

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

«На правах рукопису»

УДК 519.688:004.855.5

«До захисту допущено»

Завідувач кафедри

_____ Олег ЧЕРТОВ

«___» _____ 2023 р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-науковою програмою «Наука про дані та математичне
моделювання»
зі спеціальності 113 «Прикладна математика»
на тему «Математичне та програмне забезпечення системи купівлі / продажу
електроенергії в рамках низьковольтних енергомереж»

Виконав: студент II курсу, групи КМ-11мн

Бірук Станіслав Володимирович

Керівник:

завідувач кафедри ПМА, д-р. техн. наук, професор,

Чертів Олег Романович

Консультант з нормоконтролю:

старший викладач,

Мальчиков Володимир Вікторович

Рецензент:

д-р техн. наук, професор

Терейковський Ігор Анатолійович

Засвідчую, що в цій магістерській
дисертації немає запозичень із праць інших
авторів без відповідних посилань.

Студент _____

Київ — 2023

Національний технічний університет України

«Київський політехнічний інститут

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти — другий (магістерський)

Спеціальність – 113 «Прикладна математика»

Освітньо-наукова програма «Наука про дані та математичне моделювання»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олег ЧЕРТОВ

«___» _____ 2023 р.

ЗАВДАННЯ

на магістерську дисертацію студента

Бірука Станіслава Володимировича

1. Тема дисертації: «Математичне та програмне забезпечення системи купівлі / продажу електроенергії в рамках низьковольтних енергомереж», науковий керівник дисертації Чертов Олег Романович, завідувач кафедри ПМА, д-р. техн. наук, професор, затверджені наказом по університету від «30» березня 2023 р. № 1359-С.

2. Термін подання студентом дисертації: «15» травня 2023 р.

3. Об'єкт дослідження: гібридна енергетична система, низьковольтна енергомережа та її складові, системи керування мікромережами: HOMER, HYBRID 2 та RETScreen. Поняття арбітражу на ринку електроенергії, процеси виробництва, перетворення та зберігання енергії в межах гібридної системи.

Підходи та методи управління нелінійними та складними системами: навчання з підкріпленням; нечітке логічне управління; динамічне програмування та управління з передбаченням поведінки моделі.

4. Предмет дослідження: математичне та програмне забезпечення системи купівлі / продажу електроенергії, в основу яких покладені моделі та алгоритми навчання з підкріпленням, що здійснюють арбітраж виробництва та споживання електроенергії.

5. Для розв'язання поставленої задачі необхідно:

- виконати ґрунтовний аналіз та огляд існуючих рішень систем прийняття рішень в рамках низьковольтних енергомереж;
- в результаті дослідження надати концептуальну модель системи, що здійснює купівлю / продаж електроенергії;
- описати математичне забезпечення системи та розробити програмне забезпечення для верифікації та валідації застосування системи в реальному середовищі.

6. Орієнтовний перелік ілюстративного матеріалу: актуальність, постановка задачі, завдання, огляд теоретичних та практичних праць, пов'язаних із керуванням енергомереж, представлення компонентної діаграми системи, програмне забезпечення, аналіз рішення, висновки, подальша робота.

7. Орієнтовний перелік публікацій:

Тези за темою: «Математичне і програмне забезпечення системи купівлі / продажу електроенергії потужностей відновлювальної енергетики».

8. Дата видачі завдання: «2» вересня 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Ґрунтовне ознайомлення з предметною областю	20.09.2022	
2	Визначення структури магістерської дисертації	31.09.2022	
3	Вивчення літератури	07.10.2022	
4	Пошук даних для проведення експериментальних досліджень	25.10.2022	
5	Проведення наукового дослідження; робота над першим і другим розділом магістерської дисертації	16.02.2023	
6	Розробка та тестування системи	14.04.2023	

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
7	Завершення роботи над основною частиною магістерської дисертації	01.05.2023	
8	Оформлення текстової і графічної частин магістерської дисертації	12.05.2023	

Студент

Станіслав БІРУК

Науковий керівник дисертації

Олег ЧЕРТОВ

Реферат

Дисертацію виконано на 84 аркушах, вона містить 2 додатки та перелік посилань на використані джерела з 47 найменувань. У роботі наведено 32 рис. та 8 таблиць.

Актуальність теми полягає в тому, що впровадження потужностей відновлюваної енергетики в повсякденне життя людини дозволить розподілити навантаження на централізовану мережу передачі електроенергії та зменшити частку використання електроенергії від теплових електростанцій, які негативно впливають на навколишнє середовище.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась згідно з планом науково-дослідних робіт кафедри прикладної математики Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Метою дослідження є оптимізація процесу купівлі / продажу електроенергії з метою зменшення періоду повернення інвестицій, які були залучені при встановленні технічних засобів для генерації та збереження відновлюваної енергії.

Для досягнення вказаної мети було розв'язано такі задачі:

- розглянуто існуючі рішення управління нелінійними та складними системами, зокрема, навчання з підкріпленням, управління з передбаченням поведінки моделі;
- розроблено компонентне та динамічне представлення низьковольтної енергомережі;
- розроблено програмне забезпечення для мультиагентної системи, що здійснює купівлю / продаж електроенергії.

Об'єктом дослідження є гібридна енергетична система, низьковольтна енергомережа та її складові, системи керування мікромережами: HOMER, HYBRID2 та RETScreen. Поняття арбітражу на ринку електроенергії, процеси виробництва, перетворення та зберігання енергії в межах гібридної системи.

Підходи та методи управління нелінійними та складними системами: навчання з підкріпленням; нечітке логічне управління та управління з передбаченням поведінки моделі.

Предмет дослідження: математичне та програмне забезпечення системи купівлі / продажу електроенергії, в основу яких покладені моделі та алгоритми навчання з підкріпленням, що здійснюють арбітраж виробництва та споживання електроенергії.

Методи дослідження. Для розв’язання поставленої задачі використовувалися такі методи: методи машинного навчання, навчання з підкріпленням, управління з передбаченням поведінки моделі, моделювання, проектування систем Data Science, методи теорії алгоритмів та програмування, методи аналізу даних та математичної статистики.

Наукова новизна одержаних результатів роботи. Вперше в рамках єдиної мультиагентної системи прийняття рішень з купівлі/продажу електроенергії були поєднані симулювання роботи накопичувача електроенергії через модель Дойл-Фуллер-Ньюмана і обробка реальних даних про сонячне випромінювання, метеорологічні показники, споживання електроенергії та її вартість, що дозволяє приймати обґрунтовані рішення з планування роботи мікромережі для зменшення рахунків за електроенергію.

Практичне значення одержаних результатів. Завдяки запропонованому математичному і програмному представленні системи купівлі / продажу електроенергії можна зменшити рахунок за електроенергію. Додатково з’являється можливість дослідити вплив постійного розрядження та зарядження накопичувача електроенергії на термін його служби.

Апробація результатів дисертації. Основні положення та результати роботи представлено та опубліковано на конференції ПМК 2022 (Прикладна Математика та Комп’ютинг).

Публікації. Результати дисертації викладено у науковій праці: тези «Математичне та програмне забезпечення системи купівлі / продажу електроенергії потужностей відновлюваної енергетики» на XV конференції молодих вчених «Прикладна математика та комп’ютинг - ПМК’2022» [1].

Ключові слова: гібридна енергетична система, низьковольтні енергомережі, мультиагентна система прийняття рішень, навчання з підкріпленням.

ABSTRACT

The thesis is presented in 84 pages. It contains 2 appendixes and bibliography of 47 references, 32 figures and 8 tables are given in the thesis.

Relevance of the topic. The introduction of renewable energy capacities allows us to distribute the load on the centralized electricity transmission network and reduce the share of electricity consumption from thermal power plants that have a negative impact on the environment.

Thesis connection to scientific programs, plans, and topics. The thesis was prepared according to the scientific research plan of the Applied Mathematics Department of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute.”

Research objective. To optimize the buying/selling of electricity in order to reduce the period of return on investment that was attracted when installing technical means for generating and storing renewable energy.

To achieve this goal, the following tasks were solved:

- Existing solutions for controlling nonlinear and complex systems are considered, in particular, reinforcement learning, fuzzy logic control, and control with model behavior prediction;
- A component and dynamic representation of a low-voltage power grid has been developed;
- Software was developed for a multi-agent system that buys/sells electricity.

Object of research. Hybrid power system, low-voltage power grid and its components, microgrid management systems: HOMER, HYBRID2 and RETScreen. The concept of arbitrage in the electricity market, processes of energy production, conversion and storage within a hybrid system.

Subject of research. Mathematical methods and software for the electricity buying/selling system, based on reinforcement learning models and algorithms that arbitrate electricity production and consumption.

Approaches and methods for controlling nonlinear and complex systems: reinforcement learning; fuzzy logic control and control with model behavior prediction.

Methods of research. The following methods were used to solve the problem: machine learning methods, reinforcement learning, control with model behavior prediction, modeling, design of Data Science systems, methods of algorithm theory and programming, methods of data analysis and mathematical statistics.

Scientific contribution. For the first time, a multi-agent decision-making system for the buying/selling of electricity was developed that combined simulation of the operation of an electricity storage device using the Doyle-Fuller-Newman model and processing of real data on solar radiation, meteorological indicators, electricity consumption and its cost, which allows making informed decisions on planning the operation of the microgrid to reduce electricity bills.

Practical significance of the results. Due to the proposed mathematical and programmatic representation of the electricity purchase/sale system, it is possible to reduce the electricity bill. Additionally, it becomes possible to study the impact of constant discharging and charging of the energy storage device on its service life.

Approbation of the thesis results. The main provisions and results of the work were presented and published at the conference AMC-2022 (Applied Mathematics and Computing).

Publications. The results of the dissertation are presented in the scientific work: "Mathematical and software system of buy/sale of electricity of renewable energy capacities" at the XV Conference of Young Scientists "Applied Mathematics and Computing – AMC'2022 [1].

Keywords: hybrid energy system, low-voltage power grids, multi-agent decision-making system, reinforcement learning.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	13
ВСТУП.....	14
1 АКТУАЛЬНІСТЬ.....	16
1.1 Стратегії стимулювання енергоефективності	16
1.2 Гібридна система генерації	17
1.3 Постановка задачі.....	20
1.4 Висновки до розділу 1.....	21
2. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ	22
2.1 Гібридна енергетична система	22
2.2 Мікромережа.....	22
2.3 Особливості мікромереж	24
2.4 Огляд теоретичних підходів.....	25
2.4.1 Непрогнозоване керування.....	26
2.4.2 Прогнозоване керування.....	26
2.5 Порівняння теоретичних підходів	29
2.6 Огляд практичних засобів	29
2.6.1 HOMER.....	30
2.6.2 HYBRID 2.....	31
2.6.3 RETScreen.....	31
2.7 Порівняльний аналіз	32
2.8 Висновки до розділу 2.....	33
3 КОМПОНЕНТНА МОДЕЛЬ СИСТЕМИ	34
3.1 Структурне представлення моделі	34
3.2 Опис функціональних компонентів.....	35
3.3 Динамічне представлення системи.....	36
3.4 Висновки до розділу 3.....	37

4 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КУПІВЛІ / ПРОДАЖУ ЕЛЕКТРОЕНЕРГІЇ	38
4.1 Опис компоненту «Solar PV Module»	38
4.1.1 Модель діода (single diode).....	38
4.1.2 Алгоритм MPPT.....	39
4.1.3 Сонячне випромінення.....	40
4.1.4 Набір даних	41
4.1.5 РОА випромінення	41
4.2 Опис компоненту «Electricity Storage».....	42
4.2.1 Вимоги до системи накопичення.....	43
4.2.2 Порівняльна характеристика.....	43
4.2.3 Модель SoC	45
4.2.4 Модель DFN.....	46
4.3 Опис компоненту «Decision-Making Module»	47
4.3.1 Види контролю мікромережі.....	47
4.3.2 Діаграма діяльності	47
4.3.3 Алгоритм навчання агентів	48
4.4 Опис компонентів «Power Grid» та «Consumer of Electricity».....	51
4.5 Висновки до розділу 4.....	52
5 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	54
5.1 Середовище	54
5.2 Валідація даних.....	54
5.2.1 Дані про сонячне випромінення	54
5.2.2 Дані про вартість електроенергії	57
5.2.3 Дані про споживання електроенергії будинком.....	57
5.3 Імплементація компоненту «Solar PV Module»	58
5.4 Імплементація компоненту «Electricity Storage».....	60
5.5 Процес обробки даних та створення навчального середовища	62
5.6 Імплементація компоненту «Decision-Making Module»	64
5.6.1 Середовище мікромережі	64

	12
5.6.2 Перелік дій	65
5.6.3 Винагорода	65
5.6.4 Опис інструментів	66
5.7 Висновки до розділу 5.....	67
6 ВЕРИФІКАЦІЯ ТА ВАЛІДАЦІЯ СИСТЕМИ КУПІВЛІ / ПРОДАЖУ ЕЛЕКТРОЕНЕРГІЇ	68
6.1 Формування тестової вибірки	68
6.2 Представлення отриманих результатів	68
6.3 Порівняння ефективності стратегій винагородження	71
6.4 Доцільність впровадження системи	73
6.5 Представлення роботи	76
6.6 Висновки до розділу 6.....	77
ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	80
Додаток А. Лістинги програм	85
Додаток Б. Ілюстративний матеріал.....	108

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ESS — Energy Storage System (система зберігання енергії).

ISO — Independent System Operator (незалежний системний оператор).

MPC — Model Predictive Control (модельне прогнозує керування).

MPPT — Maximum Power Point Tracking (точка максимальної потужності).

NOAA — National Oceanic and Atmospheric Administration (національне управління океанічних і атмосферних досліджень).

NSRDB — National Solar Radiation Database (національний набір даних про сонячну радіацію).

RL — Reinforcement Learning (навчання з підкріпленням).

SSE — NASA's Surface Meteorology and Solar Energy (наземні метеорологічні фактори та сонячне випромінення від NASA).

WIND — Wind Integration National Dataset (національний набір даних вітрової інтеграції).

ВСТУП

Глобальне потепління та екологічні проблеми в цілому змусили світ шукати альтернативні джерела енергії на заміну викопному паливу. Протягом останніх років відновлювані джерела енергії впевнено впроваджуються в побут завдяки своїм техніко-економічним та екологічним особливостям.

Частка споживання електроенергії приватними та промисловими будинками складає близько 36% у світі, що відповідає за 40% загальних викидів CO₂ [2]. Інтеграція систем генерації відновлювальної електроенергії дозволить перейти до низьковуглецевої енергетики, а також розширити можливості споживачів енергії, оскільки вони можуть значно сприяти децентралізації та самоспоживанню. Однак, стохастичний процес генерації, обмежена ємність накопичувачів, втрати потужності при перетворенні струму та ряд інших технічних факторів не дозволяють задовільнити всі потреби людини.

Цілком очевидно, що для вирішення зазначених проблем необхідні значні зміни у своїй інфраструктурі виробництва, зберігання та транспортування електроенергії. Гібридні енергетичні системи, як вони визначені в літературі [3, 4], дозволяють перейти до розумного та дбайливого споживання електроенергії.

Проте, величина початкових інвестицій у відповідне технічне забезпечення та середній термін окупності відлякують потенційних інвесторів, що робить майбутній перехід до гібридних систем електроспоживання довготривалим. Ця проблема ускладнюється ще й тим, що енергетичні потужності відновлюваної енергетики та засоби управління ними потребують високотехнологічних рішень і систем обробки великих обсягів даних. Тому впровадження інтелектуальних систем, що здатні обробляти великі об'єми даних та приймати оптимальні рішення в межах гібридних систем, дозволяє зменшити період окупності початкових капіталовкладень.

Таким чином, основною задачею дисертаційного дослідження є пошук та розробка ефективних стратегій керування гібридною мережею. Більш того, кожне конкретне рішення повинно враховувати фізичні обмеження компонентів «розумної»

мережі, дані про споживання електроенергії користувачами, метеорологічні фактори та вартість електроенергії, які визначають місцеві провайдери.

1 АКТУАЛЬНІСТЬ

1.1 Стратегії стимулювання енергоефективності

Світова політика енергоефективності «спрямована на стимулювання споживачів до більш ефективного використання енергії» [5]. Регулюючі органи розробляють та впроваджують програми управління попитом, тобто програми, спрямовані на економію електроенергії шляхом зміни моделей використання енергії споживачами.

Це може бути досягнуто за рахунок загального скорочення споживання енергії, що призводить до прямого підвищення енергоефективності. Але існують й інші стратегії управління попитом:

- стратегія управління навантаженням, яка спрямована на зниження споживання електроенергії в години пік. Завдяки інтелектуальним лічильникам споживачі мають можливість отримувати інформацію про ціну на електроенергію в режимі реального часу. Вартість електроенергії в періоди пікового навантаження вища в порівнянні з періодами низького попиту, що стимулює людей до зміни поведінки використання електроенергії [6];
- стратегії стимулювання бізнесу для розробки та розповсюдження енергоефективних приладів;
- стратегії належного інформування споживачів.

Ефективність зазначених стратегій підвищення енергоефективності безпосередньо залежить від людини та її поведінки. Що вносить невизначеність та не може гарантувати стійкість програми розумного споживання електроенергії протягом тривалого періоду часу.

1.2 Гібридна система генерації

Як альтернативу централізованій розглядають гібридну (розподілену) систему генерації – набір технологій, які дозволяють використовувати та керувати електроенергією децентралізовано та в невеликих масштабах, розміщуючи елементи генерації поблизу елементів навантаження. Відповідно, гібридна система генерації передбачає, що споживачі самостійно, повністю чи частково, виробляють необхідну їм енергію. Такі споживачі можуть зробити систему електропостачання більш ефективною, оскільки попит на електроенергію з центральної мережі зменшиться, а відповідно зменшиться і навантаження на централізовану мережу транспортування.

Крім того, гібридна система генерації тісно взаємодіє з відновлювальними джерелами енергії: сонячна, вітрова або геотермальна (рис. 1.1).

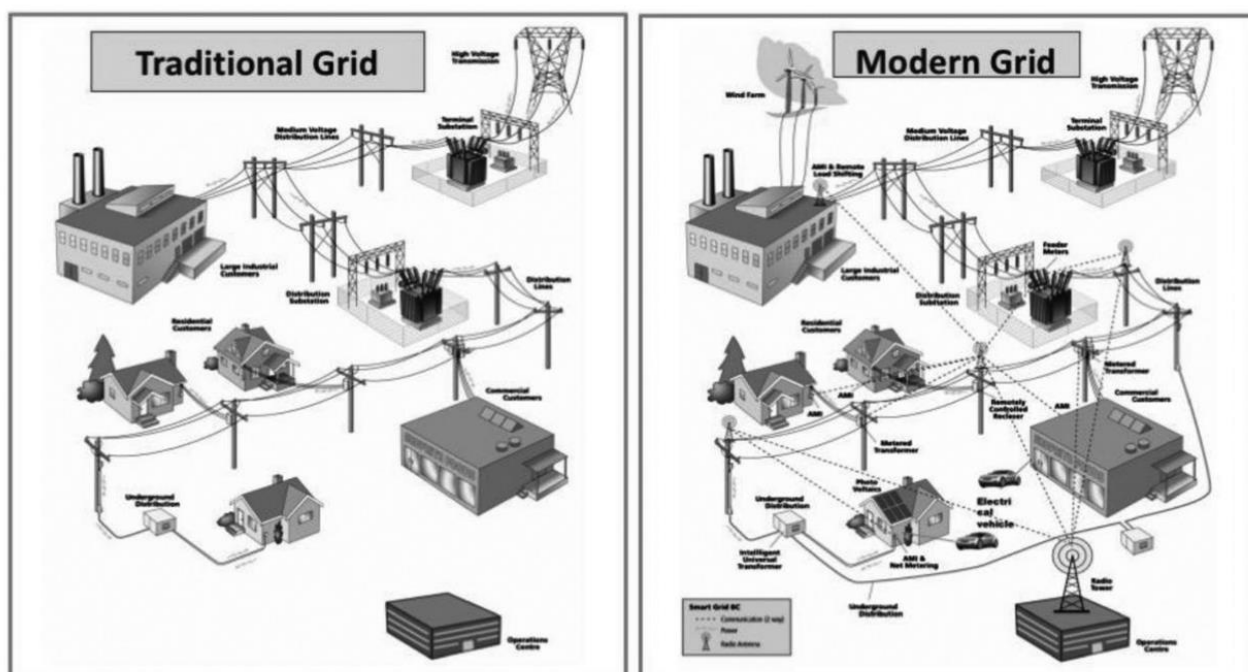


Рисунок 1.1 – Гібридна енергетична система в порівнянні із традиційною [7]

Згідно з програмою низьковуглецевого енергетичного переходу стає очевидним, що такі джерела енергії повинні замінити викопне паливо. Однак джерела сонячної та вітрової енергії є нестійкими, стохастичними джерелами. Іншими словами,

генерація електроенергії відбувається непостійно і залежить від погодних умов. Тим не менш, «зелена» електроенергія виробляється з дуже низькою граничною вартістю без виснаження природних ресурсів. Отже, використання енергії, виробленої сонячними та вітровими джерелами, та її раціональне використання дає значний приріст ефективності, а також економічну вигоду, не кажучи вже про її важливість для зусиль щодо пом'якшення наслідків зміни клімату.

Хоча гібридна енергетична система має цілу низку переваг, існують деякі труднощі та проблеми, пов'язані з гібридними енергетичними системами, які необхідно вирішити [8]:

а) Передача електроенергії від відновлювальних потужностей до центральної комунальної мережі на даний момент відбувається неконтрольовано. Комунальні підприємства контролюють рівень напруги у центральній мережі та здійснюють подачу електроенергії, коли це необхідно, щоб згладити природні коливання споживання та підтримувати напругу на прийнятному рівні. Але під час подачі електроенергії відновлюваних джерел до мережі комунальні підприємства не можуть відстежувати звідки надходить надлишок електроенергії. Для того, щоб інтегрувати відновлювальні джерела енергії в існуючу інфраструктуру, необхідно вирішити декілька основних проблем, які роблять систему нестабільною:

- 1) Управління напругою. Організувати додатковий рівень контролю над рівнем напруги можна за рахунок вторинних та третинних регуляторів: компенсатори реактивної потужності; системи зберігання запасів електроенергії; полюсні трансформатори для дистанційного керування розгалуженням подачі електроенергії;
- 2) Регулювання частоти [9]. При додаванні відновлювальних джерел електроенергії до центральної мережі часто виникають нестабільні частоти. Знайдене рішення повинно регулювати коливання частоти в заданому діапазоні;

- 3) Керування коливаннями вихідної потужності. Поглинання надлишкової енергії у разі надмірних коливань потужності допоможуть підтримувати плавну криву потужності;
- 4) Імпульсні навантаження. Ця проблема стає все більш складною, оскільки електроавтомобілі набувають все більш широкого поширення, а швидка зарядка може призвести до раптового збільшення навантаження на систему. Накопичувачі електроенергії можуть використовуватися для зберігання надлишкової енергії, мінімізуючи негативний вплив заряджання електроавтомобіля на центральну мережу.
- б) Гібридна енергетична система складається із різноманітної кількості компонентів, кожен з яких виконує конкретну функцію. Щоб узгодити роботу цих компонентів для досягнення найвищого рівня стабільності мережі необхідно мати відповідні механізми керування та моніторингу. Крім того, гібридна мережа повинна включати окремий рівень управління для виконання оптимізаційних операцій мережі, в основному спрямованих на управління пристроями накопичення електроенергії, планування та диспетчеризацією розподіленої генерації, а також управління імпорту та експорту електроенергії до центральної комунальної мережі.
- в) Величина початкових інвестицій у відповідне технічне забезпечення та середній термін окупності відлякують потенційних інвесторів, що робить майбутній перехід до гібридних систем електроспоживання довготривалим. Тому планування і оптимізація схем постачання та споживання електроенергії за рахунок використання інформації про змінну вартість електроенергії протягом дня, дозволить зменшити час окупності придбаного технічного забезпечення.

1.3 Постановка задачі

Гібридна енергетична система є майбутнім енергетичної промисловості, яка дозволить відмовитися від викопного палива, зменшити залежність від централізованої енергетичної мережі, споживати високоякісну та дешеву електроенергію. Однак, вартість технічних компонентів гібридної системи та труднощі, що виникають при керуванні енергетичними потужностями, не дозволяють здійснити перехід до нової системи генерації, транспортування та споживання електроенергії в найближчий період часу.

Таким чином, виникає потреба в засобах управління енергетичними потужностями, що використовують інтелектуальні системи, які здатні обробляти великі об'єми даних та приймати оптимальні чи квазіоптимальні рішення в межах гібридних систем, щоб зменшити період окупності початкових капіталовкладень.

Об'єктом дослідження є гібридна енергетична система, низьковольтна енергомережа та її складові, системи керування мікромережами: HOMER, HYBRID2 та RETScreen. Поняття арбітражу на ринку електроенергії, процеси виробництва, перетворення та зберігання енергії в межах гібридної системи.

Для розв'язання поставленої задачі повинні використовуватися такі методи як методи машинного навчання, навчання з підкріпленням, управління з передбаченням поведінки моделі, математичного моделювання, проектування систем Data Science, методи теорії алгоритмів та програмування, методи математичної статистики та аналізу даних.

До об'єкту дослідження також відносяться підходи та методи управління нелінійними та складними системами: навчання з підкріпленням; нечітке логічне управління та управління з передбаченням поведінки моделі.

Предметом дослідження є математичне та програмне забезпечення системи купівлі / продажу електроенергії, в основу яких покладені моделі та алгоритми навчання з підкріпленням, що здійснюють арбітраж виробництва та споживання електроенергії.

Метою дослідження є оптимізація процесу купівлі / продажу електроенергії з метою зменшення періоду повернення інвестицій, які були залучені при встановленні технічних засобів для генерації та збереження відновлюваної енергії.

Результат даної роботи – моделі і алгоритми прийняття рішень для системи купівлі / продажу електроенергії. Математичне, програмне та методичне забезпечення відповідної системи.

Для досягнення вказаної мети було розв'язано такі задачі:

- розглянуто існуючі рішення управління нелінійними та складними системами, зокрема, навчання з підкріпленням, нечітке логічне управління та управління з передбаченням поведінки моделі;
- розроблено компонентне та динамічне представлення низьковольтної енергомережі;
- розроблено програмне забезпечення для мультиагентної системи, що здійснює купівлю / продаж електроенергії.

1.4 Висновки до розділу 1

В розділі 1:

- 1) перелічено головні стратегії стимулювання енергоефективності, які спрямовані на людську відповідальність та обізнаність;
- 2) розглянуто альтернативний варіант, який передбачає впровадження гібридних енергомереж;
- 3) встановлено, що вартість технічного забезпечення і складність процесів гібридної мережі не дозволяють зробити перехід до «розумного» споживання швидким.

2. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

2.1 Гібридна енергетична система

Гібридна енергетична система оперує компонентами генерації, збереження та транспортування електроенергії на різних рівнях енергосистеми. Зазвичай, ці рівні прийнято розділяти таким чином [10]:

- макромережі (macrogrids). Зазвичай, це комунальні енергосистеми всієї країни, що забезпечують надійне енергопостачання за низькою ціною. Включають промислові об'єкти генерації, наприклад, атомні електростанції; розгалужену систему транспортування та перетворення електроенергії на шляху до кінцевого користувача.
- мікромережі (microgrid) або низьковольтні енергомережі. Прикладами можуть бути університети, лікарні та військові бази. Мікромережа підключена до макромережі, але має можливість виробляти власну електроенергію для покриття попиту та забезпечення стійкості.
- автономні мережі. Острови та віддалені місця, які не під'єднані до макромережі.

В цій роботі, розглядаються мікромережі, які можуть взаємодіяти із макромережею (центральною) і мають змогу виробляти власну електроенергію.

