusty // Mathematical problems in engineering. – 2022. – Т. 2022. – С. 1–8.
175. Nikbakht S. Optimizing the neural network hyperparameters utilizing genetic algorithm / Saeid Nikbakht, Cosmin Anitescu, Timon Rabczuk // Journal of Zhejiang University-SCIENCE A. – 2021. – Т. 22, № 6. – С. 407–426.
- 176.10.4. Bidirectional Recurrent Neural Networks – Dive into Deep Learning 1.0.0-alpha1.post0 documentation [Электронный ресурс] // Dive into Deep Learning – Dive into Deep Learning 1.0.0-alpha1.post0 documentation. – Режим доступа: https://d2l.ai/chapter_recurrent-modern/bi-rnn.html.
177. Hochreiter S. Long Short-Term Memory [Электронный ресурс] / Sepp Hochreiter, Jürgen Schmidhuber // Neural computation. – 1997. – Т. 9, № 8. – С. 1735–1780.
178. Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network / Alex Sherstinsky // Physica D: nonlinear phenomena. – 2020. – Т. 404. – С. 132306.
179. On the properties of neural machine translation: encoder–decoder approaches / Kyunghyun Cho [та ін.] // Proceedings of SSST-8, eighth

- workshop on syntax, semantics and structure in statistical translation, Doha, Qatar. – Stroudsburg, PA, USA, 2014.
180. Belete D. M. Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results / Daniel Mesafint Belete, Manjaiah D. Huchaiiah // International journal of computers and applications. – 2021. – С. 1–12.
181. Willkoehrsen. Intro to model tuning: grid and random search [Электронный ресурс] / willkoehrsen // Kaggle: Your Machine Learning and Data Science Community. – Режим доступа: <https://www.kaggle.com/code/willkoehrsen/intro-to-model-tuning-grid-and-random-search/notebook>.
182. Yang L. On hyperparameter optimization of machine learning algorithms: theory and practice / Li Yang, Abdallah Shami // Neurocomputing. – 2020. – Т. 415. – С. 295–316.
183. Sivanandam S. N. Introduction to genetic algorithms / S. N. Sivanandam. – Berlin : Springer, 2007. – 442 с.
184. DEAP documentation – DEAP 1.3.3 documentation [Электронный ресурс] // DEAP documentation – DEAP 1.3.3 documentation. – Режим доступа: <https://deap.readthedocs.io/en/master/>.
185. Online learning: a comprehensive survey / Steven C. H. Hoi [та ин.] // Neurocomputing. – 2021. – Т. 459. – С. 249–289.
186. TechEmpower framework benchmarks [Электронный ресурс] // TechEmpower. – Режим доступа: <https://www.techempower.com/benchmarks/#section=test&runid=7464e520-0dc2-473d-bd34-dbd7e85911&hw=ph&test=query&l=zijzen-7>.
187. What is airflow? – airflow documentation [Электронный ресурс] // Apache Airflow. – Режим доступа: <https://airflow.apache.org/docs/apache-airflow/stable/index.html>.

188. Asyncio Asynchronous I/O – Python 3.8.14 documentation [Электронный ресурс] // 3.11.1 Documentation. – Режим доступа: <https://docs.python.org/3.8/library/asyncio.html>.
189. GitHub - f0uriest/keras2c: A simple library to deploy Keras neural networks in pure C for realtime applications [Электронный ресурс] // GitHub. – Режим доступа: <https://github.com/f0uriest/keras2c>.
190. Finite difference methods in heat transfer, second edition. – [Б. м.] : CRC Press, 2017. – Режим доступа: <https://doi.org/10.1201/9781315121475>.
191. Battiti R. BFGS optimization for faster and automated supervised learning / Roberto Battiti, Francesco Masulli // International neural network conference. – Dordrecht, 1990. – С. 757–760. – Режим доступа: https://doi.org/10.1007/978-94-009-0643-3_68.
192. Yang X.-S. Particle swarm optimization [Электронный ресурс] / Xin-She Yang // Nature-Inspired optimization algorithms. – [Б. м.], 2021. – С. 111–121. – Режим доступа: <https://doi.org/10.1016/b978-0-12-821986-7.00015-9>.

Додаток А

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. публікації у фахових виданнях України:

8. Sinkevych O. Statistical Analysis of the Thermal Parameters of Smart Homes / O. Sinkevych, L. Monastyrskii, B. Sokolovskyi // Electronics and Information Technologies. – 2018. – No. 10. – P. 99–108.
9. Sinkevych O. Gas Disaggregation Approach Based on Cluster Analysis / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Boyko // Computer Technologies of Printing. – 2019. – Vol.- 41 (1). – P. 23-33.
10. Sinkevych O. Estimation of effective thermal parameters of heating sources based on dynamic measurements in smart home / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Boyko // Bulletin of the National University "Lviv Polytechnic". Series: "Computer systems of design. Theory and practice – 2020 – No. 1(1). – P.–58-66.
11. Sinkevych O. Development of neuro-controller based on STM 32 / O. Sinkevych, L. Monastyrskii, B. Sokolovskii, Ya. Boyko // Electronics and Information Technologies. – 2020. – No. 13. – P. 118–125.
12. Sinkevych O. Low complexity recurrent neural networks for edge computing / O. Sinkevych // Electronics and Information Technologies. – 2021. – No. 16. – P. 11–19.
13. Sinkevych O. MLOPS prototype of AI system for edge computing / O. Sinkevych. Ya. Boyko, L. Monastyrskyy // Electronics and Information Technologies. – 2022. – No. 17. – С. 74–83.
14. Сінькевич О. Регресійні моделі для аналізу даних інтелектуального будинку / О. Сінькевич // Електроніка та інформаційні технології. – 2022. – Вип. 20. – С. 62–73.

