

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти – бакалавр
за освітньо-професійною програмою
«Автоматизація та комп'ютерно-інтегровані технології»

зі спеціальності
151 – Автоматизація та комп'ютерно-інтегровані технології

тема роботи:

***«Розробка комп'ютерно-інтегрованої системи
короткострокового прогнозування генерації сонячної
електростанції на основі методів машинного навчання»***

Виконав студент гр. АКІТ-22	_____ Галушка В. О.
Керівник	_____ Жосан А. А.
Нормоконтроль	_____ Маринич І. А.
Завідувач кафедри	_____ Рубан С. А.

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Бакалавр

Спеціальність: 151 - Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Зав. кафедрою: к.т.н. Рубан С.А.

« 26 » січня 2026 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

студентові групи АКІТ-22 Галушки Валерію Олександровичу

1. Тема кваліфікаційної роботи: «Розробка комп'ютерно-інтегрованої системи короткострокового прогнозування генерації сонячної електростанції на основі методів машинного навчання»

затверджено наказом по університету № 171с від 30.03.2025 р.

2. Термін здачі кваліфікаційної роботи: 10.06.2024 р.

3. Склад кваліфікаційної роботи: Пояснювальна записка, презентація у Microsoft PowerPoint в електронному та друкованому вигляді

4. Консультанти кваліфікаційної роботи:

Розділ 1-2

доц. Жосан А. А.

Нормоконтроль

доц. Маринич І. А.

5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>01.03.26</i>
2	<i>Розділ 1</i>	<i>05.04.26</i>
3	<i>Розділ 2</i>	<i>01.05.26</i>
4	<i>Висновки</i>	<i>25.05.26</i>
5	<i>Оформлення кваліфікаційної роботи</i>	<i>28.05.26</i>
6	<i>Підготовка презентації та графічного матеріалу</i>	<i>20.05.26</i>
7	<i>Підготовка доповіді до захисту</i>	<i>05.06.26</i>

6. Дата видачі завдання: 28.01.2026р.

Керівник _____ / Жосан А. А./

7. **Запевнення:** Я, Галушка Валерій Олександрович, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Студент _____ / Галушка В. О./

АНОТАЦІЯ

Галушка В. О. Розробка комп'ютерно-інтегрованої системи короткострокового прогнозування генерації сонячної електростанції на основі методів машинного навчання: кваліфікаційна робота бакалавра : 151 – Автоматизація, комп'ютерно-інтегровані технології. Кривий Ріг. Криворізький національний університет, 2026. 72 с.

Метою роботи є розробка та реалізація моделі прогнозування генерації сонячних електричних станцій із використанням методів машинного навчання для підвищення точності короткострокових прогнозів.

У першому розділі розглянуто сучасний стан і тенденції розвитку сонячної енергетики та зростаючу роль фотоелектричних станцій в енергосистемах. Показано, що їх інтеграція потребує врахування залежності від метеорологічних факторів, що обумовлює необхідність застосування методів аналізу даних і прогнозування. Обґрунтовано, що використання машинного навчання та технологій обробки великих даних є ключовим для підвищення ефективності управління відновлюваною генерацією, зменшення залежності від традиційних джерел енергії та забезпечення балансу між виробництвом і споживанням електроенергії.

У другому розділі детально розглянуто основні підходи машинного навчання, які застосовуються для задач короткострокового прогнозування генерації електроенергії, включаючи регресійні та ансамблеві методи, що забезпечує підвищення точності прогнозів. Проведено порівняльний аналіз реалізацій моделей машинного навчання у бібліотеках Python та MATLAB.

ВІДНОВЛЮВАНІ ДЖЕРЕЛА ЕНЕРГІЇ, КОРОТКОСТРОКОВЕ ПРОГНОЗУВАННЯ, МАШИННЕ НАВЧАННЯ, ПРОГНОЗУВАННЯ, СОНЯЧНА ЕЛЕКТРОСТАНЦІЯ, СОНЯЧНА ГЕНЕРАЦІЯ

					КНУ КРБ.151.26.03.00.ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Галушка В. О.			АНОТАЦІЯ			Літ.	Арк.	Аркушів	
Перевір.		Жосан А. А.								3	1
Н. Контр.		Маринич І.А.									
Затверд.		Маринич І.А.									
								АКНТ АКІТ-22			