2.2 Мікромережа

У літературі можна знайти ряд визначень для мікромережі. В роботі [10] мікромережу представляють як концепцію, що оперує множиною електричних та теплових навантажень (споживачів) разом із невеликими джерелами електричної та теплової енергії, включаючи системи збереження енергії. Джерела енергії, зазвичай, гетерогенні: відновлювальні джерела (фотоелектричні та вітрові станції); геотермальні станції; мікротурбіни; паливні елементи (електрохімічні генератори)

тощо. Взаємозв'язок між цією та центральною (макро) мережею здійснюється через чітко визначений інтерфейс.

Мікромережі можуть використовуватися в цілях одного споживача, житлової забудови, навчальних та дослідницьких кампусів, комерційних/промислових хабів та міської забудови з мільйонами мешканців [11]. Приклад мікромережі, спроектованої для міської забудови, наведено на рис. 2.1.

Мікромережі можуть запропонувати кінцевим користувачам:

- можливість зниження загальних витрат;
- підвищення ефективності, надійності та безпеки (порівняно з макромережами);
- перехід до екологічно чистих джерел електроенергії, таких як відновлювальні джерела електроенергії та паливні елементи.

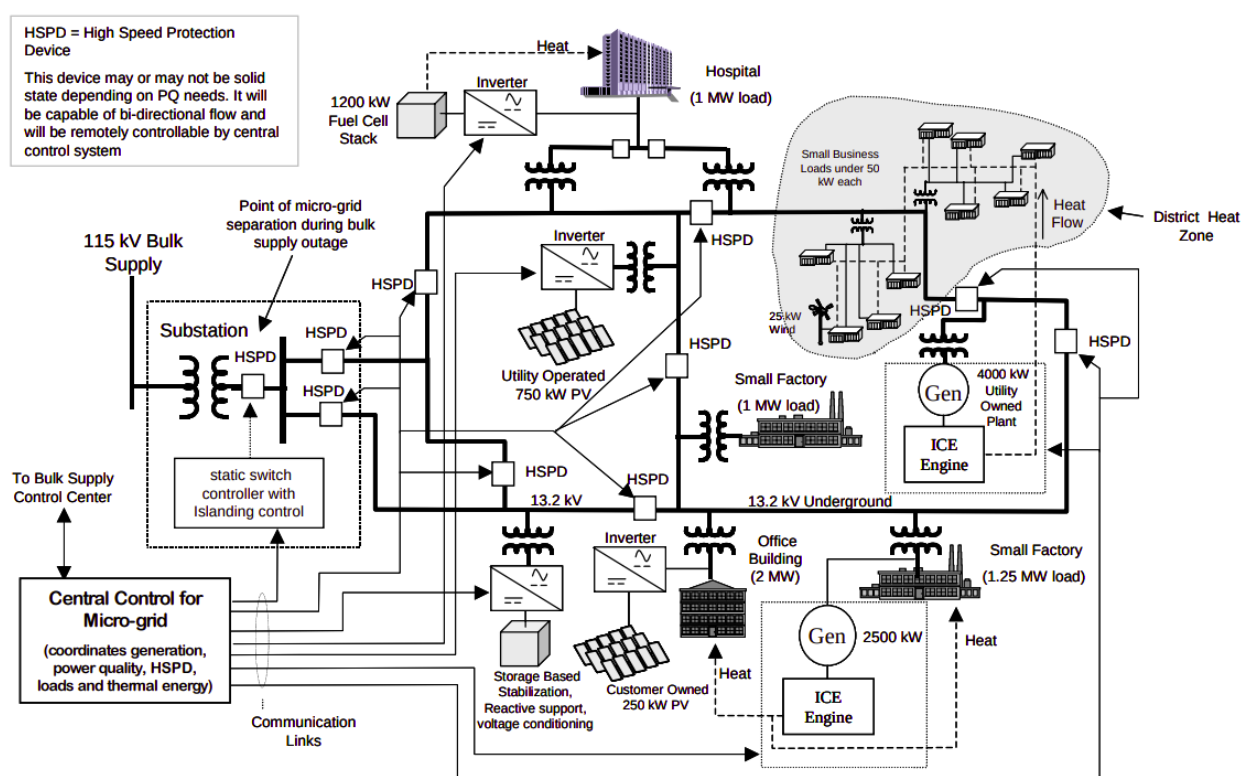


Рисунок 2.1 – Мікромережа міської забудови [11]

2.3 Особливості мікромереж

Інфраструктура мікромережі була розроблена з кінцевою метою уникнути ситуації, коли необхідно суттєво наростити виробничі потужності центральних блоків генерації, щоб задовільнити постійно зростаючу потребу в електроенергії кінцевими користувачами.

Мікромережа активно використовує розумні прилади «Інтернету речей», які можуть відстежувати і контролювати потоки електроенергії в мережі. Це дозволяє встановити відповідність між обсягом генерації електроенергії та її споживанням, а також мінімізувати загальну вартість електроенергії, що транспортується до кінцевого користувача.

Комунальні підприємства можуть застосовувати стратегію динамічного ціноутворення на електроенергію, тобто встановлювати різні ціни на електроенергію в залежності від часу доби та місця. Така політика стимулюватиме споживачів здійснювати управління попитом, шляхом пристосування їхніх навантажень до поточного стану мережі. Іншими словами, зміщення навантаження з пікових періодів на непікові. Ключову роль в управлінні попитом відіграють відновлювальні джерела енергії, зокрема фотоелектричні джерела.

Однак, існує невідповідність між часом максимальної генерації електроенергії відновлювальними джерелами і часом пікового споживання. Це призводить до того, що згенерована електроенергія не може бути оптимально використана для зменшення пікового навантаження. Більш того, в кожен момент часу вихідна потужність фотоелектричних джерел залежить від змінного сонячного випромінення.

Ефективним вирішенням вищезазначених проблем є впровадження систем накопичення електроенергії [12]. Ці системи збереження електроенергії в складі мікромережі дозволяють накопичувати енергію в непікові періоди дня від центральної мережі або фотоелектричних модулів, тоді як в пікові — забезпечувати енергією кінцевого користувача або продавати до комунальної мережі.

Таким чином, впровадження стратегій управління системою накопичення енергії з урахуванням цін на електроенергію та індивідуальних потреб споживача є стратегічно важливим для впровадження «розумних» мікромереж.

2.4 Огляд теоретичних підходів

В роботі [13] наведено таксономію існуючих підходів (рис. 2.3), які розглядають керування мікромережами для досягнення конкретних цілей.

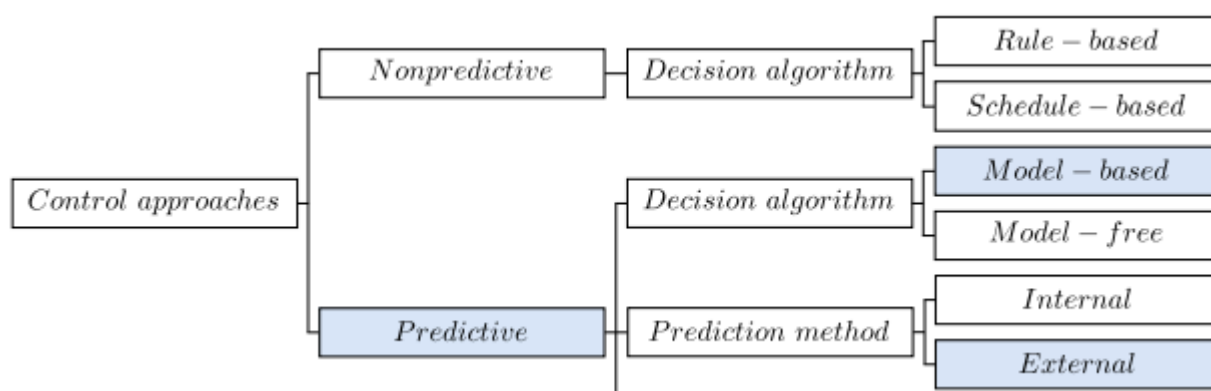


Рисунок 2.2 – Таксономія існуючих методів [12]

Автори роботи [13] детально розглядають два класи методів:

- Непрогнозоване керування. Клас методів, що використовує чіткі правила (інструкції) для прийняття рішень, орієнтуючись на виконання певних умов. Наприклад, продати електроенергію до центральної мережі, яка була накопичена протягом дня від сонячних панелей, о 17:00 годині дня.
- Прогнозоване керування. Клас методів, що визначають контроль мікромережі на основі прогнозованого попиту, кількості виробленої енергії відновлювальними джерелами та ціни.

2.4.1 Непрогнозоване керування

У роботі [14] розглядається два джерела електроенергії: фотоелектричний модуль (сонячні панелі) та централізована мережа. Додатково використовується Li-ion батарея для збереження електроенергії. Автори роботи намагаються зменшити рахунок за електроенергію. Процес розрядження Li-ion батареї триває протягом конкретного період часу за умови, що стан заряду (State of charge, SoC) складає конкретного значення (визначеного авторами без шкоди для батареї). При цьому, процес зарядження і розрядження чітко визначений: сонячна панель — батарея — центральна мережа.

2.4.2 Прогнозоване керування

Цей клас методів прийнято розділяти на дві групи:

- алгоритми модельного прогнозуючого керування (model predictive control, MPC), формулюють фізичну модель світу і розв'язують її за допомогою точного або наближеного математичного оптимізаційного підходу;
- алгоритми, що опосередковано взаємодіють із фізичною моделлю світу, зазвичай прогнозуючи результати роботи або фактори, що безпосередньо впливають на її роботу. При цьому, внутрішні властивості моделі залишаються прихованими. Конкретне рішення ґрунтується або на експертних знаннях (експертні оцінки, нечітка логіка) або на досвіді взаємодії із середовищем (навчання з підкріпленням).

2.4.2.1 Перший клас (явне використання моделі)

В роботах [15-18] розглядається взаємодія між акумуляторною батареєю з сонячними фотоелектричними панелями. Вони розглядають два підходи:

- лінійну оптимізацію для мінімізації функції витрат;
- квадратичну оптимізацію для зменшення зворотного потоку енергії до центральної мережі в пікові моменти навантаження.

Однак автори цих робіт брали до уваги ідеальні прогнози навантаження та сонячної активності. Безсумнівно, такий підхід не є реалістичним для систем контролю в реальних умовах.

У роботі [19] запропоновано алгоритм детермінованого динамічного програмування для енергоменеджменту процесів будинку у Фінляндії. Цей будинок був обладнаний фотоелектричними панелями, накопичувачем електроенергії та змінними навантаженнями. Автори роботи порівняли ефективність алгоритму максимального самоспоживання із / без підключення до центральної мережі, що базується на правилах. Аналіз проводився протягом року, використовуючи метод ковзного вікна з горизонтом прогнозування і керування 24 години з роздільною здатністю в 1 годину. Результати роботи показують, що керувати навантаженням найефективніше, використовуючи системи накопичення електроенергії.

У роботі [20] було запропоновано детермінований алгоритм MPC для енергоменеджменту житлового будинку, що містить фотоелектричні елементи, акумуляторну батарею та тепловий насос. Дослідження авторів було зосереджене на потенціалі скорочення витрат на основі гнучкості попиту та динамічних цін на електроенергію. Використовуючи дані про навантаження, вони виявили, що системи енергоменеджменту, які працюють із використанням інформації про ціни на добу вперед або в режимі онлайн, можуть бути корисними для енергосистеми і приносити власникові значні прибутки. В роботі була сформульована квадратична задача, вирішили її за допомогою методу ковзного горизонту з горизонтом прогнозування

16 годин та роздільною здатністю в 1 годину. Було підраховано річні витрати на електроенергію та залежність від центральної енергомережі для різних сценаріїв.

Узагальнюючи клас алгоритмів, що явно використовують фізичне представлення моделі світу, формулюють задачу управління електроенергією як нелінійну оптимізаційну задачу. Для рішення якої використовуються оптимізаційні методи, такі як змішане цілочисельне програмування, послідовне квадратичне програмування [21], оптимізація рою частинок [22], нейронні мережі [23] тощо.

2.4.2.2 Другий клас (неявне використання моделі)

Як зазначалося раніше, існує клас алгоритмів, що неявно використовують фізичну модель і приймають рішення на основі експертних знань. У роботі [24] запропоновано експертну систему, що використовує нечітку логіку (fuzzy logic). Як зазначають автори роботи, така інтелектуальна система здатна приймати рішення в умовах невизначеності або часткової інформації про модель світу. Однак, залишається залежною від експертних знань та не дозволяє досягти достатнього рівня адаптивності і оперативно реагувати на нові зміни.

Альтернативно розглядають алгоритми, які використовують теорію ігор із аналізом рівноваги рішення [25, 26] та навчання з підкріпленням (Reinforcement Learning, RL). Останні успішно застосовуються для пошуку стратегій керування мікромережами і детально описані в роботах [27-30].

2.5 Порівняння теоретичних підходів

Відсутність єдиної метрики ефективності не дозволяє об'єктивно порівняти кожен із запропонованих алгоритмів. Ситуація ускладнюється ще й тим, що дослідження в рамках кожної із робіт проводилось за різних умов, різної архітектури мікромережі та набору даних. Однак, можна виділити ряд якісних характеристик, які притаманні кожному із класів.

Таблиця 2.1 – Якісні характеристики непрогнозованого та прогнозованого керування

Вид керування	Адаптивність до зовнішніх умов	Фактор людської помилки	Ігнорування наявної інформації
Непрогнозоване	відсутня	більший	більший
Прогнозоване	присутня	менший	менший

Порівняльна характеристика між алгоритмами, що використовують фізичну модель світу явно / неявно, зводиться до часової складності рішення. Через гетерогенність алгоритмів явно визначити її складно, однак, посилаючись на роботу [17], можна припустити, що явне використання є більш часоємним.

2.6 Огляд практичних засобів

Окрім різних математичних моделей, що використовуються для аналізу поведінки компонентів мікромережі, програмне забезпечення для моделювання та оптимізації також є важливою складовою для різноманітних застосувань. Сфери застосувань включають проектування, стратегію управління, економічну оптимізацію та багатоцільову оптимізацію.

У роботі [13] програмні інструменти, пов'язані із мікромережами, класифікували на три групи: попереднє техніко-економічне обґрунтування компонентів; визначення розмірів; моделювання середовища (постфактум).

Було відібрано практичні рішення, які орієнтовані на менеджмент мікромережі. До цього списку належать такі реалізації.

2.6.1 HOMER

Гібридна модель оптимізації для відновлювальних джерел електроенергії (Hybrid Optimization Model for Electric Renewables, HOMER) [31] є найбільш поширеним, вільно доступним програмним забезпеченням для проведення швидкого попереднього техніко-економічного обґрунтування, оптимізації та аналізу чутливості визначеної конфігурації мікромережі.

Переваги:

- а) використовує достовірні дані (рівень дискретизації складає 60 хв.):
 - 1) національний набір даних про сонячну радіацію (National Solar Radiation Database, NSRDB);
 - 2) національний набір даних вітрової інтеграції (Wind Integration National Dataset, WIND);
 - 3) набір даних про сонячне випромінювання та наземні метеорологічні фактори від NASA (NASA's Surface Meteorology and Solar Energy, SSE);
 - 4) набір даних від національного управління океанічних і атмосферних досліджень (National Oceanic and Atmospheric Administration, NOAA).
- б) є можливість завантажувати власні метеорологічні дані;
- в) наявна обширна документація користування та допоміжні матеріали.

Недоліки:

- не розглядає мережі із складною конфігурацією;

- не враховуються характеристики накопичувачів електроенергії, наприклад глибина розряду акумуляторної батареї;
- комерційне використання вимагає платної ліцензії.

2.6.2 HYBRID 2

HYBRID 2 [32]— це імовірнісна модель прогнозування часових залежностей, яка використовує статичні методи для врахування варіацій між спостереженнями в часі. Додатково може виконувати довгостроковий аналіз продуктивності та економічних показників мікромережі різної конфігурації.

Переваги:

- використовує достовірні дані (рівень дискретизації складає 60 хв.): NSRDB, SSE, NOAA;
- розділена відповідальність між функціональними блоками: графічний інтерфейс користувача, модуль моделювання, економічний модуль, економічний модуль;

Недоліки:

- обмежена документація;
- не підтримується та не оновлюється розробниками.

2.6.3 RETScreen

RETScreen [33] використовують для аналізу різних типів енергоефективних та відновлювальних технологій, що охоплює в основному виробництво енергії, витрати життєвого циклу та зменшення викидів парникових газів.

Глобальна кліматична база складається з понад 6000 наземних станцій (дані про сонячне випромінення), карти енергетичних ресурсів (наприклад, карти вітрів), гідрологічні дані, технологічні деталі сонячних панелей і криві потужності вітрогенераторів.

Переваги:

- а) використовує достовірні дані (рівень дискретизації складає 10-60 хв.):
 - 1) SSE, NOAA;
 - 2) дані від європейського центру прогнозування погоди (European Centre for Medium-Range Weather Forecasts, ECMWF);
- б) є інструменти фінансового аналізу та аналізу ризиків для підтримки прийняття рішень.

Недоліки:

- першочергово орієнтований на проведення фінансового аналізу мікромережі, контроль мікромережі не здійснюється;
- обмежена інтеграція із сторонніми джерелами даних та сервісами;
- не враховується температура зовнішнього середовища при генерації електроенергії фотоелектричними модулями.

2.7 Порівняльний аналіз

Розглянуті практичні засоби орієнтовані на створення конфігурації мікромережі і проведення фінансового аналізу, використовуючи дані про сонячне випромінення та погоду. Однак, рішення, що контролюють внутрішні процеси мікромережі в реальному часі, відсутні.

2.8 Висновки до розділу 2

В розділі 2 було:

- 1) наведено таксономію теоретичних підходів, що контролюють процес розрядження та зарядження системи накопичення електроенергії;
- 2) аргументовано перевагу передбачувальних алгоритмів над алгоритмами, побудованими на чітких правилах;
- 3) наведено компроміс між часовою складністю та неповнотою системи (відома реакція фізичної моделі на зовнішні фактори або результат роботи) не дозволяє однозначно визначити, який із класів алгоритмів краще підходить для вирішення поставленої задачі.

Узагальнюючи, можна сформулювати таку гіпотезу із подальшим її спростуванням або підтвердженням: комбіноване використання алгоритмів, які по-різному використовують фізичну модель, дозволить розділити функціональну відповідальність між ними для досягнення кінцевої мети роботи.

3 КОМПОНЕНТНА МОДЕЛЬ СИСТЕМИ

3.1 Структурне представлення моделі

Компонентна модель системи купівлі / продажу електроенергії в рамках низьковольтних енергомереж складається з таких компонентів (див. рис. 3.1): фотоелектричного модуля (Solar PV Module), накопичувача електроенергії (Electricity Storage), центральної комунальної мережі (Power Grid), споживача електроенергії (Consumer of Electricity) та блоку прийняття рішень (Decision-Making Module).

Головне призначення системи – узгодження операцій: генерації, перетворення, передачі та збереження електроенергії в рамках використання індивідуальним споживачем (будинком), з кінцевою метою зменшити величину оплати за електроенергію.

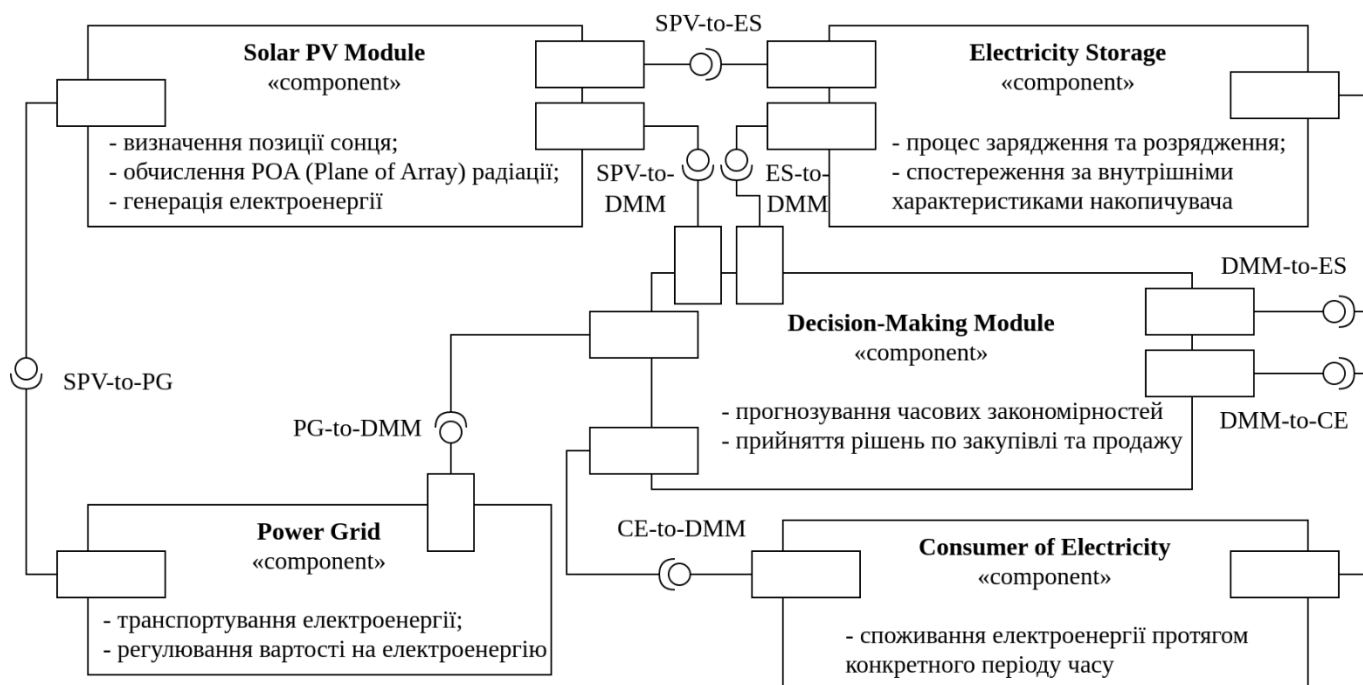


Рисунок 3.1 – Уніфікована модель системи купівлі / продажу електроенергії.

Діаграма компонентів у нотації UML [1]

Характер взаємодії компонентів між собою встановлюється за допомогою визначених інтерфейсів:

- SPV-to-PG – інтерфейс передачі інформації про кількість надлишкової згенерованої електроенергії, яку необхідно продати;
- SVP-to-ES – інтерфейс передачі інформації про кількість згенерованої електроенергії, яку необхідно зберегти;
- PG-to-DMM – інтерфейс передачі інформації про вартість електроенергії за попередні періоди часу;
- ES-to-DMM – інтерфейс передачі інформації про стан заряду накопичувача відповідно до його номінальної ємності за попередні періоди часу;
- CE-to-DMM – інтерфейс передачі інформації про історію споживання електроенергії споживачем за попередні періоди часу;
- SPV-to-DMM – інтерфейс передачі інформації про метеорологічні дані та історію згенерованої електроенергії;
- DMM-to-CE – інтерфейс передачі інструкцій від центрального модуля прийняття рішень до споживача. Визначає джерело електроенергії, яке надасть її споживачу в наступний період часу;
- DMM-to-ES – інтерфейс передачі повідомлень від центрального модуля прийняття рішень до накопичувача. Визначає доцільність продажу або купівлі електроенергії від центральної мережі.

3.2 Опис функціональних компонентів

Кожен із компонентів системи купівлі / продажу електроенергії має власні функціональні особливості [1]:

- компонент «Solar PV Module» відповідає за генерацію електроенергії за чітко визначених умов середовища: рівня сонячної випромінювання, що потрапляє на сонячні панелі, температури активних модулів, матеріалів панелі;
- компонент «Electricity Storage» відповідає за збереження електроенергії. Процес зарядження та розрядження накопичувача регулюється контролером;

- компонент «Power Grid» визначає тариф на продаж та закупку електроенергії згідно пори доби та навантаження;
- компонент «Consumer of Electricity» визначає величину електроенергії, яка необхідна для коректної роботи побутових приладів, підтримки температури та освітлення приміщень;
- компонент «Decision-Making Module» здійснює керування мікромережею і синхронізує дій між її компонентами. Приймає рішення коли і в яких об'ємах купити або продати електроенергію, яке джерело електроенергії використати, щоб задовільнити потреби споживача.

3.3 Динамічне представлення системи

Динамічне представлення системи найвищого рівня ієрархії наведено на рис.

3.2. Ця діаграма формалізує взаємодію компонентів на рівні продукування та споживання проміжних ресурсів відповідними процесами.

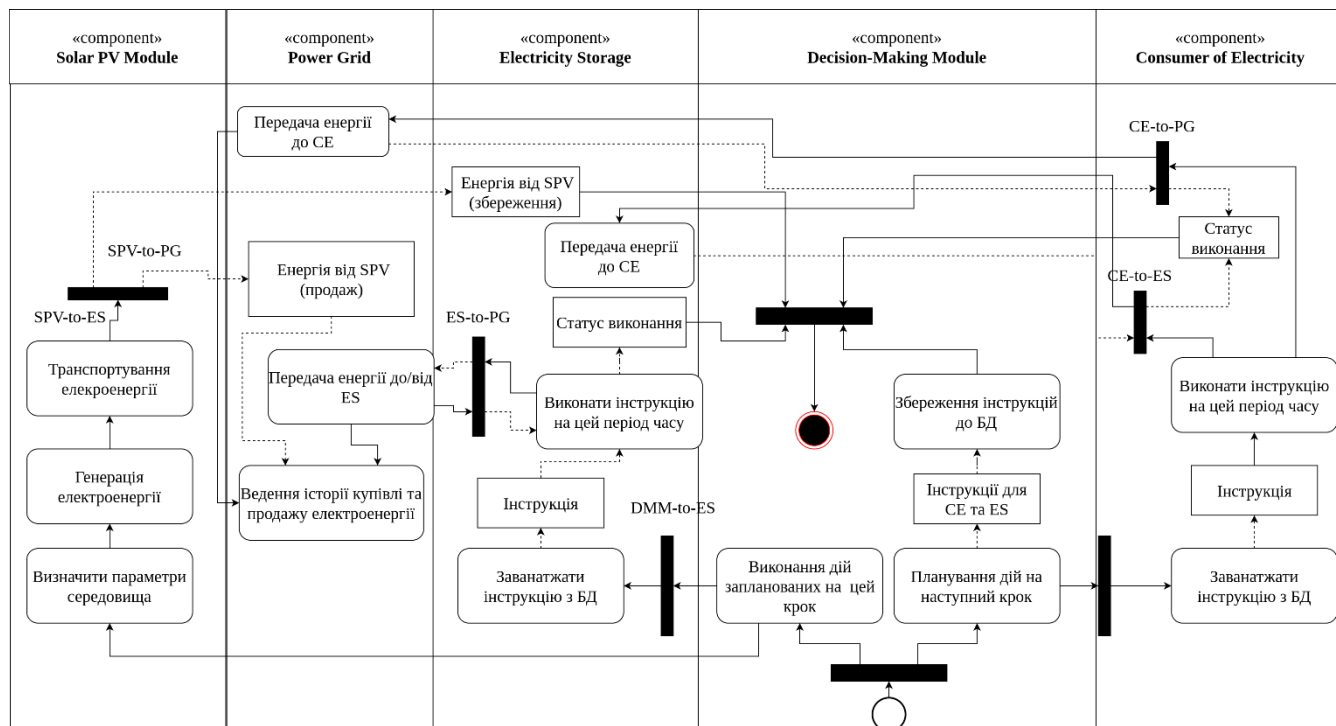


Рисунок 3.2 – Уніфікована модель системи купівлі / продажу електроенергії.

Діаграма діяльності у нотації UML [1]

3.4 Висновки до розділу 3

В розділі 3:

- 1) запропоновано компонентне представлення низьковольтної мережі із описом відповідних компонентів та інтерфейсів взаємодії;
- 2) наведено динамічне представлення моделі системи купівлі/продажу електроенергії.