2. публікації у виданнях, які проіндексовані у міжнародній наукометричній базі Scopus:

1. Estimation of smart home thermophysical parameters using dynamic series of temperature and energy data / Oleh Sinkevych [et al.] // 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), Lviv, Ukraine, 2–6 July 2019. P. 934-937.
2. Algorithm of tuning heating source thermophysical parameters in smart home / Oleh Sinkevych [et al.] // 2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, Ukraine, 22–26 April 2020. P. 9-12.
3. Embedding sequence model in STM32 based neuro-controller / Oleh Sinkevych [et al.] // 2021 IEEE 12th International Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 19–21 March 2021. P. 113-118.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Sinkevych O. Determination of Regression Parameters for the Thermal and Energy Components of Smart Homes / O. Sinkevych, L. Monastyrskiy, B. Sokolovskiy // International Scientific and Practical Conference "Electronics and Information Technologies" (ELIT-2018). A-92 A-95. 2018.
2. Сінькевич О. Встановлення кореляційних зв'язків між тепловими параметрами розумних будинків / О. Сінькевич, Л. Монастирський, Б. Соколовський // Матеріали Міжнародної наукової конференції студентів і молодих науковців з теоретичної та експериментальної фізики "Еврика-2018". – Н13.
3. Cluster analysis of smart home energy time series / Олег Сінькевич [та ін.] // Матеріали IV Міжнародної науково-технічної конференції "теоретичні та прикладні аспекти радіотехніки, приладобудування і комп'ютерних технологій" присвячена 80-ти річчю з дня народження професора Я. І. Проця, 2019. – С. 237–240.

4. Sinkevych O. Dynamic approach to identification of smart home thermophysical parameters / Oleh Sinkevych // International Conference of Students and Young Researchers in Theoretical and Experimental Physics “HEUREKA-2019”, 14-16 May 2019. – II.
5. Sinkevych O. To the modeling of smart home heating behavior / Oleh Sinkevych // International Conference of Students and Young Researchers in Theoretical and Experimental Physics “HEUREKA-2020”, 6-7 October. 2020. – G7.
6. Sinkevych O. Genetic hyper-parameters optimization for STM32 deep learning model / Oleh Sinkevych, Oleksandr Rechynskyi // International Conference of Students and Young Researchers in Theoretical and Experimental Physics “HEUREKA-2021”, 18-20 May 2021. – G10.
7. Пат. № 140565 Україна, МПК G01N 25/20, G01N 27/18, G06F 15/00. Спосіб визначення теплофізичних характеристик архітектурних споруд / Монастирський Л. С., Соколовський Б. С., Сінкевич О. О., Оленич І. Б. Заявник і власник – Львівський національний університет імені Івана Франка. – № u201905725; заявл. 27.05.2019; опубл. 10.03.2020 р. Бюл. № 5.

Відомості про апробацію результатів дисертації:

1. X Міжнародна науково-практична конференція "Електроніка та інформаційні технології" (ЕЛІТ-2018), Львів, с. Карпати (Закарпатська обл.), 30 серпня - 2 вересня 2018 року, форма участі – очна, усна доповідь.
2. Міжнародна наукова конференція студентів і молодих науковців з теоретичної та експериментальної фізики “Еврика-2018”, м. Львів, 15-17 травня 2018 року, форма участі – очна, усна доповідь.
3. IV Міжнародна науково-технічна конференція “Теоретичні та прикладні аспекти радіотехніки, приладобудування і комп’ютерних технологій” присвячена 80-ти річчю з дня народження професора Я.

- I. Проця, м. Тернопіль, 20-21 червня 2019 року, форма участі – заочна.
4. Міжнародна наукова конференція студентів і молодих науковців з теоретичної та експериментальної фізики “Еврика-2019”, м. Львів, 15-17 травня 2019 року, форма участі – очна, усна доповідь.
 5. IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), м. Львів, 2–6 липня 2019 року, форма участі – очна, усна доповідь.
 6. 2020 IEEE XVIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), м. Львів, 22–26 квітня 2020 року, форма участі – дистанційна, усна доповідь.
 7. Міжнародна наукова конференція студентів і молодих науковців з теоретичної та експериментальної фізики “Еврика-2020”, м. Львів, 6-7 жовтня 2020 року, форма участі – дистанційна, усна доповідь.
 8. 2021 IEEE 12th International Conference on Electronics and Information Technologies (ELIT), м. Львів, 19-21 березня 2021 року, форма участі – дистанційна, усна доповідь.
 9. Міжнародна наукова конференція студентів і молодих науковців з теоретичної та експериментальної фізики “Еврика-2021”, м. Львів, 18-20 травня 2021 року, форма участі – дистанційна, усна доповідь.
 10. Звітні наукові конференції Львівського національного університету імені Івана Франка за 2018-2022 роки, форми участі – очна/дистанційна, усні доповіді.