4 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ КУПІВЛІ / ПРОДАЖУ ЕЛЕКТРОЕНЕРГІЇ

4.1 Опис компоненту «Solar PV Module»

4.1.1 Модель діода (single diode)

Процес перетворення енергії сонця (випромінення) в електричну енергію відбувається згідно з фізичною моделлю одиничного діода. Алгебраїчне та графічне представлення моделі наведено рівністю 4.1 та на рис. 4.1 відповідно [34].

$$I_{PV} = I_{PH} - I_s \left[\exp \left(\frac{q}{AKT} \right) (V_{PV} + I_{PV} R_s) - 1 \right], \quad (4.1)$$

де I_{PV} та V_{PV} – результуюча сила струму та напруга відповідно; R_s – опір резисторів; I_{PH} – величина сонячного випромінення; I_s – струм насичення; q – заряд електрона; K – стала Больцмана; A – коефіцієнт ідеальності діода, T – температура навколишнього середовища.

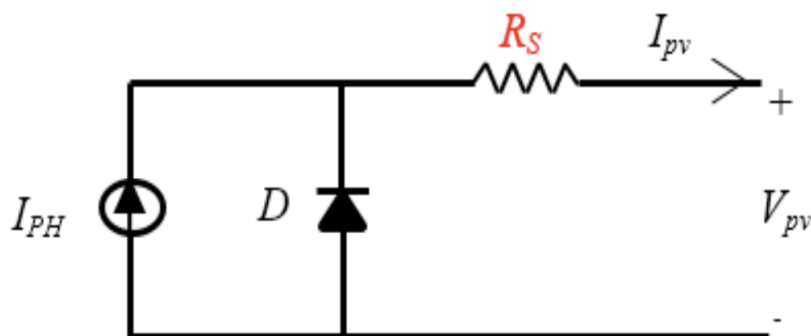


Рисунок 4.1 – Модель одного діода [33]

4.1.2 Алгоритм MPPT

Модель одиничного діода дозволяє однозначно визначати величину електричної потужності $P_{PV} = I_{PV} \times V_{PV}$. Однак ця величина не є постійною і залежить від величини вихідного струму (I_{PV}), яка безпосередньо залежить від температури навколишнього середовища та освітлення. Характер поведінки кривої потужності від величини струму наведено на рис. 4.2. Для збільшення продуктивності фотоелектричного елемента використовується алгоритм відстеження максимальної точки потужності (Maximum Power Point Tracking, MPPT) [35].

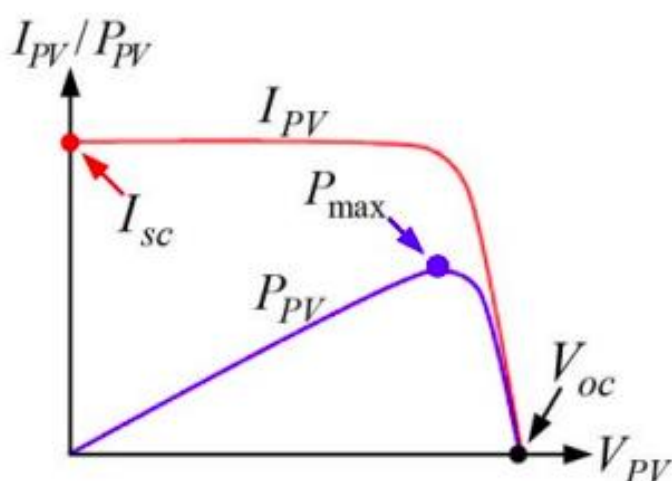


Рисунок 4.2 – Поведінка кривої потужності [34]

Протягом дослідження будемо припускати, що модель одиничного діода і алгоритм MPPT неперервно пов'язані. І на виході від фотоелектричного елемента отримується максимальне значення потужності за даних умов середовища.

4.1.3 Сонячне випромінення

Головною змінною, яка впливає на вихідну потужність фотоелектричного елемента, є сонячне випромінення (solar irradiance). Сонячне випромінення (Ват/м^2) — це величина потужності отриманої енергії від сонця відносно одиниці площі поверхні [36]. Для практичних цілей, вимірювання та інтерпретації сонячного випромінення розділяють на три складові: DNI, DHI та GHI (рис. 4.3).

Розсіяне нормальне випромінення (Diffused Normal Irradiance, DNI) — кількість короткохвильового випромінення, яке перпендикулярно потрапляє на активну поверхню.

Розсіяне горизонтальне випромінення (Diffused Horizontal Irradiance, DHI) — кількість короткохвильового випромінення, яке потрапляє на активну поверхню, заломлюючись від водяних частинок атмосфери або відбиваючись від поверхні землі.

Загальне горизонтальне випромінення (Global Horizontal Irradiance, GHI) — кількість короткохвильового випромінення, яке безпосередньо потрапляє на активну поверхню. Значення GHI є найголовнішою змінною при розрахунку вихідної потужності фотоелектричного елемента.



Рисунок 4.3 – Складові сонячного випромінення

4.1.4 Набір даних

Виділяють два провайдери даних про сонячне випромінення:

- наземні метеорологічні станції, які здійснюють заміри GHI за допомогою піранометра, піргеліометра або інших технічних засобів;
- орбітальні станції, які здійснюють заміри DNI.

Цілком очевидно, що орбітальні станції надають інформацію про рівень DNI випромінення для кожної точки земної поверхні, однак, для обчислення GHI треба враховувати додаткову інформацію про атмосферний тиск та повітряні маси.

До сервісів, які надають інформацію про сонячне випромінення, зібрану за допомогою орбітальних станцій, належать:

- SolarGIS та Satel-Light Project;
- World Radiation Monitoring Center's Baseline Surface Radiation Network (BSRN);
- набір даних від Всесвітнього Радіаційного Центру (World Radiation Center, WRC) та інші.

Використання даних від наземних станцій дозволяє уникнути неточності при розрахунку GHI. До найбільш вживаних наборів даних належать:

- національна база даних сонячної радіації NSRDB;
- оцінка ресурсів сонячної та вітрової енергії (Solar and Wind Energy Resource Assessment, SWERA);
- архів світового енергетичного балансу (Global Energy Balance Archive, GEBA).

4.1.5 POA випромінення

Важливим фактором при обчисленні кінцевої потужності електроенергії, отриманої від фотоелектричного модуля, є його орієнтація відносно Сонця. В цьому

випадку GHI випромінення, що потрапляє на активну поверхню, називають Plane-of-Array (POA) випроміненням.

Позиціонування фотоелектричних елементів визначається за допомогою двох параметрів (рис. 4.4):

- нахил панелі (tilt) — вимірюється в градусах відносно горизонтальної поверхні;
- азимут панелі (azimuth) — вимірюється в градусах відносно півночі.

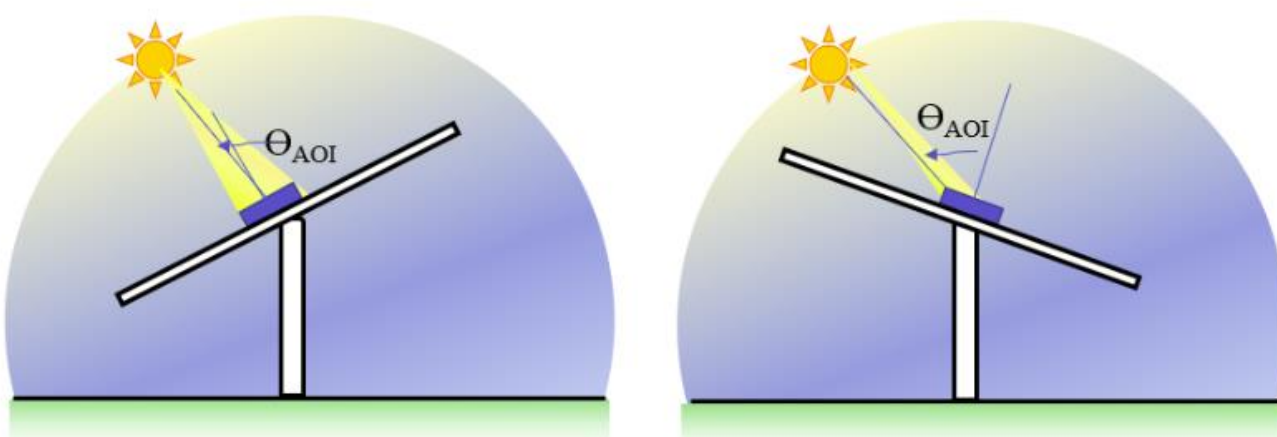


Рисунок 4.4 – POA випромінення

4.2 Опис компоненту «Electricity Storage»

Системи збереження електроенергії (Energy Storage System, ESS), відповідно до кінцевої мети використання, фізико-хімічних та технологічних властивостей суттєво різняться. У роботі [37] виділяють такі види систем зберігання електроенергії:

- гідроакумуючі (Pumped-hydro);
- електрохімічні (Electrochemical);
- термальні (Thermal);
- електромеханічні (Electromechanical).

Доля використання кожної в світі наведено на рис. 4.5.

Storage Technology	Installed Capacity (GW)	% Share
2016 World Total	168.6	100
Pumped-hydro	162.2	96.2
Electrochemical	1.6	0.95
Thermal	3.2	1.9
Electromechanical	1.6	0.95

Рисунок 4.5 – Доля введених в експлуатацію ESS [37]

4.2.1 Вимоги до системи накопичення

Для активного використання відновлювальної енергетики, ESS повинні задовольняти таким вимогам:

- висока ємність (high capacity);
- швидкість заряду/розряду (rapid charge/discharge rates);
- висока щільність енергії (high energy density);
- тривалий термін служби (long cycle life);
- стабільна працездатність і продуктивність (stable operation and performance).

4.2.2 Порівняльна характеристика

Порівняльна характеристика технологій систем збереження електроенергії проводиться в контексті кінцевої мети використання, зокрема: арбітраж

електроенергії, регулювання частоти, резервування для екстрених ситуацій, зміщення кривої навантаження та інших [38].

При побудові порівняльного аналізу ми розглядатимемо лише ті параметри, які важливі для вирішення задач арбітражу електроенергії та зміщенню кривої навантаження.

У роботі [38] проведено глибокий порівняльний аналіз різних технологій ESS. Таблиця із технічними параметрами наведено на рис. 4.6.

Systems	Energy and power density				Life time and cycle life		Influence on environment	
	Wh/kg	W/kg	Wh/L	W/L	Life time (years)	Cycle life (cycles)	Influence	Description
PHS	0.5–1.5		0.5–1.5		40–60		Negative	Destruction of trees and green land for building the reservoirs
CAES	30–60		3–6	0.5–2.0	20–40		Negative	Emissions from combustion of natural gas
Lead-acid	30–50	75–300	50–80	10–400	5–15	500–1000	Negative	Toxic remains
NiCd	50–75	150–300	60–150		10–20	2000–2500		
NaS	150–240	150–230	150–250		10–15	2500		
ZEBRA	100–120	150–200	150–180	220–300	10–14	2500+		
Li-ion	75–200	150–315	200–500		5–15	1000–10,000+		
Fuel cell	800–10,000	500+	500–3000	500+	5–15	1000+	Negative	Remains and/or combustion of fossil fuel
Metal-Air	150–3000		500–10,000			100–300	Small	Little amount of remains
VRB	10–30		16–33		5–10	12,000+	Negative	Toxic remains
ZnBr	30–50		30–60		5–10	2000+		
PSB	–	–	–	–	10–15			
Solar fuel	800–100,000		500–10,000		–	–	Benign	Usage and storage of solar energy
SMES	0.5–5	500–2000	0.2–2.5	1000–4000	20+	100,000+	Negative	Strong magnetic fields
Flywheel	10–30	400–1500	20–80	1000–2000	~15	20,000+	Almost none	
Capacitor Super-	0.05–5	~100,000	2–10 capacitor	100,000+ 2.5–15	~5 500–5000	50,000+ 10–30	Small	Little amount of remains 100,000+
20+		100,000+	Small	Little amount of remains				

Рисунок 4.6 – Порівняння технічних характеристик технологій ESS [38]

Аналізуючи дані з рис. 4.6, виділимо список технологій, що мають велику кількість циклів розрядження та зарядження, оскільки проведення арбітражу електроенергії в контексті мікромережі передбачає активне використання ESS. Величина цього параметру безпосередньо впливає на довговічність системи накопичення.

Інший параметр, який важливий для проведення арбітражу електроенергії, — це тривалість розрядження. Період часу, протягом якого можна продати електроенергію за найвищою ціною, триває від 1-3 годин.

У роботі [37] наведено графічну візуалізацію порівняння за цим параметром (див. рис. 4.7).

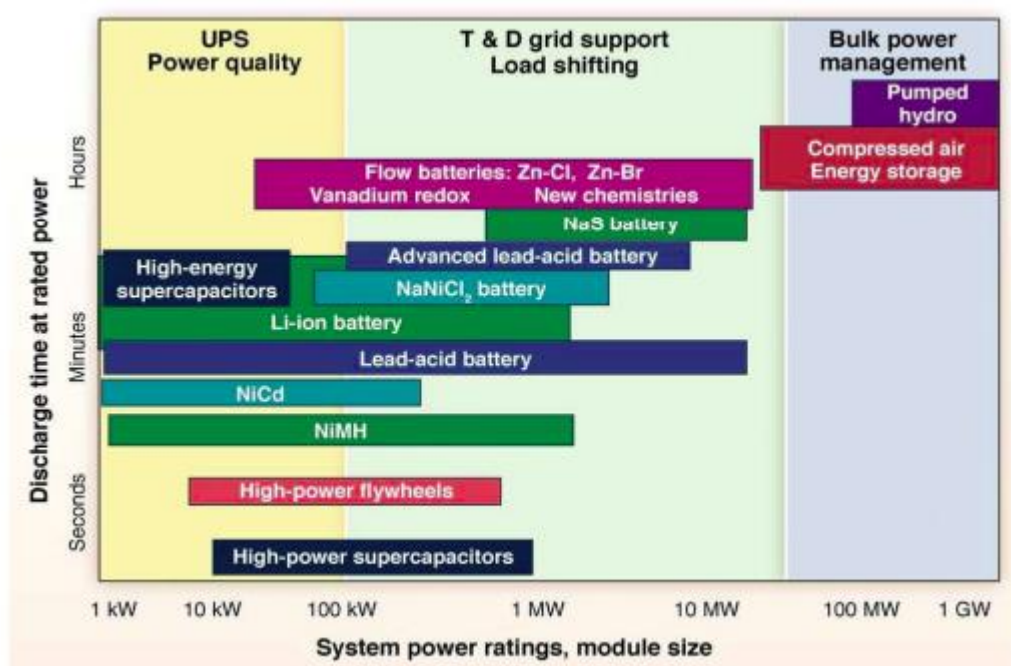


Рисунок 4.7 – Тривалість розрядження [37]

Підсумовуючи, можна виділити Li-ion, NiCd та NaS технології накопичення електроенергії. Всі вони мають велику кількість циклів розрядження, можливість швидкого розрядження та не шкодять екології.

4.2.3 Модель SoC

В літературі [39] широко прийнято моделювати процес роботи компонентів накопичення електроенергії, використовуючи лише рівень заряду (State of Charge, SoC). SoC визначає рівень збереженою енергії в межах від 0 до 100% відносно

номінальної ємності накопичувача. Алгебраїчне представлення такої моделі визначається таким чином:

$$SOC(t) = SOC(t-1) - \int_{t-1}^t \frac{I_{Batt}}{C_{Batt}} dt, \quad (4.2)$$

де C_{Batt} — номінальна ємність накопичувача, I_{Batt} — сила струму, $[t, t-1]$ період часу розрядження.

4.2.4 Модель DFN

Альтернативою для SoC моделі, за умови використання Li-ion накопичувачів, розглядають модель Дойл-Фуллер-Ньюмана (Doyle-Fuller-Newman, DFN) [40]. Цю модель іноді називають псевдодвовимірною моделлю або просто моделлю Ньюмана.

Модель складається з набору сильно зв'язаних нелінійних параболічних та еліптичних диференціальних рівнянь у частинних похідних. В літературі цю систему рівнянь розв'язують чисельними методами, включаючи скінченно-різницеві методи, методи контрольних об'ємів, методи скінченних елементів та ортогональної колокації.

4.3 Опис компоненту «Decision-Making Module»

4.3.1 Види контролю мікромережі

У роботі [41] виділяють три підходи до прийняття рішень в межах мікромережі:

- централізований — інформація про стан кожного із компонентів мікромережі потрапляє до центрального блоку прийняття рішень (контролера), аналізується, визначається перелік інструкцій;
- децентралізований — один або множина компонентів мають власний контролер. Інформація про стан інших компонентів цьому блоку прийняття рішень невідома;
- розподілений — один або множина компонентів мають власний контролер, при цьому контролери можуть обмінюватися інформацією про рішення між собою.

4.3.2 Діаграма діяльності

Формалізуємо роботу компоненту «Decision-Making Module» у вигляді деталізованої діаграми діяльності (рис. 4.8). При цьому будемо використовувати децентралізований підхід контролю, розділений між двома агентами.

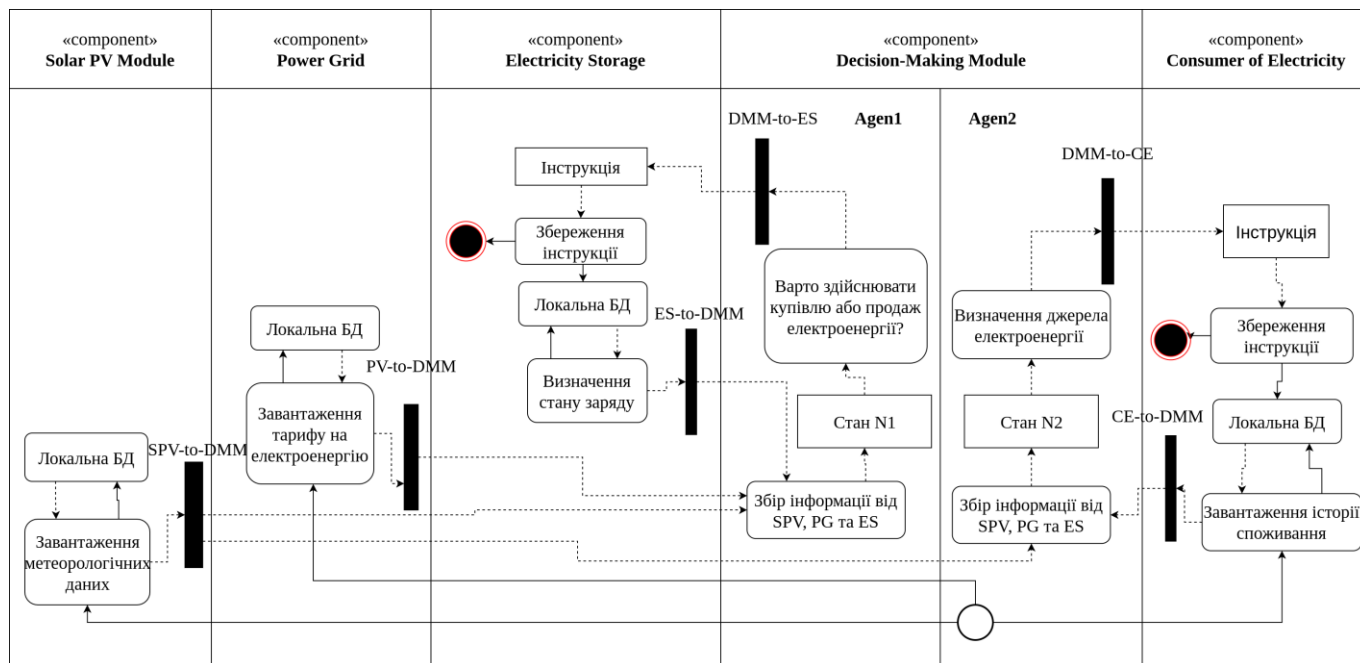


Рисунок 4.8 – Деталізована діаграма діяльності
компоненту «Decision-Making Module»

4.3.3 Алгоритм навчання агентів

Процес навчання кожного з агентів формалізується за допомогою математичного апарату навчання з підкріпленням [42].

Агент послідовно взаємодіє з оточенням, в якому він існує. На відповідний стан (навколишні умови) агент здійснює дію, яка винагороджується або штрафується. Формально, процес прийняття рішень формулюється як кортеж виду $\langle S, A, R, P, \rho_0 \rangle$, де:

- S – множина всіх можливих станів. Спостереження s_t має вигляд кортежу виду $[X_m, X_p, X_c, X_b]$, де X_m – метеорологічні значення за попередні періоди часу; X_p – вартість електроенергії; X_c – значення споживання електроенергії в попередні періоди часу; X_b – внутрішні характеристики накопичувача електроенергії;
- A – множина всіх можливих дій для агента: купити, продати або зачекати;
- $R: S \times A \times S \rightarrow R$ – функція винагороди в грошовому еквіваленті. Різниця в грошовому еквіваленті за куплену і продану електроенергію до мережі;

- $P: S \times A \rightarrow P(S)$ – функція ймовірності переходу, де $P(s' | s, a)$ – ймовірність переходу в стан s' , якщо в стані s зроблено вибір a ;
- ρ_0 – початковий розподіл станів.

Задача навчання формулюється як пошук стратегії (policy) – набору правил, які використовує агент для вибору дії. Позначається таким чином: $a_t \sim \pi(\cdot | s_t)$. Параметризована стратегія $a_t \sim \pi_\theta(\cdot | s_t)$ представлена нейронною мережею з параметрами θ , які потрібно дізнатися [42].

Оптимізаційна проблема RL [42] зводиться до пошуку стратегії, що максимізує очікувану винагороду:

$$\pi^* = \arg \max_{\pi} J(\pi) , \quad (4.3)$$

де $J(\pi)$ — сумарна винагорода протягом всього періоду взаємодії (траєкторії).

Винагорода в межах однієї траєкторії визначається як:

$$J(\pi) = \sum_{t=0}^{\infty} \gamma^t r_{\pi}^t , \quad (4.4)$$

де γ^t – коефіцієнт дисконту, що приймає значення на інтервалі $(0, 1)$; r_{π}^t – винагорода, отримана для дії a_t , яка визначається таким чином: $r_{\pi}^t = R(s_t, a_t, s_{t+1})$.

Пошук оптимальних параметрів стратегії здійснюється згідно одного з алгоритмів навчання. Ключова особливість RL виражається в тому, що навіть незначна зміна параметрів стратегії веде до того, що винагорода суттєво змінюється.

Будемо застосовувати алгоритм Trust Region Policy Optimization (TRPO) [43], який дозволяє досягти монотонного покращення ефективності стратегії. Ідея алгоритму полягає в тому, що параметри стратегії на ітерації $k+1$ не суттєво відрізняються від ітерації k .

Ітераційний процес пошуку параметрів стратегії виглядає таким чином:

$$\begin{aligned} \theta_{k+1} &= \arg \max_{\theta} L(\theta_k, \theta) \\ \text{за умови } D_{KL}(\theta \parallel \theta_k) &\leq \sigma, \end{aligned} \quad (4.5)$$

де $\sigma = \text{const}$; $L(\theta_k, \theta)$ визначає як веде себе π_{θ} щодо попередньої стратегії:

$$L(\theta_k, \theta) = E_{s,a \sim \pi_{\theta_k}} \left[\frac{\pi_{\theta}(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a) \right], \quad (4.6)$$

де $A^{\pi_{\theta_k}}(s, a)$ оцінює наскільки кращою є дія a в стані s , щодо випадкової дії $\pi(\cdot | s)$.

Обмеження (див. 4.5) визначається як середнє значення відстані Кульбака — Лейблера (Kullback–Leibler divergence, KL) між двома дискретними розподілами для стратегій $\pi_{\theta}(\cdot | s)$ та $\pi_{\theta_k}(\cdot | s)$:

$$D_{KL}(\theta \parallel \theta_k) = E_{s \sim \pi_{\theta_k}} \left[D_{KL}(\pi_{\theta}(\cdot | s) \parallel \pi_{\theta_k}(\cdot | s)) \right]. \quad (4.7)$$

На практиці знайти точний розв’язок (4.5) складно, тому використовується лінійна та квадратична апроксимація для виразів (4.6 та 4.7) відповідно. Тоді вираз 4.5 матиме такий вигляд:

$$\begin{aligned} \theta_{k+1} &= \arg \max_{\theta} g^T(\theta - \theta_k) \\ \text{за умови } \frac{1}{2} (\theta - \theta_k)^T H (\theta - \theta_k) &\leq \sigma, \end{aligned} \quad (4.8)$$

де g^T — градієнт; H — матриця Гесе.

4.4 Опис компонентів «Power Grid» та «Consumer of Electricity»

Головними особливостями компонентів Power Grid та Consumer of Electricity є необмежене надання електроенергії до мікромережі та формування запитів на її отримання відповідно.

Формалізуємо роботу компонентів Power Grid та Consumer of Electricity з іншими у вигляді деталізованих діаграм діяльності (рис. 4.9) та (рис.4.10) відповідно.

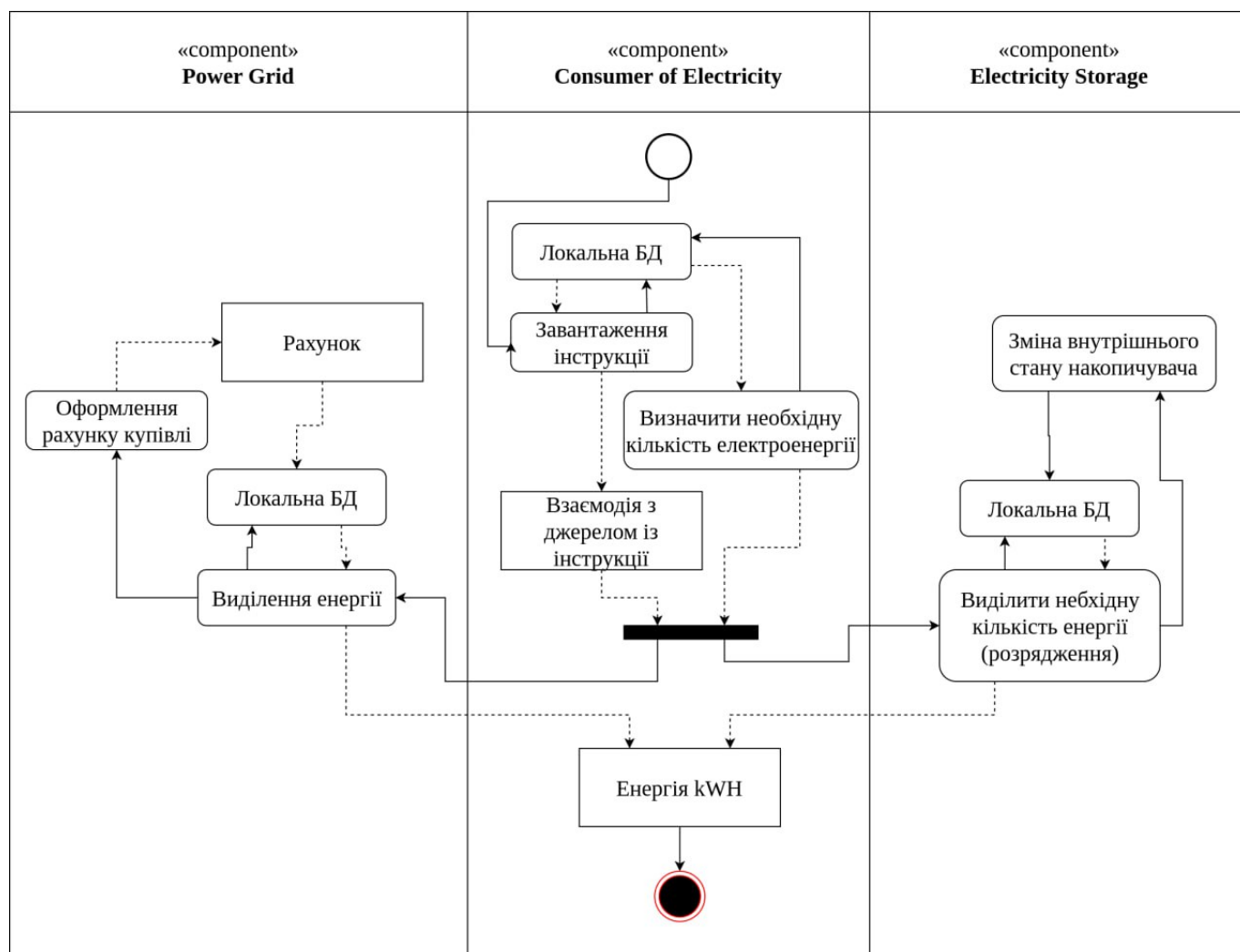


Рисунок 4.9 – Взаємодія компоненту «Consumer of Electricity» з іншими компонентами

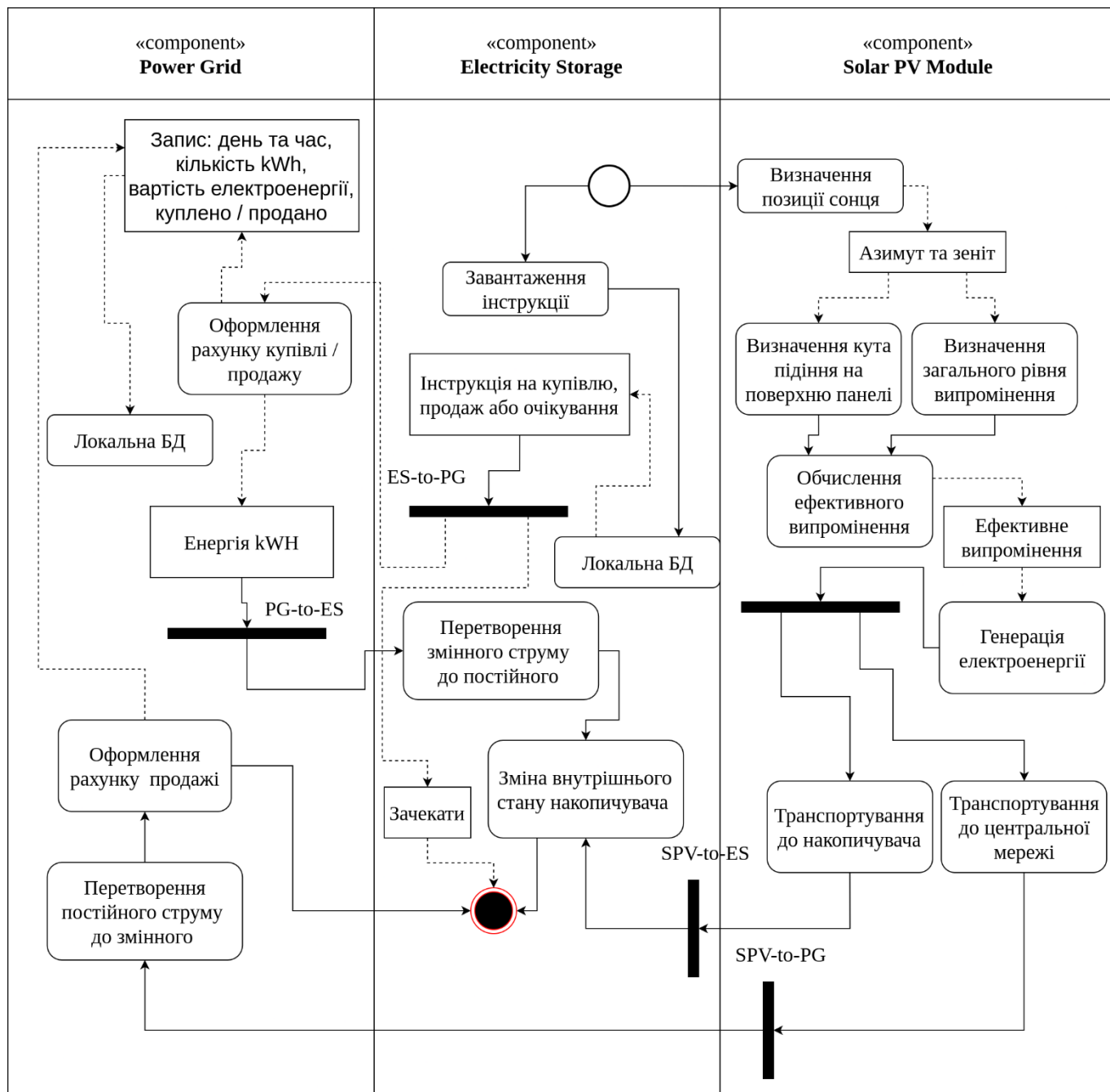


Рисунок 4.10 – Взаємодія компоненту «Electricity Storage» з іншими компонентами

4.5 Висновки до розділу 4

В розділі 4:

- 1) було формалізовано математичну модель одного діода та алгоритма MPPT;

- 2) проаналізовано ринок технологій системи накопичення електроенергії і вибрано модель Li-ion накопичувача;
- 3) формалізовано мультиагентну систему прийняття рішень у вигляді деталізованої діаграми діяльності. Наведено основні поняття алгоритмів навчання з підкріпленням.

5 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

5.1 Середовище

Важливим моментом є синхронізація даних про сонячне випромінення, метеорологічних даних, споживання електроенергії індивідуальним будинком та її вартості відносно однієї геолокації та часу.

Час дискретизації подій повинен бути невеликим, оскільки мінливість погоди і сонячного випромінення вагомо буде впливати на процес генерації електроенергії фотоелектричним модулем.

Як геолокацію було обрано штат Вірджинія (VA). У ньому щільність наземних метеорологічних станцій найвища в порівнянні з іншими штатами.

Геолокація будинку:

- широта та довгота відповідно (38.97203, -77.4869);
- висота над рівнем моря: 98 м.

Часовий пояс: ETC / GMT-5.

Незалежний системний оператор (Independent System Operator, ISO) встановлює вартість електроенергії від центральної мережі. За регіон Dominos обраного будинку відповідає PJM оператор.

5.2 Валідація даних

5.2.1 Дані про сонячне випромінення

Дані від наземної станції Starling мають роздільну здатність 1 хв. Період даних включає спостереження за весь 2018 рік.

Аналіз спектральної діаграма випромінення (рис. 5.1) дозволяє виявити, що деякі спостереження мають пропуски. Наявні дані були згруповані відносно години

дня протягом всього року, а знайдені медіанні значення були використані для заповнення пропусків.

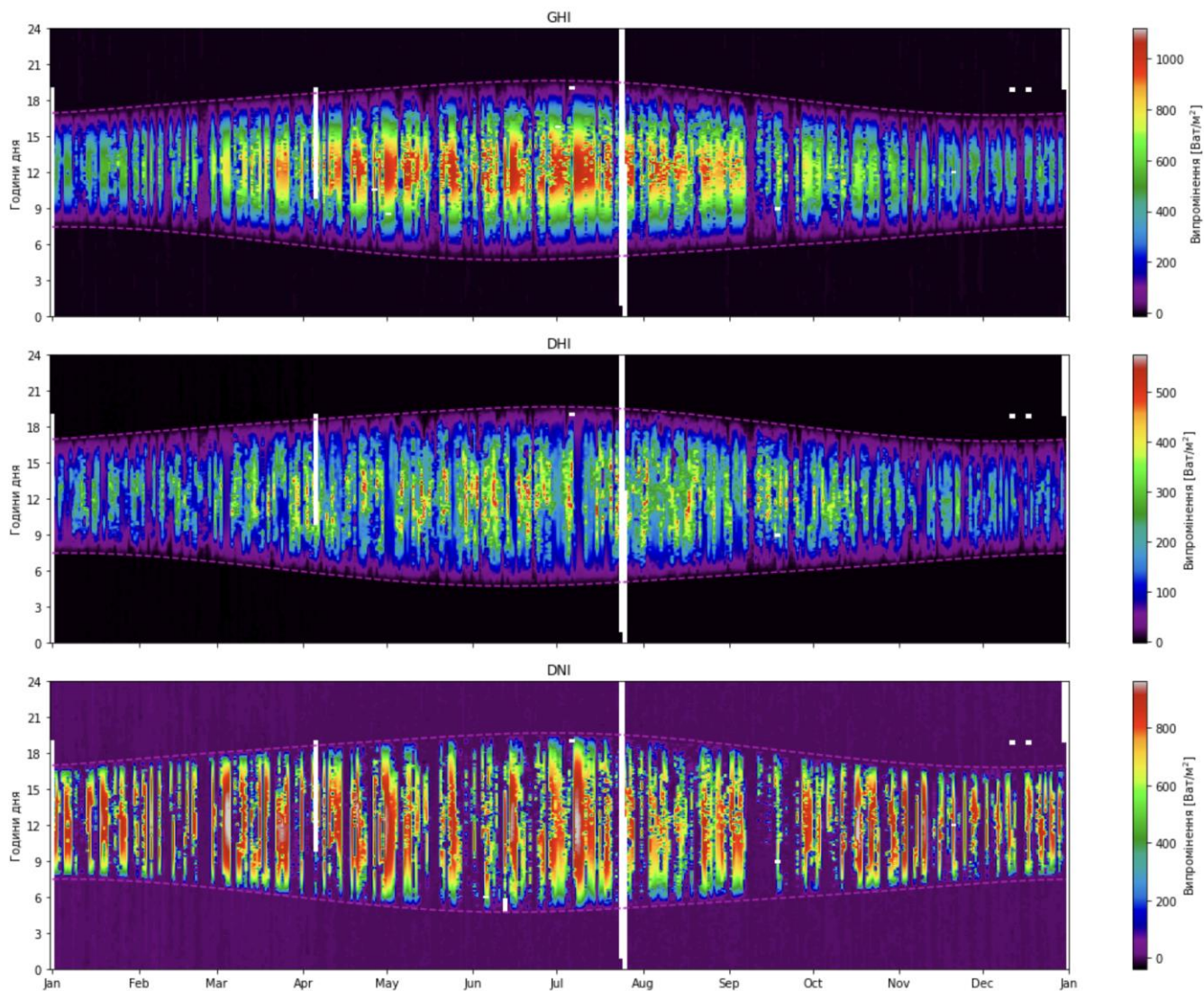


Рисунок 5.1 – Спектральна діаграма випромінення

Для перевірки достовірності вимірювань сонячного випромінення було застосовано тест, запропонований Лонгом і Даттоном [44]. Цей тест дозволяє визначити надзвичайно рідкісні спостереження, які можна вважати викидами. Графічне представлення зображено на рис. 5.2

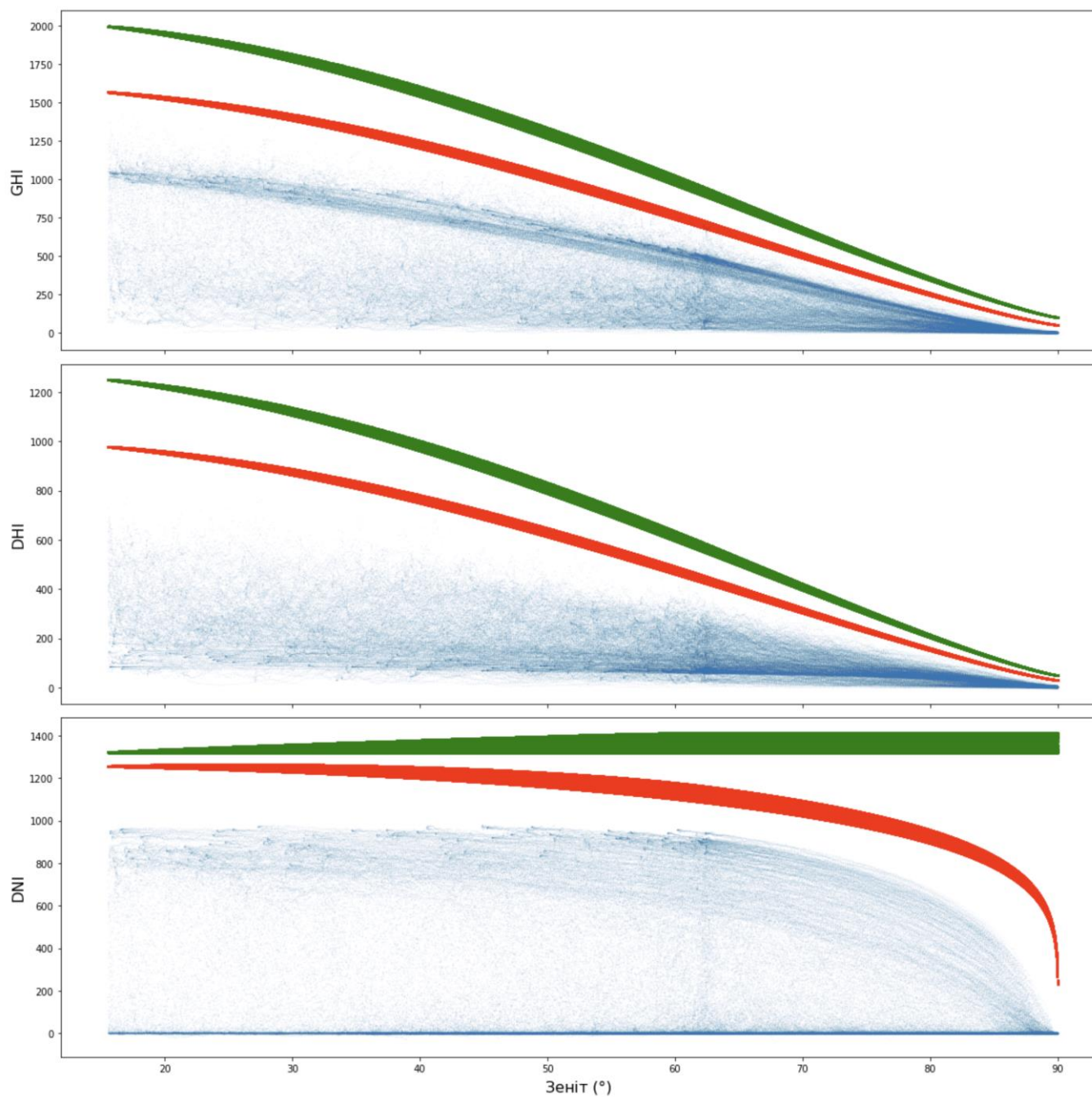


Рисунок 5.2 – Виявлення аномальних значень випромінення

На рисунку 5.2 наведено спостереження сонячного випромінення (синій колір) та дві криві, які визначають фізичну межу при якій спостереження зустрічаються дуже рідко (червона крива) та в принципі неможливі (зелена крива). Тому можна стверджувати, що викиди в спостереженні відсутні.

5.2.2 Дані про вартість електроенергії

RJM оператор надає вартість електроенергії з роздільною здатністю в 1 годину. Графічне представлення даних зображено на рис. 5.3.

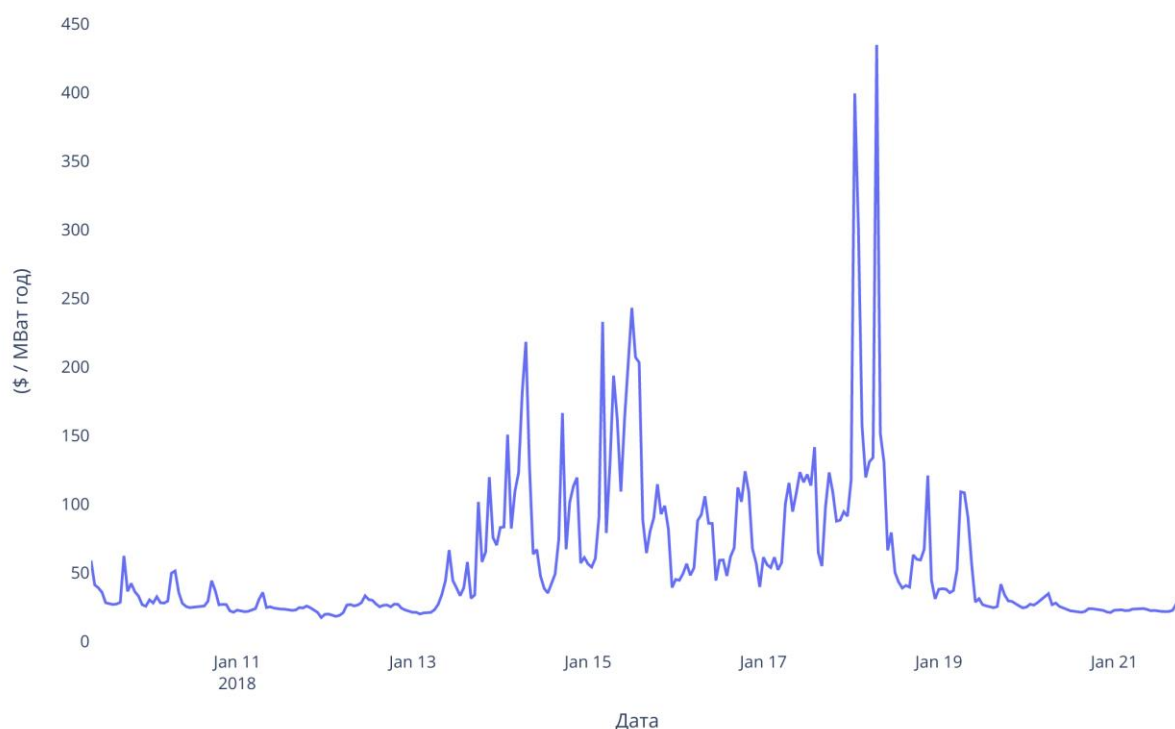


Рисунок 5.3 – Вартість електроенергії в Dominos регіоні

5.2.3 Дані про споживання електроенергії будинком

За ініціативи міністерства енергетики США було реалізовано проект End-Use Load Profiles for the U.S. Building Stock [45]. Задача проекту передбачала збір даних про споживання електроенергії окремими комерційними та житловими будівлями для проведення подальших досліджень в сфері енергетики.

Спостереження для одного із будинків відображено на рис. 5.4.

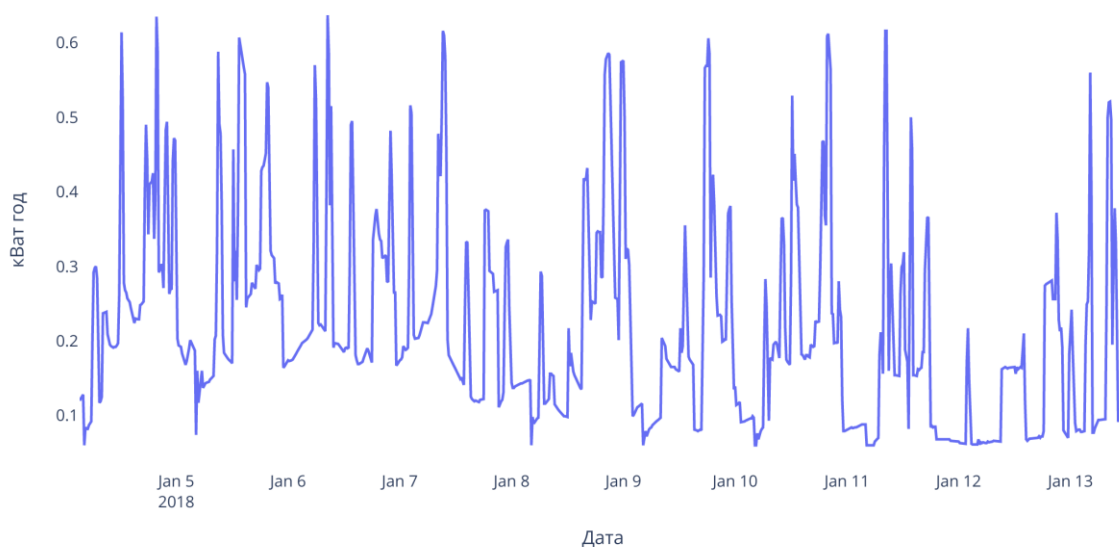


Рисунок 5.4 – Крива споживання електроенергії будинком ID 24487

5.3 Імплементация компоненту «Solar PV Module»

Перетворення сонячного випромінення в електричну енергію відбувається згідно з моделлю одного діода (4.1). Параметри фотоелектричного елемента із фізичними розмірами наведено в табл. 5.1.

Таблиця 5.1 – Фотоелектричний елемент

Технологія	A	R_s (Ом)	I_s (A)	Довжина (см)	Ширина (см)
Моно-с-Si	1,599	0,296856	9,44	1,64	0,975

Паралельне об'єднання 60 діодів дозволяє отримати результуючу напругу в 32 В.

Визначення азимуту та нахилу панелі було сформовано як оптимізаційну задачу з недиференційованою цільовою функцією двох змінних:

$$\max \int_0^T P_{PV}(azimuth, tilt) dt, \quad (5.1)$$

де P_{PV} – вихідна потужність фотоелектричного модуля; T – період моделювання (весь 2018 рік).

Для пошуку оптимального рішення було обрано алгоритм Нелдера – Міда [46]. Графічне представлення поверхні функціоналу 5.1 наведено на рис. 5.5.

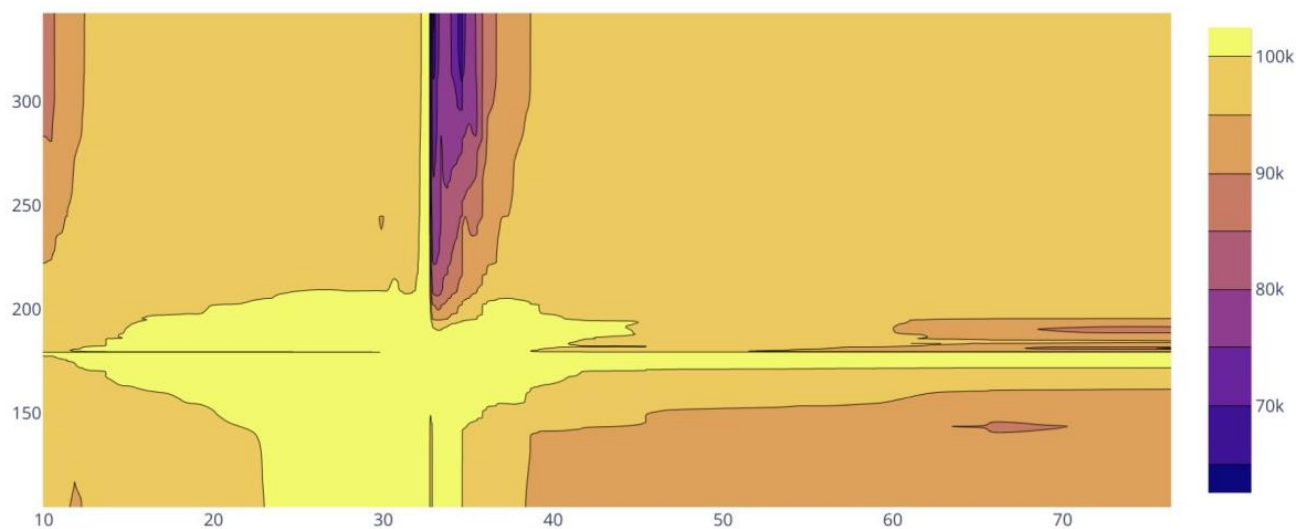


Рисунок 5.5 – Поверхня функціоналу (5.1)

Оптимальні значення азимуту та кута нахилу панелі наведено в табл. 5.2.

Таблиця 5.2 – Позиціонування панелі

Азимут (°)	Кут нахилу (°)
32,4	179

5.4 Імплементация компоненту «Electricity Storage»

Моделювання поведінки накопичувача електроенергії відбувається згідно з моделлю Дойл-Фуллер-Ньюмана. Під час реалізації моделі було важливо отримати не тільки значення збереженої енергії SoC, але й динамічно впливати на процес зарядження і розрядження.

Під час вибору програмного забезпечення, яке використовується для моделювання акумуляторних батарей, було виділено таке:

- PyBamm — пакет моделювання акумуляторних накопичувачів з відкритим вихідним кодом, написаний на Python;
- Cantera — набір інструментів з відкритим кодом для вирішення задач з хімічної кінетики та термодинаміки для моделювання акумуляторних батарей;
- COMSOL Multiphysics — комерційне програмне забезпечення, яке включає електрохімічне моделювання накопичувачів електроенергії.

Кінцевим вибором став пакет моделювання PyBamm. Ключова функціональність, яка закладена в даний програмний пакет, включає:

- середовище для написання і розв'язання систем диференціальних рівнянь;
- колекції моделей акумуляторних батарей, в тому числі DFN;
- наявність засобів для проведення експериментів та візуалізації їх результатів.

Одним із недоліків зазначеного пакету є відсутність гнучкого налаштування конфігурації накопичувача. Іншими словами, можливість конфігурації батареї необхідної ємності відсутня.

Для створення повноцінної моделі Li-ion батареї було реалізовано декілька власних функціональних елементів, які дозволяють об'єднувати одиничні елементи живлення як паралельно, так і послідовно для створення кінцевої конфігурації накопичувача. Додатково було реалізовано інструменти візуалізації та взаємодії з іншими компонентами мікромережі.

Параметри одного елемента живлення наведено в табл. 5.3. Вони залишаються постійними, змінюється лише їх кількість та конфігурація з'єднань:

Таблиця 5.3 — Параметри Li-ion елемента живлення

Номінальна ємність (А•год.)	Нижня межа напруги (В)	Верхня межа напруги (В)	Початковий рівень SoC (%)
5	2,5	4,2	50

Кінцева конфігурація накопичувача підбиралася відповідно до моделей, представлених на ринку. Відповідні параметри наведено в табл. 5.4.

Таблиця 5.4 — Конфігурація компоненту «Electricity Storage»

Номінальна ємність (кВат•год)	Напруга (В)	Кількість паралельних з'єднань	Кількість послідовних з'єднань
5000	24	5	41

Функціональність PyBamm дозволяє взаємодіяти із елементом живлення за допомогою списку інструкцій із визначеними ключовими словами та значеннями величин. Демонстрація взаємодії і перевірка працездатності для тестового експерименту наведена нижче:

1. Discharge at 2 A until 13 V
2. Rest for 2 hour
3. Charge at 1 A until 20.5
4. Discharge at 1 A until 18 V
5. Rest for 1 hour
6. Charge at 2 A until 20.5 V
7. Rest for 1 hour.

Графічне представлення залежності сили струму, напруги і величини заряду накопичувача, що було транспортовано, наведено на рис. 5.6.

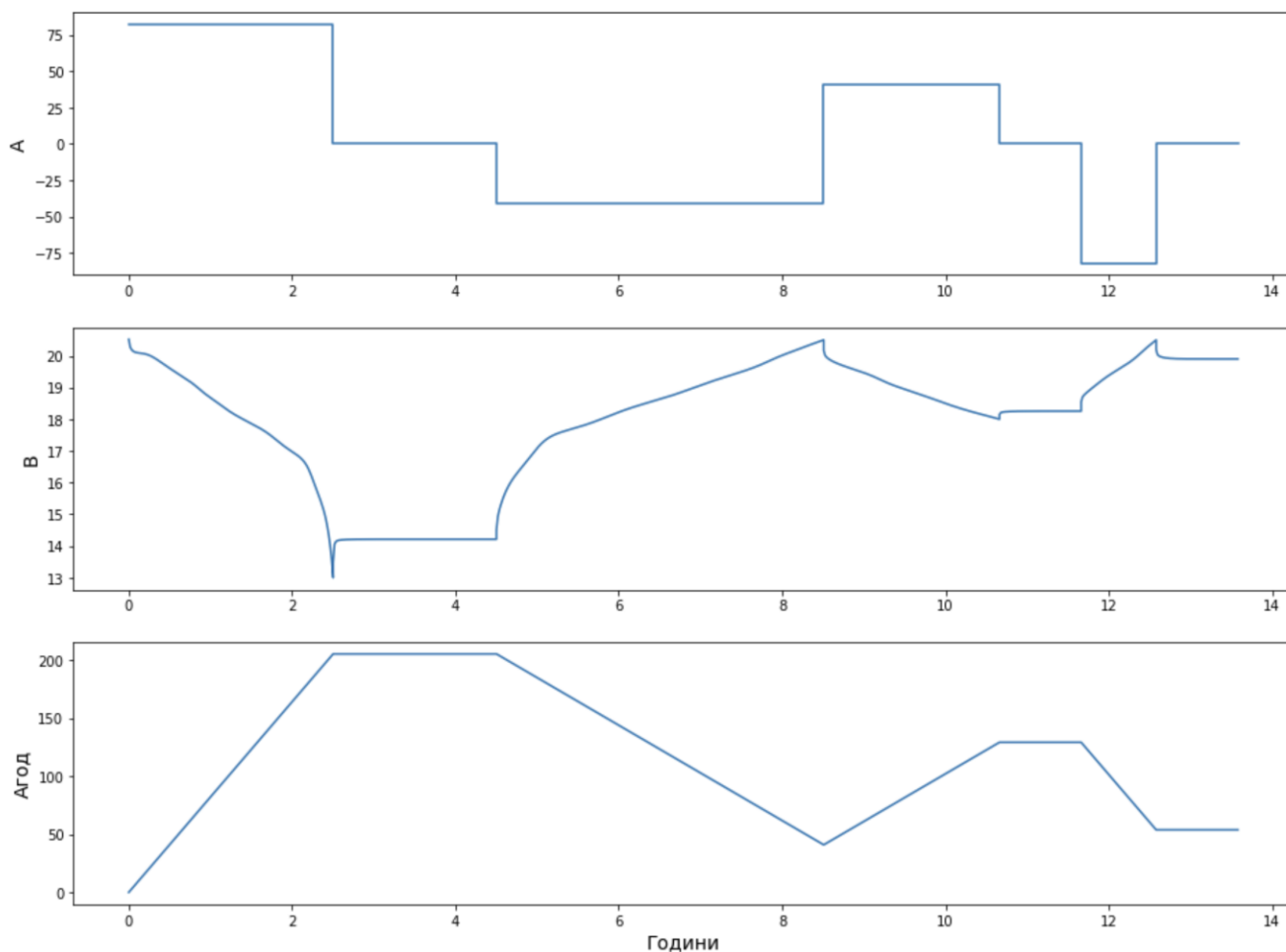


Рисунок 5.6 – Моделювання роботи компоненту «Electricity Storage»

5.5 Процес обробки даних та створення навчального середовища

На роботу компонента «Solar PV Module» стан або взаємодія інших компонентів не впливає. Тому має сенс винести роботу цього компонента в окремий модуль і результати його роботи додати до середовища навчання агентів. Загальний процес створення середовища навчання наведено на рис. 5.7.

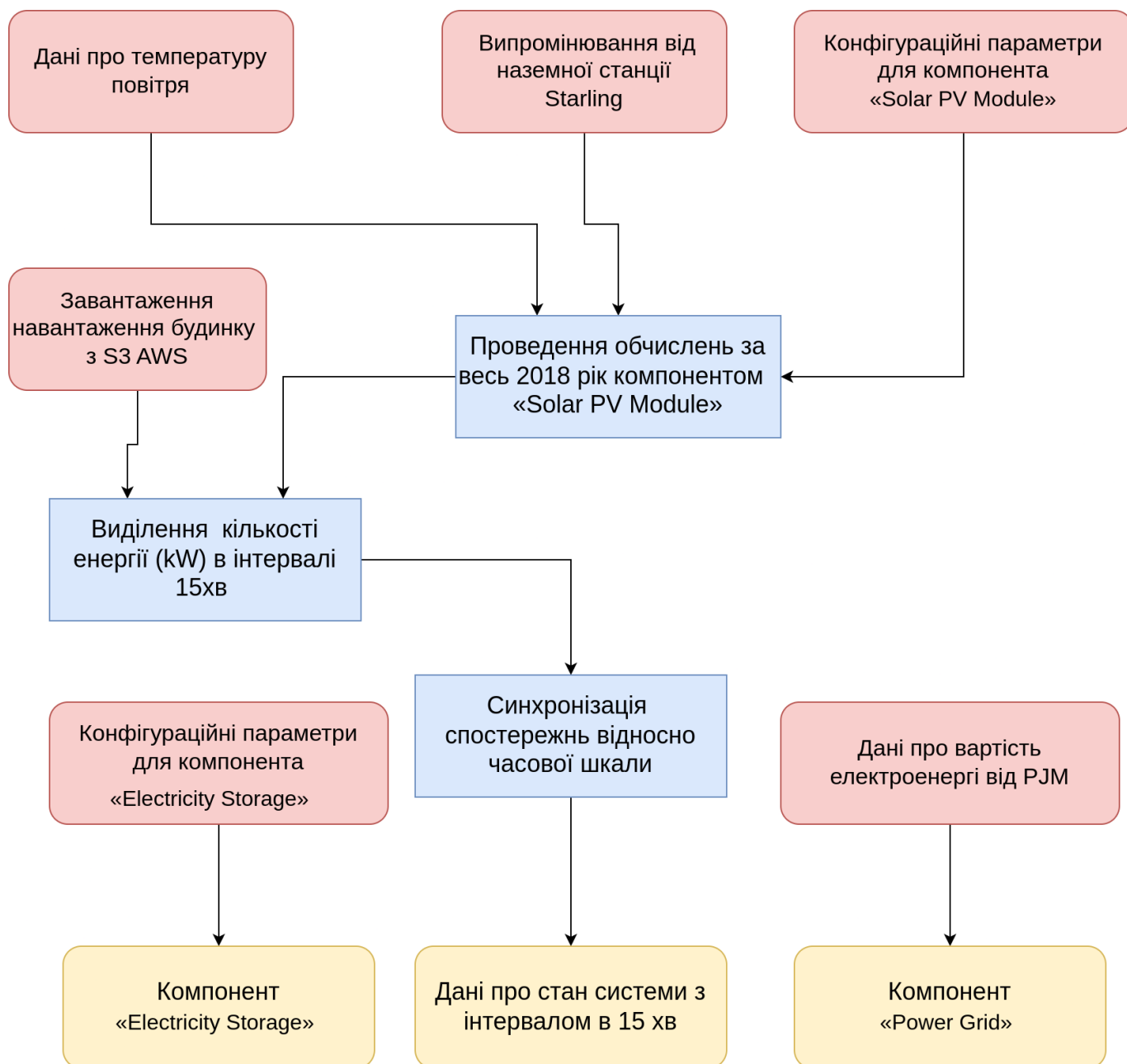


Рисунок 5.7 – Ініціалізація середовища навчання

Кроки обробки та підготовки даних на рис. 5.7 виділено синім фоном, аргументи і дані, що не змінюються, виділено червоним. Кінцеві дані, які відображають стан системи із визначеними активними компонентами мікромережі, виділено жовтим фоном.

5.6 Імплементація компоненту «Decision-Making Module»

Для навчання мультиагентної системи важливо:

- визначити середовище, в якому агенти можуть взаємодіяти з ним. Тут важлива увага приділяється передачі електроенергії між компонентами мікромережі;
- визначити перелік можливих дій, які можуть виконувати агенти;
- визначити порядок отримання винагороди кожним з агентів.

Чітке визначення цих пунктів дозволить знайти оптимальну стратегію, згідно з якою перший агент буде виконувати купівлі / продаж електроенергії, а другий — визначати від якого джерела електроенергії найоптимальніше транспортувати енергію до кінцевого споживача.

5.6.1 Середовище мікромережі

Специфіка роботи середовища мікромережі передбачає реалізацію трьох методів:

- `__init__` – завантаження даних про сонячне випромінення, метеорологічні дані, кількість згенерованої електроенергії фотоелектричним модулем. Також ініціалізується компонент «Electricity Storage» та «Power Grid». Останній підраховує загальну вартість купленої / проданої енергії та її вартість в конкретний період часу;
- `reset` – повертає стан системи до початкового стану. Як вхідний параметр передається номер тижня календарного року;
- `step` – ітератор, який при кожному виклику повертає наступний стан мікромережі: вартість електроенергії та ємність компонента «Electricity Storage» за останні 2 години.

Кожен крок відбувається в розрізі 15 хв, один епізод триває тиждень.

5.6.2 Перелік дій

Для агента N1 визначено три можливі дії:

- 0 – купити електроенергію від компоненту «Power Grid»;
- 1 – продати електроенергію до компоненту «Power Grid»;
- 2 – зачекати.

Величина купленої і проданої енергії визначається згідно з номінальним струмом заряду і розряду (c-rates). Конкретне значення було встановлено на рівні 0,1 номінальної ємності.

Для агента N2 визначено дві можливі дії:

- 0 – при наступному кроці використовувати електроенергію від компоненту «Electricity Storage»;
- 1 – при наступному кроці використовувати електроенергію від компоненту «Power Grid».

5.6.3 Винагорода

Виділяють два основні типи винагород (рис. 5.8):

- рідкісні винагороди (sparse reward) – винагорода підраховується лише після досягнення певної проміжної мети або завершення роботи. В контексті роботи мікромережі підраховується загальний рахунок за куплену і продану електроенергію на останньому кроці епізоду (по завершенню одного тижня);
- постійні винагороди (dense reward) – винагорода підраховується на кожному кроці. В цьому випадку підраховується різниця між вартістю купленої і проданої електроенергії для останнього кроку середовища (15 хв.).

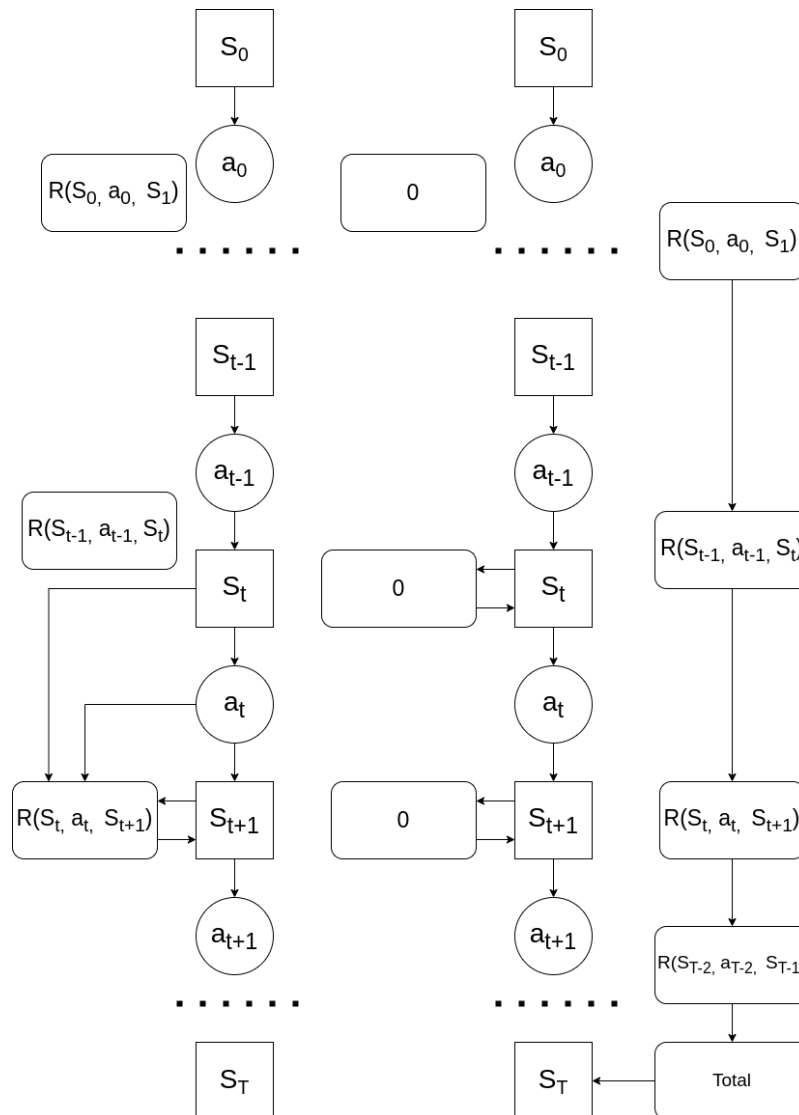


Рисунок 5.8 – Постійна та рідкісна винагороди відповідно

Оцінити доцільність кожного із цих підходів можна лише під час навчання агентів і перевірки остаточних результатів.

5.6.4 Опис інструментів

Процес навчання мультиагентної системи було реалізовано за допомогою бібліотеки Rllib [47]. Ця бібліотека належить до екосистеми Ray – фреймворку розподілених обчислень, який дозволяє легко проводити масштабування, отримуючи приріст швидкодії при виконанні обчислень різного порядку.

Велика кількість впроваджених алгоритмів навчання з підкріпленням дозволяє не зациклюватися на реалізації алгоритмів, а безпосередньо використовувати інтерфейси взаємодії для навчання та використання у власних цілях.

5.7 Висновки до розділу 5

В розділі 5:

- 1) було проведено валідацію спостережень сонячного випромінювання від наземної станції. Було виявлено пропуски в спостереженнях, аномальних спостережень, виявлено не було;
- 2) реалізовано середовище мікромережі для навчання агентів;
- 3) визначено різні механізми виплати винагород.

6 ВЕРИФІКАЦІЯ ТА ВАЛІДАЦІЯ СИСТЕМИ КУПІВЛІ / ПРОДАЖУ ЕЛЕКТРОЕНЕРГІЇ

6.1 Формування тестової вибірки

Основні функції системи купівлі / продажу електроенергії:

- вибір часового проміжку або стану системи, при якому вигідно купити або продати електроенергію; якщо дана ситуація не настає, варто просто зачекати;
- вибір джерела електроенергії, яке буде використовуватися в наступні 15 хв. споживачем електроенергії.

Проміжок часу, протягом якого відбувається взаємодія агентів із мікромережею, дорівнює одному тижню. Цей інтервал був вибраний через такі причини:

- шаблони споживання електроенергії мають схожу поведінку протягом тижня;
- необхідно розділити рік на менші періоди для навчання агентів;
- можна розділити набори наявних даних для проведення валідації роботи системи, нехтуючи першим і останнім тижнем року.

Узагальнюючи, 2018 календарний рік було розділено на п'ятдесят рівних інтервалів, в межах яких зручно вести підрахунок кількості купленої та проданої електроенергії.

Відношення тестових тижнів до навчальних складає 1 до 3. Відбір тижнів для кожної із груп відбувався випадковим чином.

6.2 Представлення отриманих результатів

Вже під час навчання агентів можна стежити за кривою середнього прибутку. Графічне представлення для різних алгоритмів навчання наведено на рис. 6.1.

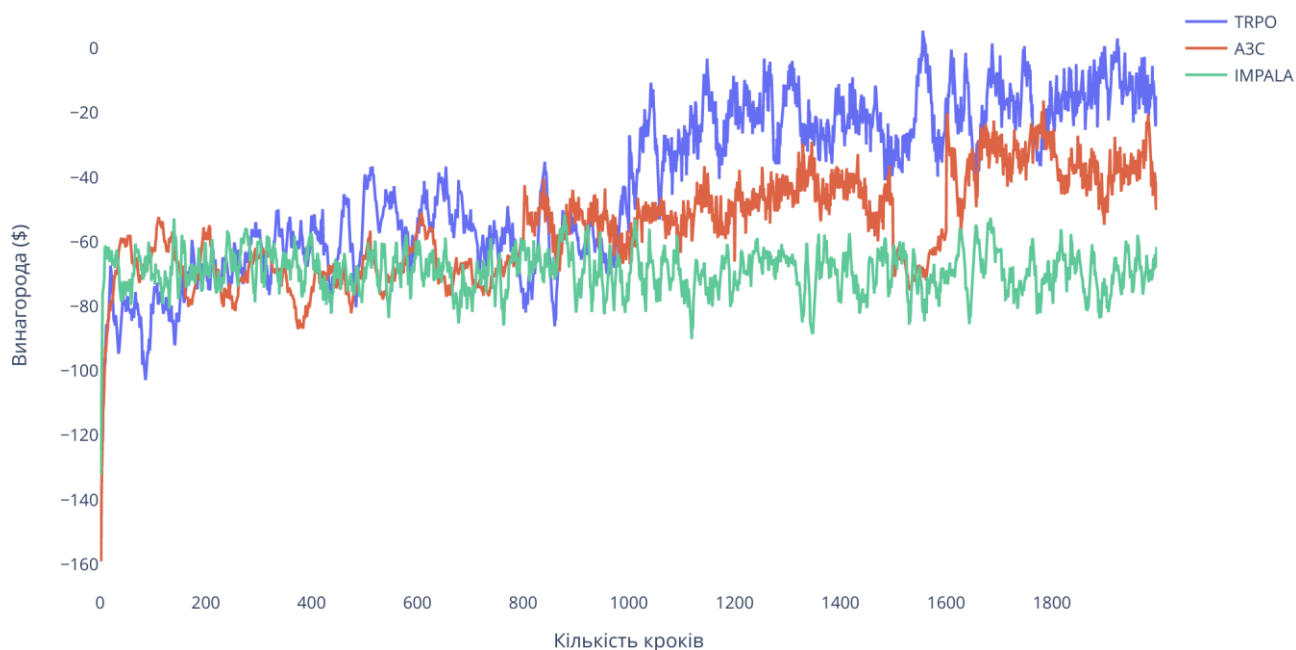


Рисунок 6.1 – Крива винагород для агентів

Аналізуючи рис. 6.1, можна стверджувати:

- використання TRPO алгоритму показує найкращу динаміку приросту винагороди;
- що множини характеристик, які були використані для опису стану системи, достатньо для навчання агентів.

Інший важливий крок передбачає перевірку працездатності системи на тестових тижнях. Оскільки загальна кількість спостережень невелика, то існує ймовірність перенавчання.

Представимо рахунок за електроенергію, яку повинен сплатити користувач для тестових і навчальних тижнів, за допомогою діаграми Voxplot (рис.6.2).

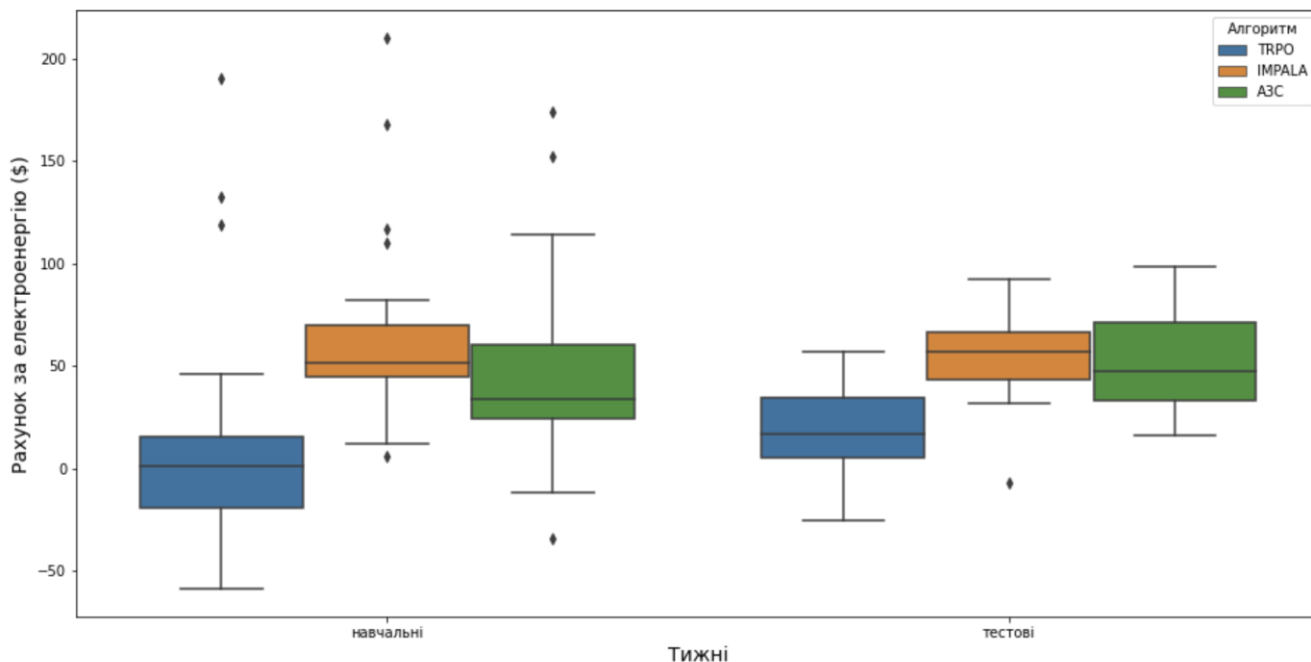


Рисунок 6.2 – Рахунок за електроенергію для навчальних і тестових тижнів

Характер розподілу дозволяє стверджувати, що:

- розмір оплати для навчальних тижнів менший, ніж для тестових;
- найменшим рахунок за електроенергію є для TRPO алгоритму;
- 50-ти процентний квантильний інтервал для навчальних тижнів не знаходиться в одній величині із тестовими тижнями.

Оперуючи цими фактами, однозначно сказати, що агенти перенавчилися і запам'ятали лише навчальні дані, не можна. Підтвердженням цьому можуть слугувати результати, представлені на рис. 6.3. В цьому випадку всі дії в рамках мікромережі виконувалися випадковим чином.

Узагальнюючи, робимо висновки:

- процес навчання агентів відбувається успішно;
- представлення стану мікромережі в конкретний період часу, використовуючи кількісні характеристики за минулі дві години, достатньо для виділення корисних шаблонів агентами.

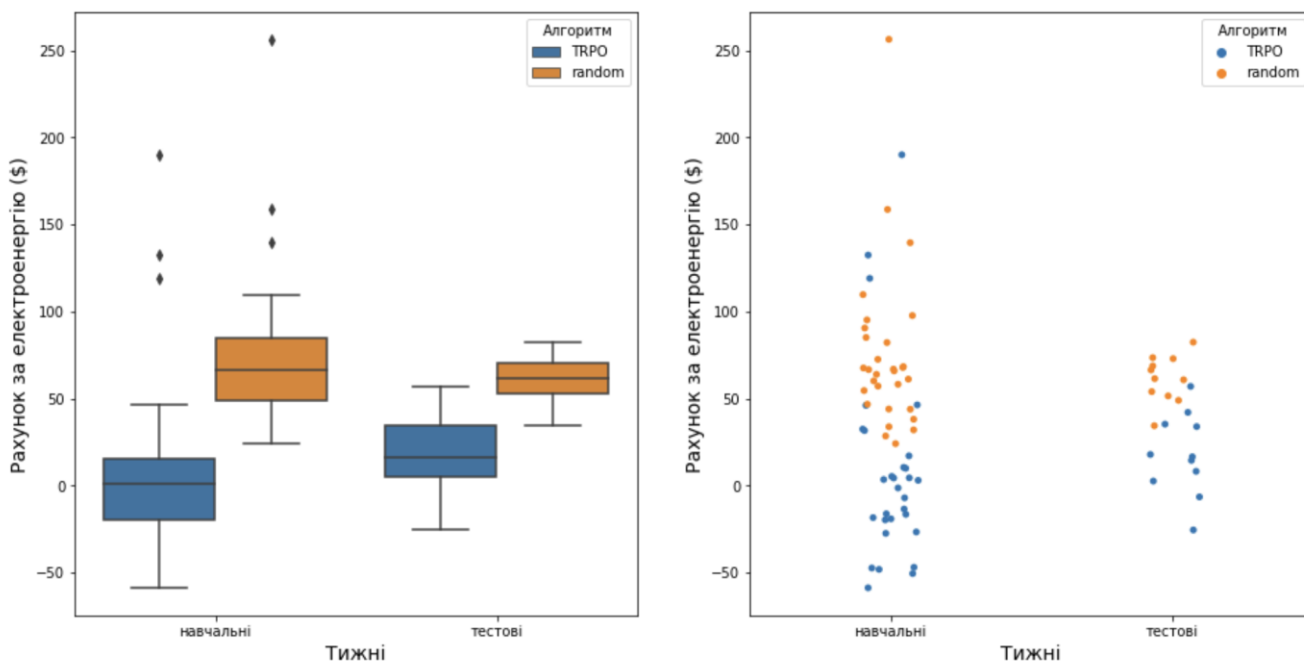


Рисунок 6.3 – Величина рахунку за електроенергію для навчених агентів та випадково зроблених дій

6.3 Порівняння ефективності стратегій винагородження

В підрозділі 5.5.3 було виділено дві стратегії винагородження агентів:

- на кожному кроці (dense);
- по завершенню періоду роботи (закінчення тижня) (sparse).

Порівняємо отримані результати для кожної із винагород і встановимо, чи цей фактор впливає на результативність навчання агентів. Результати наведено на рис. 6.4.

Для повної впевненості дізнаємося чи існує статистично значима відмінність між двома стратегіями. Будемо розглядати лише навчальні тижні, оскільки для них кількість спостережень більша. Наведемо QQ-plot візуалізацію та гістограму розподілу.

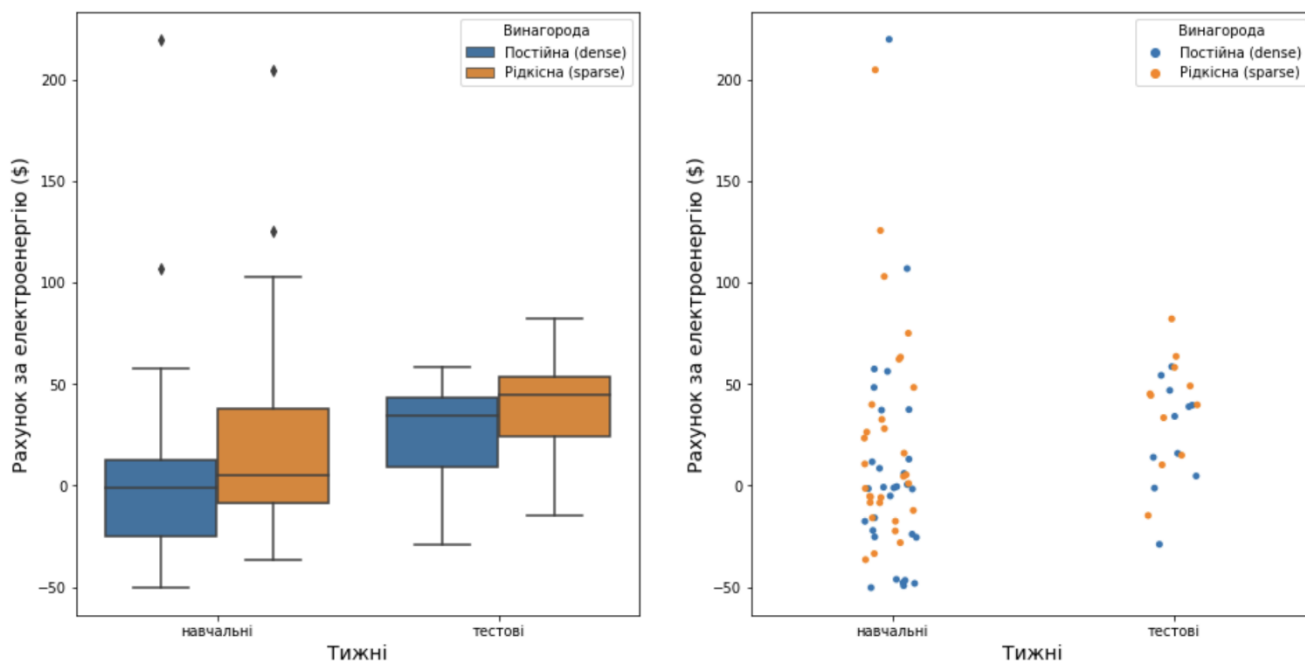


Рисунок. 6.4 – Величина рахунку за електроенергію для різних стратегій винагородження

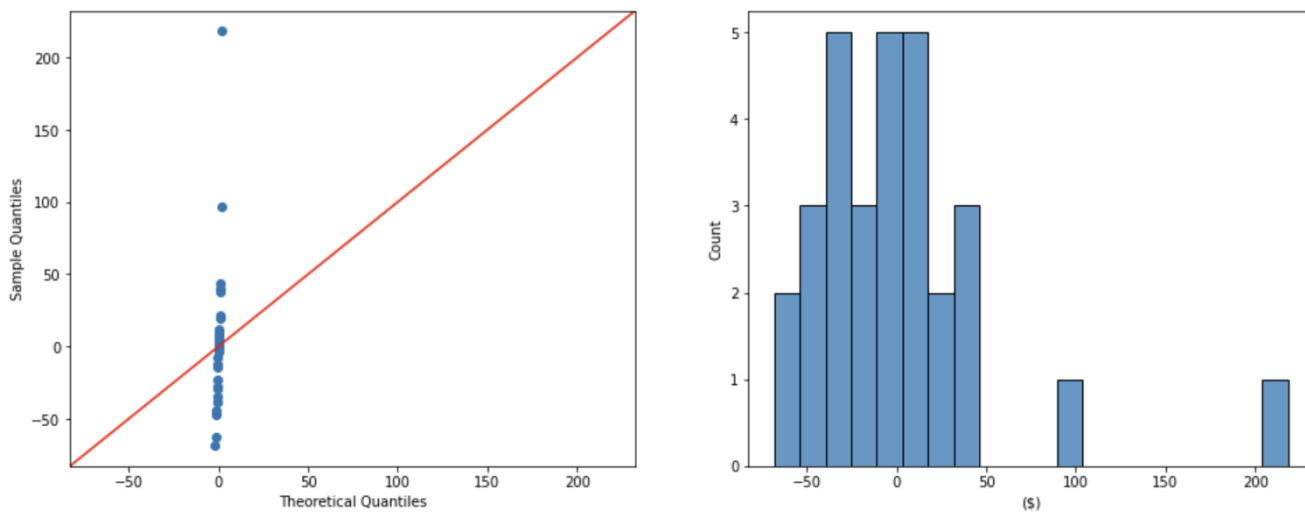


Рисунок. 6.5 – Перевірка на нормальність спостережень для навчальних тижнів (dense стратегія)

Аналізуючи рис. 6.5, можна сказати, що спостереження не має нормального розподілу. Тому для порівняння двох груп використаємо непараметричний U-критерій Манна – Уїтні. Результати наведено в табл. 6.1.

Таблиця 6.1 – Порівняння двох стратегій згідно критерію Манна – Уїтні

Значення статистики	Рівень значущості (p)
290	0,018

Рівень значущості $p < 0,05$ дозволяє нам сказати, що існує статистично значуща відмінність між двома стратегіями. Іншими словами, використання постійних винагород краще впливає на навчання агентів.

6.4 Доцільність впровадження системи

Метою роботи було визначити доцільність інтеграції інтелектуальних систем в робочі процеси низьковольтних енергомереж для зменшення періоду окупності відповідного технічного забезпечення.

Для цього достатньо порівняти рахунок за електроенергію в трьох випадках:

- перший – компонент енергомережі «Consumer of Electricity» завжди споживає електроенергію від компоненту «Power Grid»;
- другий – додаткове встановлення компонента «Solar PV Module», який постійно продає електроенергію «Power Grid»;
- третій – додаткове встановлення компонента «Electricity Storage» із інтеграцією компонента «Decision-Making Module».

Описові характеристики для кожного із випадків наведено в табл. 6.2 та 6.3, а також у вигляді стовпчастої діаграми.

Аналізуючи результати наведені у таблиці 6.5, можна сказати, що впровадження мультиагентної системи прийняття рішень (випадок N2) дозволило зменшити щотижневий рахунок за електроенергію в середньому на 8,66 доларів США, порівнюючи з випадком N3, де використовується лише фотоелектричний модуль.

Таблиця 6.2 — Описові характеристики для навчальних тижнів

Випадок	Середнє значення (\$)	Стандартне відхилення (\$)
N1	113,95	36,87
N2	34,49	36,51
N3	3,05	43,25

Таблиця 6.3 — Описові характеристики для тестових тижнів

Випадок	Середнє значення (\$)	Стандартне відхилення (\$)
N1	91,29	14,28
N2	26,49	22,82
N3	17,83	26,55

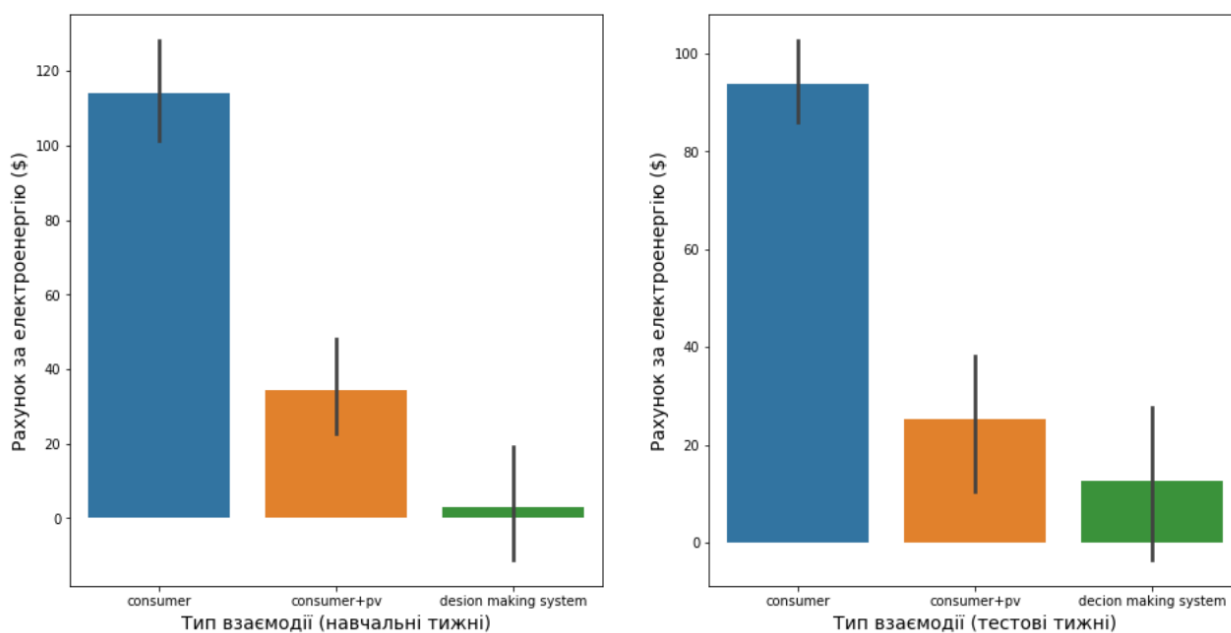


Рисунок 6.5 – Величина рахунку в залежності від встановленого технічного обладнання

Аналізуючи результати для тестових тижнів на рис. 6.5, можна зробити висновок, що величина стандартного відхилення може сигналізувати про наявність

сезонних шаблонів в межах року. Тому додатково було досліджено сезонну складову (рис. 6.6).

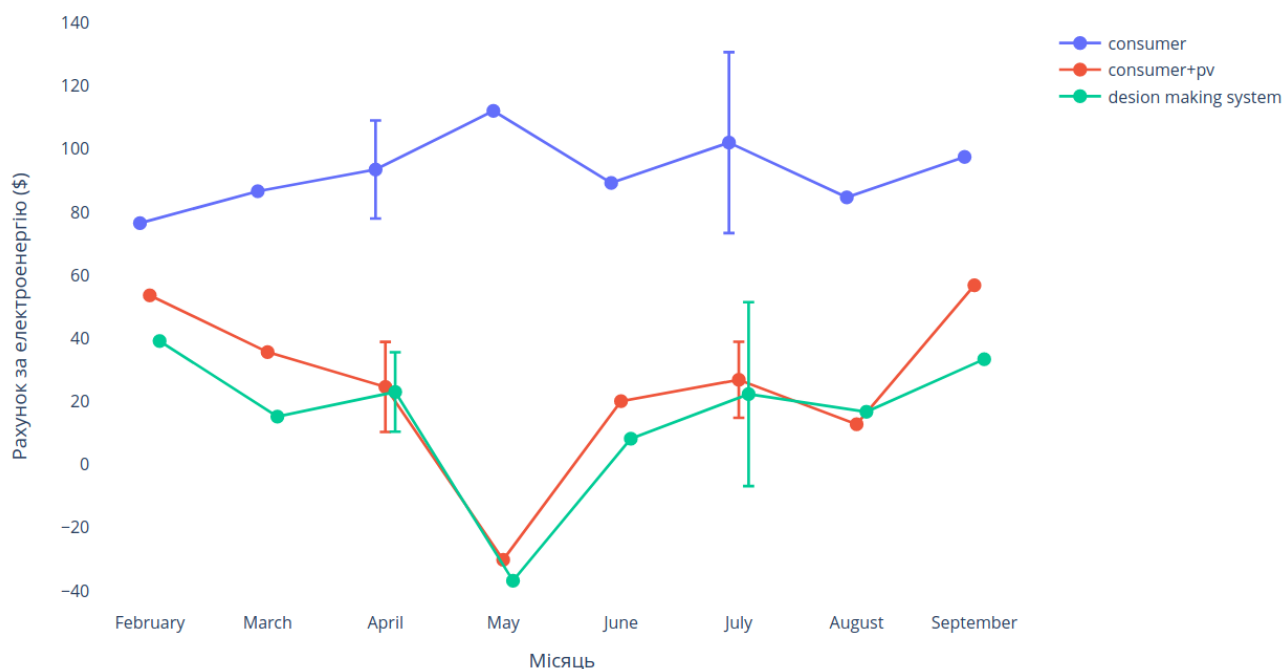


Рисунок 6.6 – Згрупована величина рахунку для кожного місяця

На рис. 6.6 чітко простежується сезонна складова:

- коли тривалість сонячного дня менша (лютий, березень, вересень), рахунок за електроенергію для третього випадку значно менший;
- для трьох місяців сказати чітко, що система прийняття дозволяє зменшити оплату електроенергії не можна.

Можна зробити припущення, що система прийняття рішень робить велику кількість нерациональних дій, погано враховуючи зроблені нею ж дії та отриману винагороду в попередні періоди часу, що не дозволяє зменшити рахунок на більшу величину.

6.5 Представлення роботи

Демонстрація роботи агента, що відповідає за купівлю / продаж електроенергії від центральної мережі, наведено на рис. 6.7.

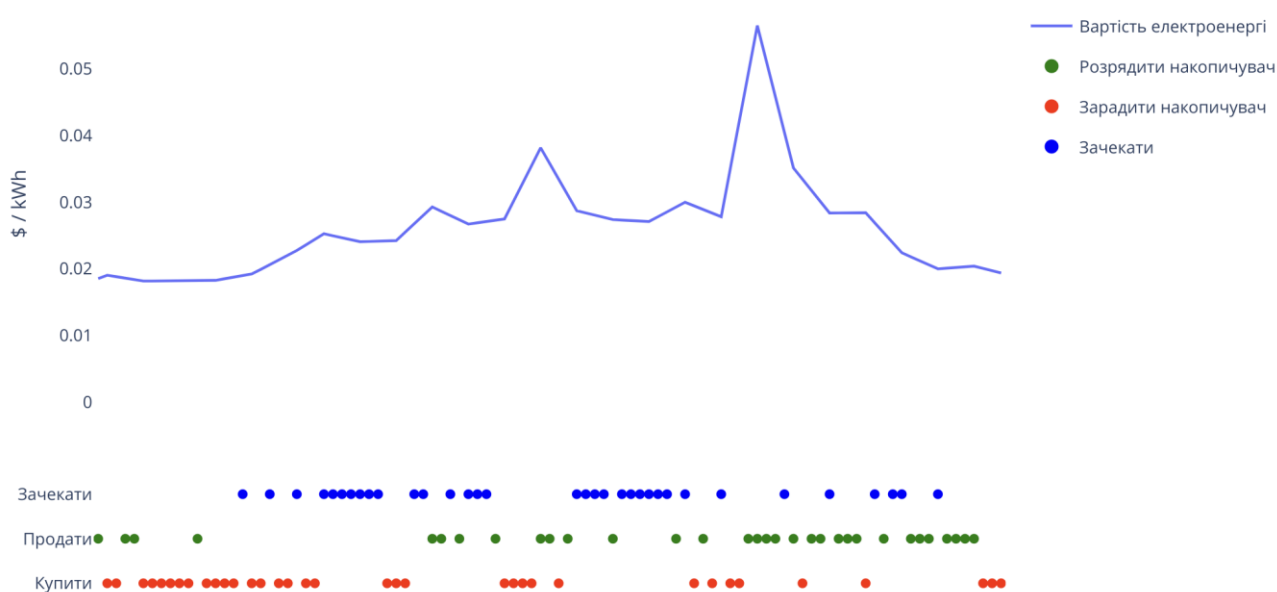


Рисунок 6.7 – Перелік дій купівлі / продажу електроенергії

На рис. 6.7 чітко простежується три періоди часу, в рамках яких, одна з подій виконується більшу кількість раз. У випадку низької ціни домінує операція купівлі, високої — продаж.

Крива споживання та генерації електроенергії (рис. 6.8), дозволяє неявно оцінити, за яких умов другий агент робив ту чи іншу дію.

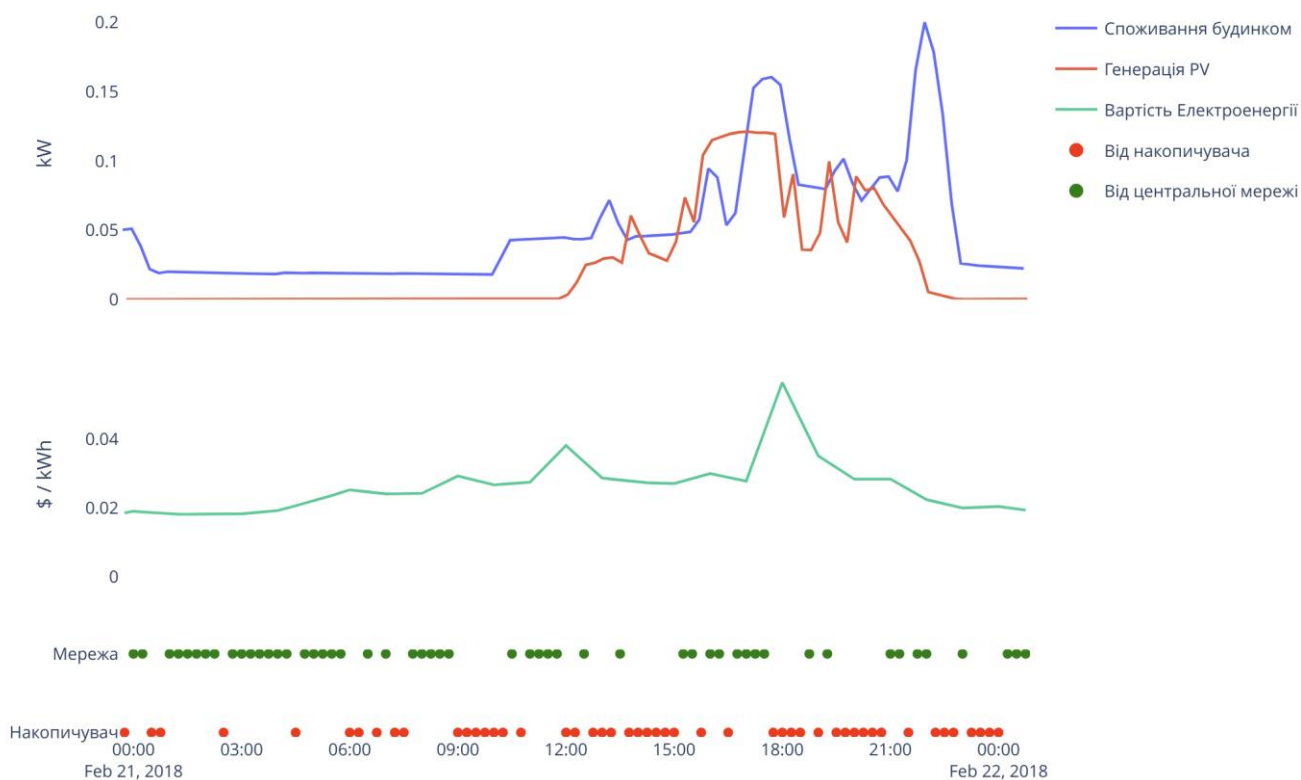


Рисунок 6.8 – Вибір джерела електроенергії

6.6 Висновки до розділу 6

- 1) представлено результати роботи системи купівлі / продажу електроенергії в рамках низьковольтних мереж;
- 2) показано, що вибір стратегії винагороди для агентів має вплив на кінцеві результати;
- 3) проведено аналіз доцільності впровадження системи прийняття рішень. Результати показують, що в короткі сонячні дні система прийняття рішень дозволяє зменшити рахунок за електроенергію.

ВИСНОВКИ

У магістерській дисертаційній роботі одержано такі нові теоретичні та практичні результати:

1. Досліджено існуючі теоретичні та практичні підходи до прийняття рішень в рамках низьковольтних енергомереж. В ході дослідження з'ясовано, що наразі спільне використання підходів явного та неявного використання фізичної моделі не застосовувалося. І насамперед, це пов'язано із часовою складністю алгоритмів, що використовують точне фізичне представлення фізичної моделі, та необхідністю їх інтеграції з даними, які збираються із різних джерел. Встановлено актуальність впровадження систем керування гібридними мережами, зокрема низьковольтними.
2. Вперше в рамках єдиної мультиагентної системи прийняття рішень з купівлі/продажу електроенергії були поєднані симулювання роботи накопичувача електроенергії через модель Дойл-Фуллер-Ньюмана та обробка реальних даних про сонячне випромінення, метеорологічні показники, споживання електроенергії та її вартість, що дозволяє приймати обґрунтовані рішення з планування роботи мікромережі для зменшення рахунків за електроенергію. Наведено структурне та динамічне представлення мікромережі.
3. Виконано збір даних та проведено проектування кожного компоненту мікромережі, наведено деталізовані динамічні представлення та інтерфейси взаємодії між ними, що дозволило вітворити процес генерації, перетворення, транспортування та споживання електроенергії в рамках енергомережі.
4. Проведено огляд технологій накопичення електроенергії. Аргументовано доцільність використання Li-ion, NiCd та NaS технології накопичення в рамках систем, що здійснюють купівлю та продаж електроенергії. Було обрано модель Дойл-Фуллер-Ньюмана для Li-ion накопичувача, що надало можливість

отримувати точні значення сили струму, напруги, температури модуля та внутрішні втрати енергії під час збереження та передачі електроенергії.

5. Було реалізовано декілька власних функціональних елементів, які дозволяють об'єднувати одиничні елементи живлення паралельно та послідовно, що дозволило створити повноцінну конфігурацію Li-ion накопичувача необхідної ємності. Додатково було реалізовано інструменти візуалізації та взаємодії з іншими компонентами мікромережі.
6. Реалізовано процес навчання агентів, кожен з яких здійснює чітко визначений перелік дій і впливає на роботу та стан мікромережі, що дозволило приймати обґрунтовані рішення в рамках нелінійної системи.
7. Встановлено, що розроблена система купівлі / продажу електроенергії в рамках низьковольтних енергомереж дозволяє зменшити щотижневий рахунок за електроенергію для кінцевого користувача (табл. 6.3) в середньому на 8,66 доларів США.

В контексті подальших досліджень та інтеграції гібридних енергомереж в центральну мережу, оптимального керування низьковольтними енергомережами, зокрема, використання мультиагентних систем прийняття рішень, можливі такі кроки:

- збір спостережень про метеорологічні дані, випромінення сонячної радіації та споживання електроенергії кінцевих споживачів для більшого календарного горизонту;
- імплементація різних алгоритмів навчання мультиагентних систем та подальша перевірка їх ефективності;
- використання алгоритмів прогнозування часових залежностей для формування стану системи мікромережі в майбутньому з кінцевою метою покращення ефективності рішень агентів

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Чертов, О. Р., Бірук С.В. Математичне та програмне забезпечення системи купівлі / продажу електроенергії потужностей відновлюваної енергетики. Прикладна математика та комп'ютинг. ПМК-2022: п'ятнадцята науково-практична конференція магістрантів та аспірантів, Київ, 16-17 лист. 2022 р.: зб. Тез доп./ [редкол.: Дичка І. А. та ін.]. — К. : Просвіта, 2022. — С. 57-72.
2. Zhao D. An empirical study on the energy consumption in residential buildings after adopting green building standards / Dong Zhao, Andrew McCoy, Jing Du // *Procedia engineering*. – 2016. – Vol. 145. – P. 766–773.
3. A review on bi-source, off-grid hybrid power generation systems based on alternative energy sources / M. S. Aziz et al. *Journal of renewable and sustainable energy*. 2015. Vol. 7, no. 4.
4. Morris C. G., Cleveland C. J. *Dictionary of energy* / ed. by K. (Firm). Amsterdam : Elsevier, 2009. 581 p.
5. Más H. F. Beyond energy savings: The necessity of optimising smart electricity systems with resource efficiency and coherent waste policy in Europe / Heyd F. Más, Dirk Kuiken // *Energy research & social science*. – 2020. – Vol. 70.
6. Sioshansi F. *Smart grid: integrating renewable, distributed and efficient energy*. Elsevier Science & Technology Books, 2011.
7. Henderson M. I. Electric power grid modernization trends, challenges, and opportunities [Electronic resource] / Michael I. Henderson // Homepage - CMU - Carnegie Mellon University. – Mode of access: <https://www.cmu.edu/epp/irle/readings/henderson-novosel-crow-electric-power-grid-modernization.pdf> (date of access: 10.05.2023)
8. Faiz A. Hybrid renewable energy systems: hybridization and advance control [Electronic resource] / Asghar Faiz, Abdul Rehman // 2015 power generation systems and renewable energy technologies (PGSRET), Islamabad, 10–11 June 2015. – [S. l.], 2015. – Mode of access: <https://doi.org/10.1109/pgsret.2015.7312256> (date of access: 13.05.2023).

9. Control strategy of a hybrid energy storage system to smooth photovoltaic power fluctuations considering photovoltaic output power curtailment / Wei Ma [et al.] // Sustainability. – 2019. – Vol. 11, no. 5. – P. 1324.
10. Garg V. K. Overview on microgrid system / Vijay Kumar Garg, Sudhir Sharma // 2018 fifth international conference on parallel, distributed and grid computing (PDGC), Solan Himachal Pradesh, India, 20–22 December 2018.
11. EPRI home. EPRI Home. URL: <https://www.epri.com/research/products/1003973> (date of access: 09.05.2023).
12. Energy-management solutions for microgrids / Mostafa H. Mostafa [et al.] // Distributed energy resources in microgrids. – [S. l.], 2019. – P. 483–515.
13. Fischer D., Madani H. On heat pumps in smart grids: a review. Renewable and sustainable energy reviews. 2017. Vol. 70. P. 342–357.
14. Battery management for grid-connected PV systems with a battery / S. Park et al. The 2012 ACM/IEEE international symposium, Redondo Beach, California, USA, 30 July – 1 August 2012. New York, New York, USA, 2012.
15. Caron S., Kesidis G. Incentive-Based energy consumption scheduling algorithms for the smart grid. 2010 1st IEEE international conference on smart grid communications (smartgridcomm), Gaithersburg, MD, USA, 4–6 October 2010.
16. Ratnam E. L., Weller S. R., Kellett C. M. An optimization-based approach for assessing the benefits of residential battery storage in conjunction with solar PV. 2013 IREP symposium - bulk power system dynamics and control - IX optimization, security and control of the emerging power grid (IREP), Rethymno, Greece, 25–30 August 2013.
17. Ratnam E. L., Weller S. R., Kellett C. M. An optimization-based approach to scheduling residential battery storage with solar PV: assessing customer benefit. Renewable energy. 2015. Vol. 75. P. 123–134.
18. Distributed energy storage system scheduling considering tariff structure, energy arbitrage and solar PV penetration / O. Babacan et al. Applied energy. 2017. Vol. 205. P. 1384–1393.
19. Salpakari J., Lund P. Optimal and rule-based control strategies for energy flexibility in buildings with PV. Applied energy. 2016. Vol. 161. P. 425–436.

20. Predictive Control of buildings for Demand Response with dynamic day-ahead and real-time prices / E. Vrettos et al. 2013 european control conference (ECC), Zurich, 17–19 July 2013.
21. Cecati C., Citro C., Siano P. Combined operations of renewable energy systems and responsive demand in a smart grid. *IEEE transactions on sustainable energy*. 2011. Vol. 2, no. 4. P. 468–476.
22. Real-Time energy management of a stand-alone hybrid wind-microturbine energy system using particle swarm optimization / S. A. Pourmousavi et al. *IEEE transactions on sustainable energy*. 2010. Vol. 1, no. 3. P. 193–201.
23. Real time operation of smart grids via FCN networks and optimal power flow / P. Siano et al. *IEEE transactions on industrial informatics*. 2012. Vol. 8, no. 4. P. 944–952.
24. Multiobjective intelligent energy management for a microgrid / A. Chaouachi et al. *IEEE transactions on industrial electronics*. 2013. Vol. 60, no. 4. P. 1688–1699.
25. Distributed energy trading in microgrids: a game-theoretic model and its equilibrium analysis / J. Lee et al. *IEEE transactions on industrial electronics*. 2015. Vol. 62, no. 6. P. 3524–3533.
26. Wang H., Huang J. Incentivizing energy trading for interconnected microgrids. *IEEE transactions on smart grid*. 2018. Vol. 9, no. 4. P. 2647–2657.
27. Risk-Averse energy trading in multienergy microgrids: a two-stage stochastic game approach / C. Li et al. *IEEE transactions on industrial informatics*. 2017. Vol. 13, no. 5. P. 2620–2630.
28. Lu R., Hong S. H. Incentive-based demand response for smart grid with reinforcement learning and deep neural network. *Applied energy*. 2019. Vol. 236. P. 937–949.
29. Real-Time energy management of a microgrid using deep reinforcement learning / Y. Ji et al. *Energies*. 2019. Vol. 12, no. 12. P. 2291.
30. A unified decision making framework for supply and demand management in microgrid networks / D. R. Bharadwaj et al. 2018 IEEE international conference on communications, control, and computing technologies for smart grids (smartgridcomm), Aalborg, 29–31 October 2018.

- 31.Homer software and data download [Electronic resource]. – Mode of access: <http://homer.ucsd.edu/homer/> (date of access: 09.05.2023).
- 32.Hybrid2 (usa) [Electronic resource] // Photovoltaic-software.com. – Mode of access: <https://photovoltaic-software.com/pv-softwares-calculators/free-photovoltaic-software/hybrid2-usa> (date of access: 09.05.2023).
- 33.RETScreen [Electronic resource] // Language selection - Natural Resources Canada / Sélection de la langue - Ressources naturelles Canada. – Mode of access: <https://natural-resources.canada.ca/maps-tools-and-publications/tools/modelling-tools/retscreen/7465> (date of access: 09.05.2023).
- 34.Adaptive neural network-based control of a hybrid AC/DC microgrid / Nadjwa Chettibi [et al.] // IEEE transactions on smart grid. – 2016. – P. 1–13.
- 35.Review of the maximum power point tracking algorithms for stand-alone photovoltaic systems / V. Salas [et al.] // Solar energy materials and solar cells. – 2006. – Vol. 90, no. 11. – P. 1555–1578.
- 36.Wald L. Basics in solar radiation at earth surface [Electronic resource] / Lucien Wald // index - Mines Paris. – Mode of access: <https://hal-mines-paristech.archives-ouvertes.fr/hal-01676634> (date of access: 09.05.2023).
- 37.Gür T. M. Review of electrical energy storage technologies, materials and systems: challenges and prospects for large-scale grid storage / Turgut M. Gür // Energy & environmental science. – 2018. – Vol. 11, no. 10. – P. 2696–2767.
- 38.Analogy between thermal, mechanical, and electrical energy storage systems / Karthik Panchabikesan [et al.] // Encyclopedia of energy storage. – [S. l.], 2022. – P. 315–328.
- 39.Duryea S. A battery management system for stand-alone photovoltaic energy systems / S. Duryea, S. Islam, W. Lawrance // IEEE industry applications magazine. – 2001. – Vol. 7, no. 3. – P. 67–72.
- 40.A Computationally Efficient Implementation of an Electrochemistry-Based Model for Lithium-Ion Batteries / L. Xia [et al.] // IFAC-PapersOnLine. – 2017. – Vol. 50, no. 1. – P. 2169–2174.
- 41.Control strategies in AC microgrid: a brief review / Asma Alfergani [et al.] // 2018 9th international renewable energy congress (IREC), Hammamet, 20–22 March 2018.

- 42.Nian R. A review On reinforcement learning: introduction and applications in industrial process control / Rui Nian, Jinfeng Liu, Biao Huang // Computers & chemical engineering. – 2020. – Vol. 139. – P. 3-7.
- 43.Schulman J. Trust region policy optimization [Electronic resource] / John Schulman // arXiv.org. – Mode of access: <https://arxiv.org/abs/1502.05477> (date of access: 09.05.2023).
- 44.Baseline Surface Radiation Network: Baseline Surface Radiation Network. – Mode of access:https://bsrn.awi.de/fileadmin/user_upload/bsrn.awi.de/Publications/BSRN_recommended_QC_tests_V2.pdf (date of access: 09.05.2023).
- 45.Open energy data initiative (OEDI) [Electronic resource] // OpenEI. – Mode of access: <https://data.openei.org/submissions/4520> (date of access: 09.05.2023).
- 46.Nelder J. A. A simplex method for function minimization / J. A. Nelder, R. Mead // The computer journal. – 1965. – Vol. 7, no. 4. – P. 308–313.
- 47.RLlib: industry-grade reinforcement learning – ray 2.4.0 [Electronic resource] // Welcome to Ray! – Ray 2.4.0. – Mode of access: <https://docs.ray.io/en/latest/rllib/index.html> (date of access: 10.05.2023).

Додаток А. Лістинги програм

```

import numpy as np
import pandas as pd
from pathlib import Path
import geopandas as gpd
import plotly.express as px

from .iso import ISO

class ISORegions:

    @classmethod
    def census_region_filename(cls, iso: ISO):
        return 'map_of_pumas_in_census_region_{}'.format(
            iso.census_region_name
        )

    @classmethod
    def load_census_region_map(cls, iso: ISO):
        project_dir = Path(os.environ["PROJECT_DIR"])
        path = project_dir.joinpath(os.environ['GEOLOCATION_DIR']).\
            joinpath(cls.census_region_filename(iso))

        samples = gpd.read_file(path)
        samples.rename(columns = {'GISJOIN': 'PUMAJOIN'}, inplace = True)
        return samples

    @classmethod
    def create_counties_samples(cls, iso: ISO, region_name: str):
        if region_name not in iso.regions:
            raise Exception(f'{iso._name} does not contain a {region_name}')
        return iso.samples.loc[iso.samples.region == region_name, :]

    @classmethod
    def plot_regions_map(cls, iso: ISO):
        census_region_data = cls.load_census_region_map(iso)
        plot_data = census_region_data.merge(iso.samples, how = 'inner', on = 'puma_tsv')
        plot_data.index = plot_data.PUMA

```

```

return px.choropleth_mapbox(
    plot_data, geojson = plot_data.geometry, locations = plot_data.PUMA,
    color="region_label", zoom=7, center=iso.center_coord,
    mapbox_style = "carto-positron", opacity = 0.5
)

```

@classmethod

```

def plot_counties_map(cls, iso: ISO, region_name: str):
    census_region_data = cls.load_census_region_map(iso)
    puma_county_data = cls.create_counties_samples(iso, region_name)

    plot_data = census_region_data.merge(puma_county_data, how = 'inner', on = 'puma_tsv')
    plot_data.index = plot_data.PUMA

    return px.choropleth_mapbox(
        plot_data, geojson = plot_data.geometry, locations = plot_data.PUMA,
        color="PUMA", zoom=7, center=iso.center_coord,
        mapbox_style = "carto-positron", opacity = 0.5
    )

```

```

import numpy as np
import os
import pandas as pd
from pathlib import Path
from abc import ABC, abstractmethod
import os

```

```

from .iso import ISO

```

```

class ISOEnergyPrice(ABC):

```

```

    def __init__(self, iso_name: str):
        self._price_data = pd.concat([
            pd.read_csv(f'{self.price_data_dir}/{iso_name}/hourly_day_ahead_part1_year2018.csv'),
            pd.read_csv(f'{self.price_data_dir}/{iso_name}/hourly_day_ahead_part2_year2018.csv')
        ])

        self.preprocess_price_data()

```

@property

```

def price_data_dir(self):
    return Path(os.environ["PROJECT_DIR"]).\
        joinpath(os.environ['ELECTRICITY_DIR'])

@property
@abstractmethod
def columns(self):
    pass

def preprocess_price_data(self):
    self._price_data.rename(columns = self.columns, inplace = True)
    self._price_data['timestamp'] = pd.to_datetime(self._price_data.timestamp)
    regions_names = [column for column in self.columns.values() if 'zone' in column]
    for name in regions_names:
        self._price_data[name] = self._price_data[name].apply(
            lambda x: float(x.replace('$', '')))

@property
def price_data(self):
    return self._price_data

class PJMEnergyPrice(ISOEnergyPrice):

    def __init__(self, iso: ISO):
        if iso.name != 'PJM':
            raise Exception("This is not PJM object")
        super().__init__(iso.name)
        self._iso = iso

    @property
    def columns(self):
        return {

            'HOURBEGINNING_TIME': 'timestamp',
            'DOM': 'zone_dom',
            'PECO': 'zone_peco'
        }

    def region_key_by_puma_id(self, puma_id: str):
        key = self._iso.samples.loc[self._iso.samples.puma == puma_id, 'region_id'].item()

```

```
return f'zone_{key}'
```

```
def price_data_by_puma_id(self, puma_id: str):
    column = self.region_key_by_puma_id(puma_id)
    frame = self.price_data.loc[:, ['timestamp', column]]
    frame.sort_values(by = 'timestamp', inplace = True)
    return frame.rename(columns = {column: 'price'})
```

s np

```
import pandas as pd
from pathlib import Path
import geopandas as gpd
import plotly.express as px
```

```
from .iso import ISO
```

```
class ISORegions:
```

```
    @classmethod
```

```
    def census_region_filename(cls, iso: ISO):
        return 'map_of_pumas_in_census_region_{ }.geojson'.format(
            iso.census_region_name
        )
```

```
    @classmethod
```

```
    def load_census_region_map(cls, iso: ISO):
        project_dir = Path(os.environ["PROJECT_DIR"])
        path = project_dir.joinpath(os.environ['GEOLOCATION_DIR']).\
            joinpath(cls.census_region_filename(iso))

        samples = gpd.read_file(path)
        samples.rename(columns = {'GISJOIN': 'PUMAJOIN'}, inplace = True)
        return samples
```

```
    @classmethod
```

```
    def create_counties_samples(cls, iso: ISO, region_name: str):
        if region_name not in iso.regions:
            raise Exception(f'{iso._name} does not contain a {region_name}')
        return iso.samples.loc[iso.samples.region == region_name, :]
```

```
    @classmethod
```

```

def plot_regions_map(cls, iso: ISO):
    census_region_data = cls.load_census_region_map(iso)
    plot_data = census_region_data.merge(iso.samples, how = 'inner', on = 'puma_tsv')
    plot_data.index = plot_data.PUMA

    return px.choropleth_mapbox(
        plot_data, geojson = plot_data.geometry, locations = plot_data.PUMA,
        color="region_label", zoom=7, center=iso.center_coord,
        mapbox_style = "carto-positron", opacity = 0.5
    )

@classmethod
def plot_counties_map(cls, iso: ISO, region_name: str):
    census_region_data = cls.load_census_region_map(iso)
    puma_county_data = cls.create_counties_samples(iso, region_name)

    plot_data = census_region_data.merge(puma_county_data, how = 'inner', on = 'puma_tsv')
    plot_data.index = plot_data.PUMA

    return px.choropleth_mapbox(
        plot_data, geojson = plot_data.geometry, locations = plot_data.PUMA,
        color="PUMA", zoom=7, center=iso.center_coord,
        mapbox_style = "carto-positron", opacity = 0.5
    )

import numpy as np
import pandas as pd
import os, io

import boto3
from botocore import UNSIGNED
from botocore.client import Config
from loguru import logger

class HouseLoad:

    def __init__(self, county_id: str, state_abbr: str, building_id: int):
        s3 = boto3.client('s3', config = Config(signature_version = UNSIGNED))
        load_key = "{}{}state={}/{}-{}.parquet".format(

```

```

        self.load_s3_key, state_abbr, building_id, os.environ['UPGRADE_ID'])

    # logger.debug(load_key)
    self._samples = pd.read_parquet(
        io.BytesIO(s3.get_object(
            Bucket = 'oedi-data-lake', Key = load_key)['Body'].read()
        )
    )

    @property
    def load_s3_key(self):
        return (
            "nrel-pds-building-stock/end-use-load-profiles-for-us-building-stock/" +
            "2022/resstock_amy2018_release_1/timeseries_individual_buildings/by_state/" +
            f"upgrade={os.environ['UPGRADE_ID']}"
        )

    @property
    def samples(self):
        return self._samples

    @property
    def electricity_consumption(self):
        samp = self.samples.loc[:, ['timestamp', 'out.electricity.total.energy_consumption']]
        samp.rename(columns = {'out.electricity.total.energy_consumption': 'kWh'}, inplace = True)
        samp.index = samp.timestamp
        return samp

import numpy as np
import pandas as pd
import pybamm

class SingleCell:

    def __init__(self, init_soc: float = 0.5):
        self._init_soc = 0.5
        self.model = pybamm.lithium_ion.DFN()
        self._init_param = pybamm.ParameterValues('Chen2020')

        experiment = pybamm.Experiment(["Rest for 1 hour"])
        simulation = pybamm.Simulation(self.model,

```

```

        experiment = experiment, parameter_values=self._init_param)

simulation.set_initial_soc(init_soc)
self._simulation = simulation.solve()

@property
def soc(self):
    return self._init_soc - self._simulation['Discharge capacity [A.h]'].entries[-1] \
        self._init_param['Nominal cell capacity [A.h]']

@property
def capacity(self):
    return self.soc * self._init_param['Nominal cell capacity [A.h]']

@property
def specification(self):
    return {
        'Lower voltage cut-off [V]': self._init_param['Lower voltage cut-off [V]'],
        'Upper voltage cut-off [V]': self._init_param['Upper voltage cut-off [V]'],
        'Nominal cell capacity [A.h]': self._init_param['Nominal cell capacity [A.h]']
    }

@property
def voltage(self):
    return self._simulation['Battery voltage [V]'].entries[-1]
class BatteryPack:

    def __init__(self,
        num_parallel: int, num_series: int,
        init_soc: float = 0.5
    ) -> None:

        self.num_series = num_series
        self.num_parallel = num_parallel
        self.single_cell = SingleCell(init_soc)

    @property
    def capacity(self):
        return self.single_cell.capacity * \
            self.num_parallel * self.voltage

```

@property

def voltage(self):

return self.single_cell.voltage * self.num_series

@property

def num_cells(self):

return self.num_series * self.num_parallel

def battery_configuration(cell: SingleCell, params: dict):

num_series = params['voltage'] / (cell.specification['Upper voltage cut-off [V]'])

num_parallel = (params['capacity'] / params['voltage']) /\

cell.specification['Nominal cell capacity [A.h]']

return {'num_series': int(num_series), 'num_parallel': int(num_parallel)}

class ChargerDC:

@classmethod

def charge_experiment(cls, battery: BatteryPack, i_mp: float):

portion_i = i_mp / battery.num_cells

return pybamm.Experiment(

[(f'Charge at {portion_i} A for 1 minutes')]

)

@classmethod

def stop_charge_condition(cls, battery: BatteryPack):

hv = battery.single_cell.specification['Upper voltage cut-off [V]']

if battery.single_cell.voltage > 0.95 * hv:

return True

return False

@classmethod

def make_simulation(cls, experiment: pybamm.Experiment,

battery: BatteryPack

) -> None:

simulation = pybamm.Simulation(

battery.single_cell.model, experiment = experiment,

parameter_values=battery.single_cell._init_param

)

```
simulation.set_initial_soc(battery.single_cell.soc)
battery.single_cell._simulation = simulation.solve(
    starting_solution = battery.single_cell._simulation)
```

@classmethod

```
def charge_battery(cls, battery: BatteryPack,
    power: pd.DataFrame, time_period: list
):

    battery_charged = False
    for timestamp in power.loc[time_period[0]:time_period[1], :].index:
        i_mp = mc.dc.loc[timestamp, "i_mp"]
        print(f'Timestamp: {timestamp} I: {i_mp}')

        experiment = cls.charge_experiment(battery, i_mp)
        cls.make_simulation(experiment, battery)

    if cls.stop_charge_condition(battery):
        battery_charged = True
        return battery, power.loc[timestamp:, :], battery_charged

    return battery, power, battery_charged
```

class DischargeDC:

@classmethod

```
def charge_experiment(cls, battery: BatteryPack, i_mp: float):
    portion_i = i_mp / battery.num_cells
    return pybamm.Experiment(
        [(f'DisCharge at {portion_i} A for 1 minutes')]
    )
```

@classmethod

```
def stop_discharge_condition(cls, battery: BatteryPack):
    hv = battery.single_cell.specification['Upper voltage cut-off [V]']
    if battery.single_cell.voltage < 0.05 * hv:
        return True
    return False
```

@classmethod

```
def make_simulation(cls, experiment: pybamm.Experiment,
```

```

        battery: BatteryPack
    ) -> None:

    simulation = pybamm.Simulation(
        battery.single_cell.model, experiment = experiment,
        parameter_values=battery.single_cell._init_param
    )

    simulation.set_initial_soc(battery.single_cell.soc)
    battery.single_cell._simulation = simulation.solve(
        starting_solution = battery.single_cell._simulation)

```

@classmethod

```

def charge_battery(cls, battery: BatteryPack,
                    power: pd.DataFrame, time_period: list
                    ):

    battery_charget = False
    for timestamp in power.loc[time_period[0]:time_period[1], :].index:
        i_mp = mc.dc.loc[timestamp, "i_mp"]
        print(f'Timestamp: {timestamp} I: {i_mp}')

        experiment = cls.charge_experiment(battery, i_mp)
        cls.make_simulation(experiment, battery)

        if cls.stop_charge_condition(battery):
            battery_charged = True
            return battery, power.loc[timestamp:, :], battery_charged

    return battery, power, battery_charged

```

from .Grid import MainGrid

from .ESSs import Battery, DcToAcInvertor, AcToDcInvertor

from .microgrid import Microgrid

class PVToBattery:

@classmethod

```

def charge_battery(cls, cstate, battery: Battery, grid: MainGrid):

```

```

if not battery.stop_charge_condition:
    battery.charge_battery(cstate.p_mp.item())
else:
    energy = DcToAcInvertor.give_you(cstate.p_mp.item())
    grid.sell_energy(cstate, energy)

```

class GridToBattery:

@classmethod

def charge_battery(cls, cstate, grid: MainGrid, battery: Battery):

```

# charge with 0.25C-rate
benergy = battery.energy_by_crate(0.1)
genergy = AcToDcInvertor.give_me(benergy)

```

```

battery.charge_battery(benergy)
grid.buy_energy(cstate, genergy)

```

@classmethod

def discharge_battery(cls, cstate, grid: MainGrid, battery: Battery):

```

# charge with 0.25C-rate # need scale to 15 min not hour
benergy = battery.energy_by_crate(0.1)
genergy = DcToAcInvertor.give_you(benergy)

```

```

battery.discharge_battery(benergy)
grid.sell_energy(cstate, genergy)

```

class HouseToGrid:

@classmethod

def get_from_grid(cls, cstate, grid: MainGrid):

```

grid.buy_energy(cstate, cstate.energy_load.item())

```

class HouseToBattery:

@classmethod

def get_from_battery(cls, cstate, grid: MainGrid, battery: Battery):

```

if not battery.stop_discharge_condition:
    benergy = DcToAcInvertor.give_me(cstate.energy_load.item())
    battery.discharge_battery(benergy)

```

else:

```
grid.buy_energy(cstate, cstate.energy_load.item())
```

```
from .Grid import MainGrid
```

```
from .ESSs import Battery, DcToAcInvertor, AcToDcInvertor
```

```
from .microgrid import Microgrid
```

```
class PVToBattery:
```

```
    @classmethod
```

```
    def charge_battery(cls, cstate, battery: Battery, grid: MainGrid):
```

```
        if not battery.stop_charge_condition:
```

```
            battery.charge_battery(cstate.p_mp.item())
```

```
        else:
```

```
            energy = DcToAcInvertor.give_you(cstate.p_mp.item())
```

```
            grid.sell_energy(cstate, energy)
```

```
class GridToBattery:
```

```
    @classmethod
```

```
    def charge_battery(cls, cstate, grid: MainGrid, battery: Battery):
```

```
        # charge with 0.25C-rate
```

```
        benergy = battery.energy_by_crate(0.1)
```

```
        genergy = AcToDcInvertor.give_me(benergy)
```

```
        battery.charge_battery(benergy)
```

```
        grid.buy_energy(cstate, genergy)
```

```
    @classmethod
```

```
    def discharge_battery(cls, cstate, grid: MainGrid, battery: Battery):
```

```
        # charge with 0.25C-rate # need scale to 15 min not hour
```

```
        benergy = battery.energy_by_crate(0.1)
```

```
        genergy = DcToAcInvertor.give_you(benergy)
```

```
        battery.discharge_battery(benergy)
```

```
        grid.sell_energy(cstate, genergy)
```

```
class HouseToGrid:
```

```

@classmethod
def get_from_grid(cls, cstate, grid: MainGrid):
    grid.buy_energy(cstate, cstate.energy_load.item())

class HouseToBattery:

    @classmethod
    def get_from_battery(cls, cstate, grid: MainGrid, battery: Battery):
        if not battery.stop_discharge_condition:
            benergy = DcToAcInvertor.give_me(cstate.energy_load.item())
            battery.discharge_battery(benergy)
        else:
            grid.buy_energy(cstate, cstate.energy_load.item())

import pickle

import os, copy
import numpy as np
import pandas as pd
from loguru import logger

from .Grid import PJM, PJMEnergyPrice
from .Load import LookupBuilding
from .RESs import Location, PVModule, \
    PVInvertor, PVSystem

class MicrogridConstructor:

    def __init__(self, surface_tilt: float = 32.4,
                surface_azimuth: float = 179):

        self.iso = PJM()
        self.grid = PJMEnergyPrice(self.iso)
        self.pv_system = self.init_pv_system(surface_tilt, surface_azimuth)
        self.house_load = self.init_house_load()
        self.price_data = self.init_price_data()

```

```

def init_pv_system(self, surface_tilt, surface_azimuth):
    logger.info('[Step 1] Initialization pv system ...')
    pv_system_param = { 'location': Location(),
                        'pv_module': PVModule(), 'pv_invertor': PVInvertor(),
                        'num_parallel_modules': int(os.environ['NUM_PARALLEL_MODULE'])
                      }

    pv_system = PVSystem(**pv_system_param)
    pv_system.fit(surface_tilt, surface_azimuth)
    return pv_system

def init_house_load(self):
    logger.info('[Step 2] Loading house load ...')
    lookup = LookupBuilding(self.iso)
    house = lookup.init_house_load(os.environ['PUMA_ID'],
                                   int(os.environ['BUILDING_INDEX']))

    house_load = house.electricity_consumption
    count_data = copy.copy(house_load)

    count_data['next'] = count_data['kWh'].shift(1)
    count_data['load'] = (count_data['next'] + count_data['kWh']) * 0.25 / 2 # Trapezoid Rule

    house_load['kWh'] = count_data.load
    house_load.rename(columns = {'kWh': 'energy_load'}, inplace = True)
    house_load.reset_index(drop = True, inplace = True)
    return house_load

def init_price_data(self):
    logger.info('[Step 3] Loading energy price ...')
    price_data = self.grid.price_data_by_puma_id(os.environ['PUMA_ID'])
    price_data.index = price_data.timestamp
    price_data.drop(columns = ['timestamp'], inplace = True)
    price_data = price_data.asfreq("15Min").interpolate(method='linear')
    price_data['timestamp'] = pd.to_datetime(price_data.index)
    price_data['price'] = price_data['price'] / 10**3 # convert to kW (original MW)
    price_data.reset_index(drop = True, inplace = True)
    return price_data

def resample_power(self, data):
    count_data = copy.copy(data)
    count_data['next_i_mp'] = count_data['i_mp'].shift(1)

```

```
count_data['cum_i_mp'] = (count_data['next_i_mp'] + count_data['i_mp']) * 1 / (60 * 2) # Trapezoid Rule
```

```
count_data['next_p_mp'] = count_data['p_mp'].shift(1)
```

```
count_data['cum_p_mp'] = (count_data['next_p_mp'] + count_data['p_mp']) * 1 / (60 * 2)
```

```
resample_data = pd.DataFrame(None)
```

```
resample_data['i_mp'] = count_data.loc[:, 'cum_i_mp'].resample("15 min").sum()
```

```
resample_data['p_mp'] = count_data.loc[:, 'cum_p_mp'].resample("15 min").sum()
```

```
resample_data['p_mp'] = resample_data.p_mp / 1000 # convert to kW / 15 min
```

```
resample_data['v_mp'] = resample_data.p_mp / resample_data.i_mp
```

```
resample_data.fillna(0, inplace = True)
```

```
resample_data.reset_index(drop = True, inplace = True)
```

```
return resample_data
```

```
def resample_data(self, data):
```

```
    resample_data = data.resample("15 min").mean()
```

```
    resample_data.fillna(0, inplace = True)
```

```
    resample_data.reset_index(drop = True, inplace = True)
```

```
return resample_data
```

```
def surface_power_value(self):
```

```
    data = copy.copy(self.pv_system.power_dc)
```

```
    resample_data = data.loc[:, ['p_mp']].resample("60min").sum()
```

```
    resample_data['p_mp'] = resample_data.p_mp / 1000 # convert to kW / 15 min
```

```
return np.sum(resample_data.p_mp)
```

```
def create_microgrid(self):
```

```
    start_timestamp = self.pv_system.power_dc.index[0].to_datetime64()
```

```
    power_data = self.resample_power(copy.copy(self.pv_system.power_dc))
```

```
    weather_data = self.resample_data(copy.copy(self.pv_system.weather))
```

```
    price_data = self.price_data.loc[self.price_data.timestamp >= start_timestamp, :]
```

```
    house_data = self.house_load.loc[self.house_load.timestamp >= start_timestamp, :]
```

```
    price_data.reset_index(drop = True, inplace = True)
```

```
    house_data.reset_index(drop = True, inplace = True)
```

```
    house_data.drop(columns = ['timestamp'], inplace = True)
```

```
    data = pd.concat([power_data, weather_data, price_data, house_data], axis = 1)
```

```
    data.dropna(inplace = True)
```

```
    columns = ['timestamp', 'i_mp', 'v_mp', 'p_mp', 'ghi', 'dhi', 'dni',
```

```
              'wind_speed', 'temp_air', 'price', 'energy_load']
```

```
data = data[columns]
return data
```

```
def save_data(self, data: pd.DataFrame):
    with open('./data/microgrid_data.pickle', 'wb') as hand:
        pickle.dump(data, hand)
```

```
from .microgrid import MainGrid, Battery
```

```
from .microgrid import HouseToGrid, HouseToBattery, \
    GridToBattery
```

```
class BuyOrSell:
```

```
    @classmethod
```

```
    def make_action(cls, action: int, cstate, grid: MainGrid, battery: Battery):
```

```
        # 0 -- charge, 1 -- discharge, 2 - wait
```

```
        if action == 0:
```

```
            if not battery.stop_charge_condition:
```

```
                GridToBattery.charge_battery(cstate, grid, battery)
```

```
        elif action == 1:
```

```
            if not battery.stop_discharge_condition:
```

```
                GridToBattery.discharge_battery(cstate, grid, battery)
```

```
class GridOrBattery:
```

```
    @classmethod
```

```
    def make_action(cls, action: int, cstate, grid: MainGrid, battery: Battery):
```

```
        # 0 -- grid, 1 -- battery
```

```
        if cstate.energy_load.item() != 0:
```

```
            if action == 0: HouseToGrid.get_from_grid(cstate, grid)
```

```
            elif action == 1: HouseToBattery.get_from_battery(
                cstate, grid, battery)
```

```
import os
```

```
import numpy as np
```

```

import pandas as pd
from loguru import logger

import gym
from gym import spaces
from ray.rllib.env.multi_agent_env import MultiAgentEnv

from .microgrid import PVTtoBattery
from .microgrid import Microgrid, MainGrid, Battery
from .agent_actions import BuyOrSell, GridOrBattery

class MicrogridEnv:

    def __init__(self, env_config = None):
        self._battery, self._grid = None, None
        self.microgrid = Microgrid(env_config['state_columns'])
        self._env_params = env_config
        self.scenes = None
        self._cstate = None

        bstep = int(os.environ['BACKWARD_HOUR'])
        fstep = int(os.environ['FORWARD_HOUR' ])
        self.state_shape = self.find_state_shape(bstep, fstep)

    def find_state_shape(self, bstep, fstep):
        return (4 * (bstep + fstep) + 1) *\
            len(self._env_params['state_columns']) + 2

    def _select_random_week(self):
        weeks = self.microgrid.weeks
        weeks = [week for week in weeks if not (week == np.max(weeks)
            or week == np.min(weeks))]
        return np.random.choice(weeks, size = 1).item()

    def _get_system_state(self, next_state: pd.DataFrame):
        return np.hstack(((next_state.values.reshape(-1),
            self._battery.soc, self._grid.calculate_total_bill()))).astype(np.float32)

    def _calculate_reward(self):
        # strategy 1

```

```

return -self._grid.calculate_bill_by_timestamp(self._cstate)

@property
def _vstate(self):
    return {
        'battery': self._battery,
        'grid': self._grid,
        'state': self._cstate
    }

def reset(self, week: int = None):
    if not week:
        week = self._select_random_week()

    logger.info(f'Week of the Year: {week}')
    self._grid = MainGrid()
    self._battery = Battery( self._env_params['total_capacity'],
        self._env_params['voltage'], self._env_params['init_soc'])

    self.scenes = iter(self.microgrid.iterate_over_week(week))
    try:
        next_state, current_state, _ = next(self.scenes)
        self._cstate = current_state
        return {
            'agent1': self._get_system_state(next_state),
            'agent2': self._get_system_state(next_state)
        }

    except StopIteration:
        return self.reset()

def step(self, action: dict):

    # make actions
    self._grid.add_step()
    PVToBattery.charge_battery(self._cstate, self._battery, self._grid)
    BuyOrSell.make_action(action['agent1'], self._cstate, self._grid, self._battery)
    GridOrBattery.make_action(action['agent2'], self._cstate,
        self._grid, self._battery)

    # count_reward
    reward = self._calculate_reward()

```

```

rewards = {'agent1': reward, 'agent2': reward}

# find next state
next_state, self._cstate, is_done = next(self.scenes)
states = {
    'agent1': self._get_system_state(next_state),
    'agent2': self._get_system_state(next_state)
}

# Generate a `done` dict (per-agent and total).
dones = {'agent1': is_done, 'agent2': is_done, '__all__': is_done}

return states, rewards, dones, {}

```

```
class MicrogridGymEnv(MultiAgentEnv):
```

```

def __init__(self, env_config = None):
    self._agent_ids = ['agent1', 'agent2']
    self._env = MicrogridEnv(env_config)

    self._spaces_in_preferred_format = True
    self._action_space_in_preferred_format = True

    self.observation_space = spaces.Dict(
        {
            'agent1': spaces.Box(low=-np.inf, high=np.inf, shape=(self._env.state_shape, ), dtype = np.float32),
            'agent2': spaces.Box(low=-np.inf, high=np.inf, shape=(self._env.state_shape, ), dtype = np.float32),
        }
    )
    self.action_space = spaces.Dict(
        {'agent1': spaces.Discrete(3), 'agent2': spaces.Discrete(2)}
    )

    super().__init__()

def reset(self, week: int = None):
    return self._env.reset(week)

def step(self, action: dict):
    return self._env.step(action)

```

```

    @property
    def vstate(self):
        return self._env._vstate

import os, pickle

import datetime
import numpy as np
import pandas as pd
from loguru import logger
from pathlib import Path

import ray
from ray import tune
from ray.rllib.env.env_context import EnvContext
from ray.rllib.agents.ppo import TRPOTrainer, DEFAULT_CONFIG

from app import MicrogridGymEnv
from ray.rllib.evaluation.metrics import collect_episodes,\
    summarize_episodes

env_config = {
    'total_capacity': 5,    #kWh
    'voltage': 24, 'init_soc': 0.5,
    'state_columns': ['p_mp', 'price', 'energy_load',
        'minute', 'hour', 'day', 'month']
}

env = MicrogridGymEnv(env_config)
# register_env("MicrogridEnv", lambda params: MicrogridGymEnv())

def policy_mapping_fn(agent_id, episode, worker, **kwargs):
    assert agent_id in ["agent1", "agent2"], f"ERROR: invalid agent ID {agent_id}!"
    return "policy1" if agent_id == "agent1" else "policy2"

policies = {
    "policy1": (None, env.observation_space['agent1'], env.action_space['agent1'], {}),
    "policy2": (None, env.observation_space['agent2'], env.action_space['agent2'], {}),
}

```

```

training_config = {
    'env': MicrogridGymEnv,
    'env_config': env_config,

    # Resource Settings
    'framework': 'torch',
    'num_workers': 2,
    'num_gpus': 1,

    # 'disable_env_checking': True,
    # 'ignore_worker_failures': True

    'multiagent': {
        'policies': policies,
        'policy_mapping_fn': policy_mapping_fn,
        # We'll leave this empty: Means, we train both policy1 and policy2.
        # "policies_to_train": policies_to_train,
    },

    # "preprocessor_pref": None
}

config = DEFAULT_CONFIG.copy()
config.update(training_config)

LOGGIN_DIR = './logs'
if not os.path.exists(LOGGIN_DIR):
    os.makedirs(LOGGIN_DIR)

analysis = tune.run(
    TRPOTrainer,
    config = config,
    local_dir = f'/{LOGGIN_DIR}/{ENVIRONMENT.lower()}',
    stop = { 'training_iteration': 10},
    checkpoint_freq = 10)

checkpoints = analysis.get_trial_checkpoints_paths(
    trial = analysis.get_best_trial('episode_reward_mean'),
    metric = 'episode_reward_mean'
)

```

```
class Agent(TRPOTrainer):
```

```
    @override(Trainable)
```

```
    def load_checkpoint(self, checkpoint_path: str) -> None:
```

```
        data = pickle.load(open(checkpoint_path, 'rb'))
```

```
        worker = pickle.loads(data['worker'])
```

```
        worker = Agent.__fix_recursively(worker)
```

```
        data['worker'] = pickle.dumps(worker)
```

```
        self.__setstate__(data)
```

```
    def __fix_recursively(data):
```

```
        if isinstance(data, dict):
```

```
            return { key: Agent.__fix_recursively(value)
```

```
                    for key, value in data.items() }
```

```
        elif isinstance(data, list):
```

```
            return [Agent.__fix_recursively(value) for value in data]
```

```
        elif data is None: return 0
```

```
        else: return data
```

```
agent = Agent(config = config)
```

```
agent.restore(path_checkpoint)
```

```
rewards, week = [], 2
```

```
state = env.reset(week)
```

```
for i in range(10000):
```

```
    action1 = agent.compute_single_action(state['agent1'], policy_id='policy1')
```

```
    action2 = agent.compute_single_action(state['agent2'], policy_id='policy2')
```

```
    actions = {'agent1': action1, 'agent2': action2}
```

```
    state, reward, done, info = env.step(actions)
```

```
    print(f'Iteration: {i :>5}, Agent1: {actions["agent1"]}, Agent2: {actions["agent2"]} Reward: {reward["agent1"]}:
```

```
.3f}')
```

```
    rewards.append(reward["agent1"])
```

```
if done['__all__']:
```

Додаток Б. Ілюстративний матеріал

Математичне та програмне забезпечення системи купівлі / продажу електроенергії в рамках низьковольтних енергомереж

Студент: Бірук С. В., КМ-11мн

Керівник: Чертов О. Р., доктор технічних наук, професор
кафедри ПМА

Рисунок Б.1 – Слайд 1

Актуальність теми

Використання відновлювальних джерел енергії в повсякденному житті людини дозволить:

- Рівномірно розподілити навантаження на централізовану мережу передачі електроенергії;
- Зменшити негативний вплив на навколишнє середовище;
- Зменшити залежність від центральної комунальної мережі.

Рисунок Б.2 – Слайд 2

Основні проблеми

- Технічні засоби для генерації "зеленої" енергії та засоби управління ними потребують високотехнологічних рішень і систем обробки великих обсягів даних;
- Величина початкових інвестицій у відповідне технічне забезпечення та середній термін окупності відлякують потенційних інвесторів.

4

Рисунок Б.3 – Слайд 3

Постановка задачі

Метою дослідження є оптимізація процесу купівлі / продажу електроенергії з метою зменшення періоду повернення інвестицій, які були залучені при встановленні технічних засобів для генерації та збереження відновлюваної енергії.

Об'єкт дослідження. Гібридна енергетична система, мікромережа та її складові, системи керування мікромережами: HOMER, HYBRID 2, RETScreen, iHOGA та INSEL. Поняття арбітражу на ринку електроенергії, процеси виробництва, перетворення та зберігання енергії в межах гібридної системи.

Підходи та методи управління нелінійними та складними системами: навчання з підкріпленням (Vanilla Policy Gradient, Proximal Policy Optimization, Deep Deterministic Policy Gradient, Twin Delayed DDPG, Soft Actor-Critic); нечітке логічне управління; динамічне програмування та управління з передбаченням поведінки моделі.

Предмет дослідження. Математичне та програмне забезпечення системи купівлі / продажу електроенергії, в основу яких покладені моделі та алгоритми навчання з підкріпленням, що здійснюють арбітраж виробництва та споживання електроенергії.

5

Рисунок Б.4 – Слайд 4

Огляд теоретичних засобів

Таксономія підходів керування мікромережами (низьковольтними енергомережами):

1. Непрогнозоване керування
 - Rule – based алгоритми;
 - Schedule – based алгоритми
2. Прогнозоване керування
 - а. Явне використання фізичної моделі
 - Змішане цілочисельне програмування;
 - Послідовне квадратичне програмування;
 - Оптимізація рою частинок;
 - Нейронні мережі;
 - б. Неявне використання фізичної моделі
 - Нечіткий логічний вивід;
 - Алгоритми теорії ігор із аналізом рівноваги рішення;
 - Навчання з підкріпленням.

8

Рисунок Б.5 – Слайд 5

Огляд практичних засобів

Було розглянуто:

1. Загальноприйняті пакети програмного забезпечення для енергетичного моделювання, такі як TRNSYS, ESP-r, Modelica, EnergyPlus та IDA ICE
2. Спеціалізовані застосунки проектування та економічної оптимізації: HOMER, HYBRID 2, RETScreen, iHOGA та INSEL

9

Рисунок Б.6 – Слайд 6

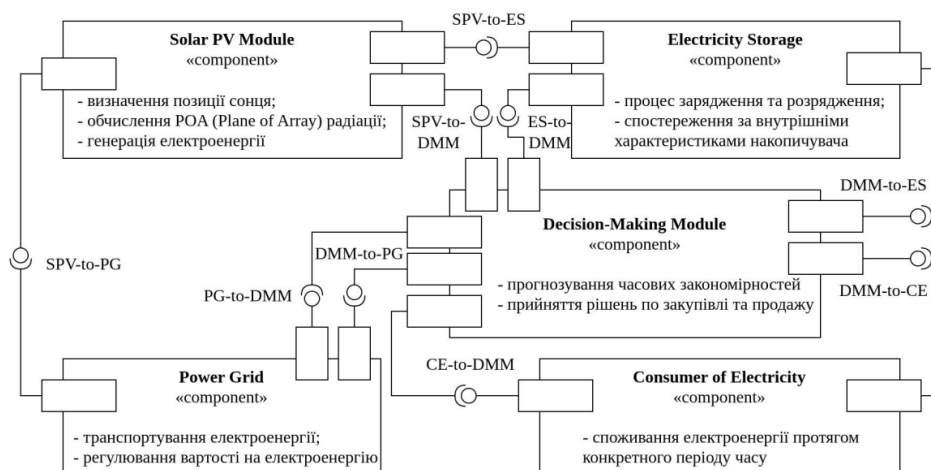
Висновки до розділу:

- Розглянуті теоретичні підходи не враховують всю інформацію про стан середовища: сонячне випромінювання, метеорологічні фактори, вартість електроенергії та її споживання в межах одного будинку.
- В найкращому випадку, розглядаються три види перелічених даних. При цьому, вони несинхронізовані відносно єдиного географічного району і часового періоду.
- Використовуються спрощені моделі компонентів:
 1. DC/AC інвертор (перетворення електроенергії на основі коеф. ефективності);
 2. Накопичувач енергії (оперує рівнем заряду [State of charge, SOC]);
 3. Не відстежується точка максимальної потужності (Maximum Power Points, MPP) в DC перетворювачах, тощо.
- Розглянуті теоретичні рішення поодиночі не дозволяють адаптуватися до нових змін середовища
- Розглянуті практичні рішення надають змогу проектувати мікромережу та проводити економічний аналіз, вони не дозволяють керувати мікромережею в реальному часі.
- Проміжні результати програмних реалізацій приховані від користувача, що не дозволяє проводити якісний, ґрунтовний аналіз.

10

Рисунок Б.7 – Слайд 7

Об'єкт автоматизації

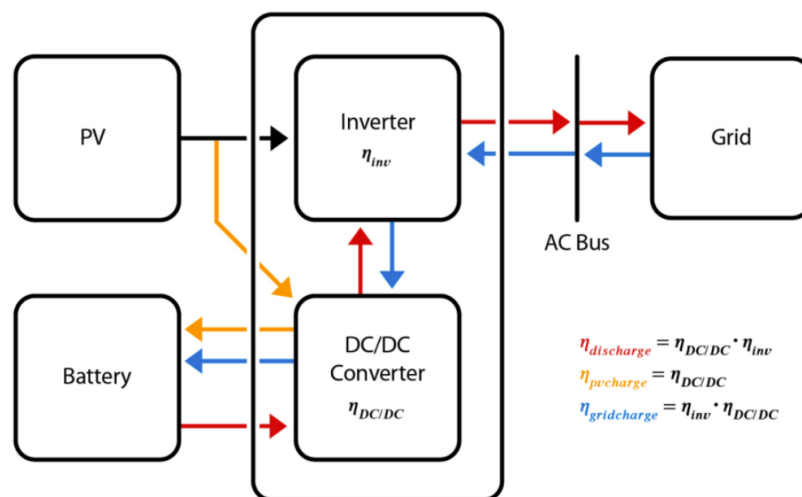


- *Solar PV Modules* – підсистема генерації електроенергії;
- *Electricity Storage* – підсистема збереження відновлювальної та отриманої від електромережі електроенергії;
- *Power Grid* – мережа передачі електроенергії; ця підсистема додатково регулює вартість електроенергії;
- *Consumer of Electricity* – споживач електроенергії;
- *Decision-Making Module* – підсистема планування, яка вирішує (1) коли і в якому об'ємі купити електроенергію або її продати; та (2) від якого із джерел (Electricity Storage, Power Grid) електроенергія потрапляє до споживача.

12

Рисунок Б.8 – Слайд 8

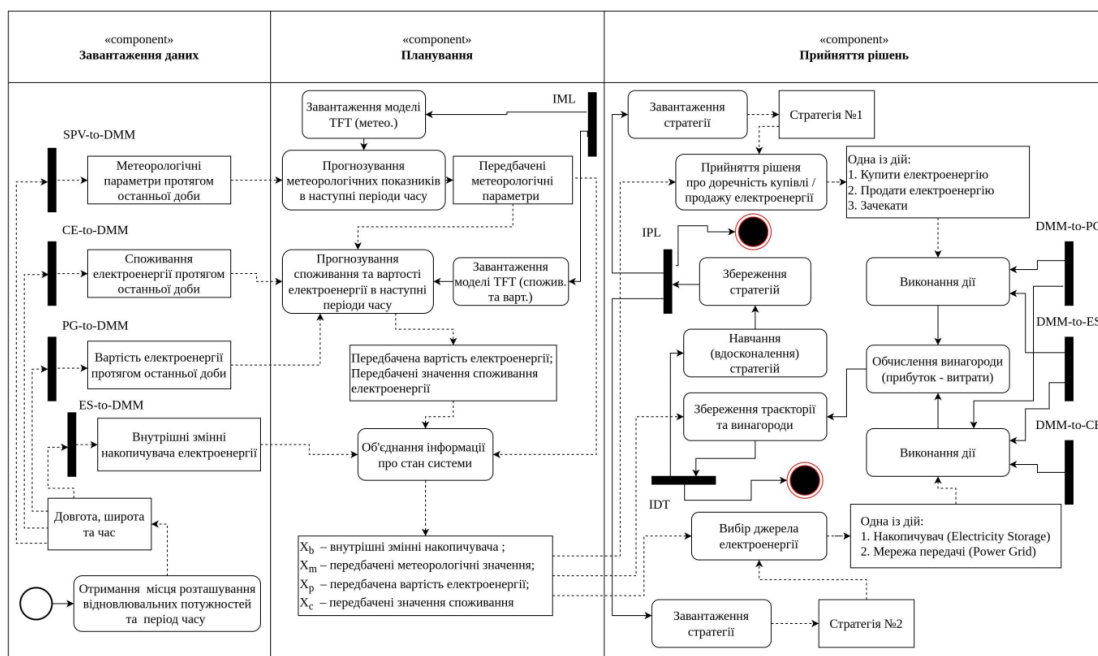
Схема передачі та перетворення електроенергії



13

Рисунок Б.9 – Слайд 9

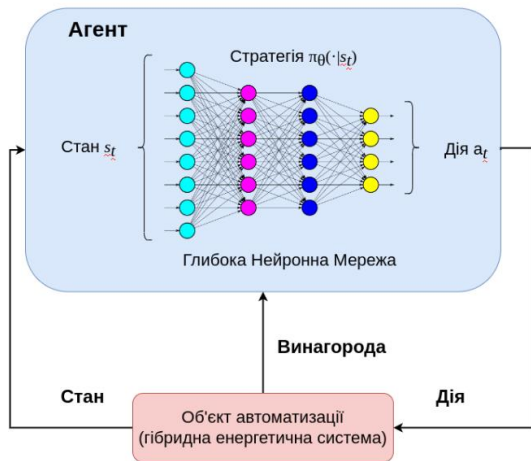
Модель компонента Decision-Making Model



14

Рисунок Б.10 – Слайд 10

Математичне забезпечення компонента Decision-Making Model



Процес прийняття рішень формалізується як задача Markov Decision Process (MDP) [1] і приймає вигляд кортежу виду $\langle S, A, R, P, \rho_0 \rangle$:

- S – множина всіх можливих станів. Спостереження s_t має вигляд кортежу виду $[X_m, X_p, X_c, X_b]$, де X_m – передбачені метеорологічні значення за допомогою TFT; X_p – передбачена вартість електроенергії; X_c – передбачені значення споживання електроенергії; X_b – внутрішні характеристики накопичувача електроенергії;
- A – множина всіх можливих дій;
- $R: S \times A \times S \rightarrow R$ – функція винагороди в грошовому еквіваленті;
- $P: S \times A \rightarrow P(S)$ – функція ймовірності переходу, де $P(s' | s, a)$ – ймовірність переходу в стан s' , якщо в стані s зроблено вибір a ;
- ρ_0 – початковий розподіл станів.

Стратегія (policy) [2] – набір правил, які використовує агент для вибору дії. Позначається наступним чином: $a_t \sim \pi_\theta(\cdot|s_t)$.

Оптимізаційна проблема зводиться до пошуку стратегії, що максимізує очікувану винагороду.

16

Рисунок Б.11 – Слайд 11

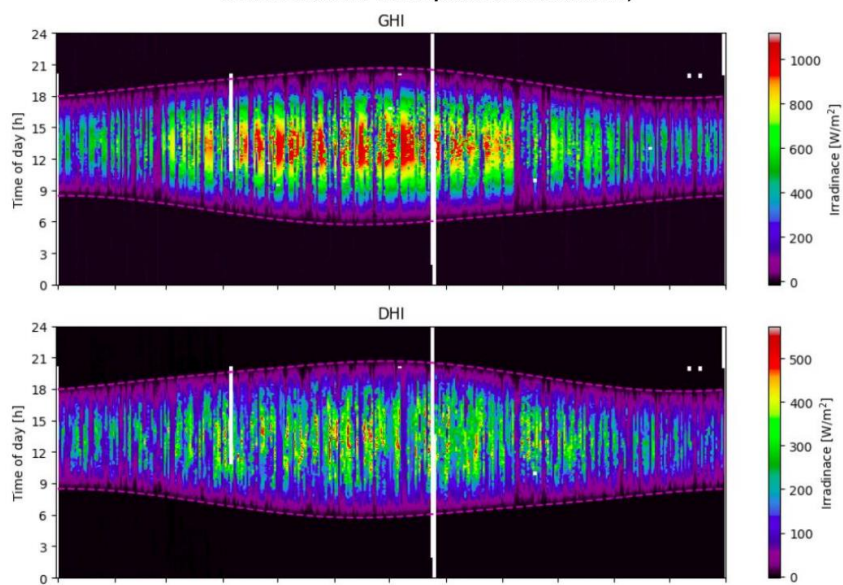
Висновки до розділу

1. Запропоновано мультиагентну систему прийняття рішень, використовуючи наступні стратегії навчання з підкріпленням:
 - Vanilla Policy Gradient;
 - Proximal Policy Optimization;
 - Deep Deterministic Policy Gradient;
 - Twin Delayed DDPG, Soft Actor-Critic
2. Інтегровано фізичну модель Дойл-Фуллер-Ньюман для Li-ion накопичувача

17

Рисунок Б.12 – Слайд 12

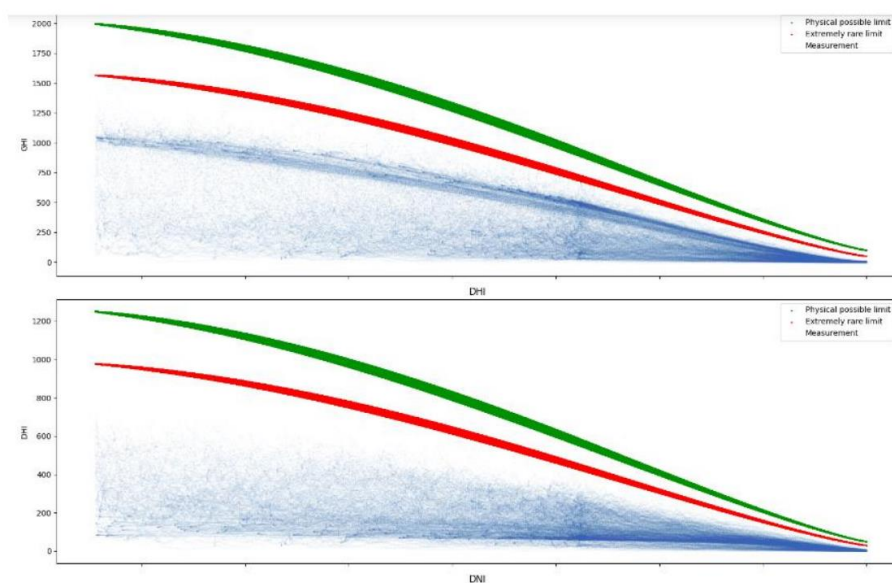
PV Module: Валідація даних (спектограма
сонячного випромінювання)



19

Рисунок Б.13 – Слайд 13

PV Module: Валідація даних (пошук аномальних
спостережень)



20

Рисунок Б.14 – Слайд 14

DC-DC перетворювач: Генерація постійного
струму згідно MPP

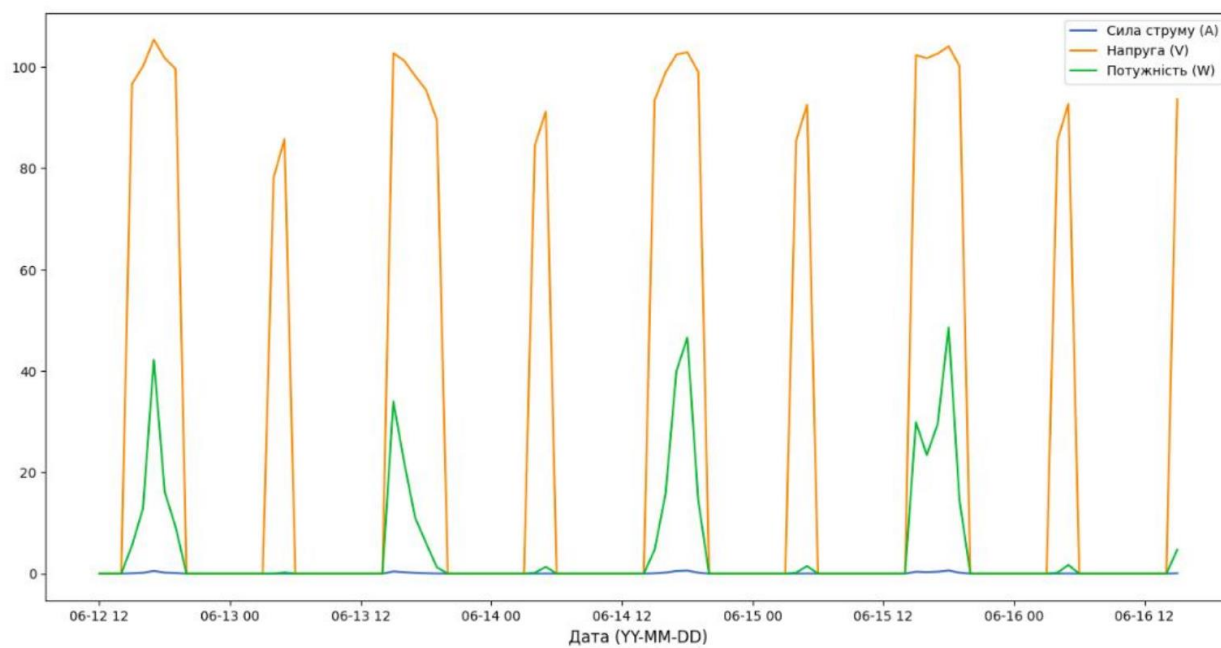
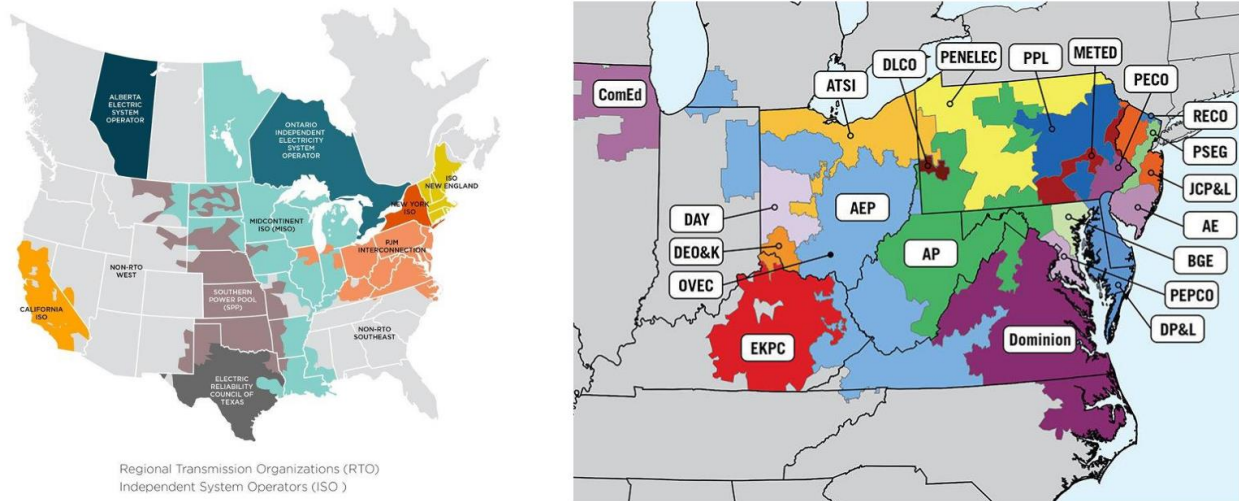


Рисунок Б.15 – Слайд 15

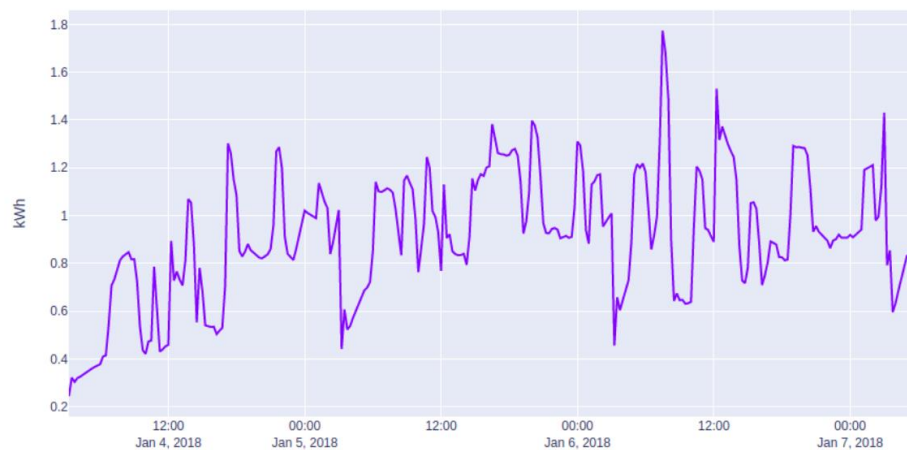
Power Grid: Регіональні зони транспортування електроенергії провайдерами



22

Рисунок Б.16 – Слайд 16

Consumer of Electricity: патерни споживання електроенергії окремим будинком



23

Рисунок Б.17 – Слайд 17

Висновки

За час проходження науково-дослідної практики було:

- формалізовано постановку задачі дисертаційного дослідження;
- досліджено і встановлено інструменти, які на сьогодні дозволяють виконувати управління низьковольтними мережами;
- запропоновано конфігурацію системи, що дозволяє вирішити поставлені задачі.

Рисунок Б.18 – Слайд 18