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 ОСНОВИ СОНЯЧНОЇ ЕНЕРГЕТИКИ ТА ОСОБЛИВОСТІ ПРОГНОЗУВАННЯ СОНЯЧНОЇ ГЕНЕРАЦІЇ.....	9
1.1 Децентралізація структури електроенергетичних систем.....	9
1.2 Принципи роботи сонячних станцій та їх особливості.....	11
1.3 Вплив зовнішніх факторів на вироблення електроенергії.....	14
1.3.1 Великі дані та роль прогнозування в роботі сонячних електростанцій.....	16
1.4 Методи прогнозування вироблення електроенергії.....	17
1.5 Математичний опис генерації сонячних панелей.....	18
Висновки за розділом:.....	22
РОЗДІЛ 2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ СОНЯЧНОЇ ГЕНЕРАЦІЇ.....	23
2.1 Метрики для аналізу якості роботи алгоритмів.....	23
2.1.1 Бінарна класифікація класів.....	24
2.1.2 Показники якості регресії.....	26
2.2 Особливості застосування методів машинного навчання.....	27
2.3 Опис моделей машинного навчання.....	29
2.3.1 Метод опорних векторів (SVM)	30
2.3.2 Метод найближчих сусідів (KNN).....	31
2.3.3 Метод лінійної регресії.....	33
2.3.4 Метод дерев рішень.....	34

					КНУ КРБ.151.26.03.00.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Галушка В. О.			ЗМІСТ	Літ.		Арк.	Аркушів
Перевір.		Жосан А. А.						4	2
						АКНТ АКІТ-22			
Н. Контр.		Маринич І.А.							
Затверд.		Маринич І.А.							

2.3.5 GridSearch	36
2.3.6 RandomSearch.....	36
2.4 Вибір програмного забезпечення для реалізації.....	37
2.5 Опис вибірки та об'єкта даних.....	39
2.6 Предобробка та підготовка даних для навчання.....	40
2.7 Побудова моделей машинного навчання в середовищі MATLAB.....	45
2.8 Навчання та налаштування моделей з використанням Python	48
2.8.1 Метод лінійної регресії.....	51
2.8.2Дерево рішень.....	52
2.8.3 Випадковий ліс.....	52
2.8.4 Gradient Boosting.....	54
2.8.5 Випадковий ліс за допомогою GridSearch.....	55
2.8.6 XGBoost та GridSearch.....	56
2.9 Оцінка якості моделей та висновки.....	56
Висновки за розділом:.....	58
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	62
ДОДАТОК А. Фрагменти ісходного коду програми на мові Python	65

ВСТУП

Прикладами відновлюваних джерел енергії (ВДЕ) є сонячна та вітрова енергія. У даній роботі розглядається фотоелектрична станція (сонячна електростанція) із застосуванням методів машинного навчання для прогнозування генерації. За загальнодоступними даними станом на 2023 рік частка відновлюваних джерел енергії в Україні продовжує зростати, проте все ще залишається меншою порівняно з традиційними джерелами генерації електроенергії [1]. Незначна частка таких об'єктів пояснюється тим, що традиційні джерела електроенергії, зокрема теплові електростанції (ТЕС) та атомні електростанції (АЕС), мають значну встановлену потужність, а їх модернізація часто є економічно доцільнішою порівняно з будівництвом нових сонячних електричних станцій (СЕС). Крім економічного аспекту, існує залежність сонячних електричних станцій від погодних умов, що також створює труднощі при інтеграції ВДЕ в електроенергетичну систему.

Переваги ВДЕ переважають їхні недоліки порівняно з традиційними джерелами електроенергії, такі станції не використовують для своєї роботи вичерпні природні ресурси та є екологічно безпечними.

Актуальність роботи. Враховуючи вищезазначене, сформульовано глобальне завдання: забезпечити споживачів стабільним електропостачанням із одночасним збереженням природних ресурсів планети [2]. Загалом актуальність використання ВДЕ постійно зростає, оскільки вони пропонують стійкий, екологічно чистий та економічно ефективний спосіб задоволення зростаючих потреб в енергії.

Прогнозування виробництва електроенергії за допомогою сонячних електричних станцій є одним із ключових завдань у сфері відновлюваної енергетики.

					<i>КНУ КРБ.151.26.03.00.ПЗ</i>						
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>ВСТУП</i>				<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розроб.</i>		<i>Галушка В. О.</i>									
<i>Перевір.</i>		<i>Жосан А. А.</i>								6	3
									<i>АКНТ АКІТ-22</i>		
<i>Н. Контр.</i>		<i>Маринич І.А.</i>									
<i>Затверд.</i>		<i>Маринич І.А.</i>									

Основна складність полягає у випадковому характері генерації, що залежить від метеорологічних умов та часових факторів. Для розв'язання цього завдання застосовуються різні методи, серед яких можна виділити три:

1. Статистичні методи – базуються на аналізі даних та побудові математичних моделей взаємозв'язків між параметрами. Ці методи є простими у реалізації та потребують мінімальних обчислювальних ресурсів, однак їхня точність знижується при роботі з параметрами, характерними для сонячної генерації [3].

2. Фізичні методи – ґрунтуються на використанні рівнянь, що описують процеси передачі сонячної енергії через атмосферу та її перетворення в електричну енергію. Перевагою фізичних методів є їхня визначеність, проте вони потребують значного обсягу вхідних даних і високих обчислювальних потужностей [4].

3. Методи машинного навчання – сучасні дослідження активно впроваджують алгоритми машинного навчання для підвищення точності прогнозів. Найбільш популярними методами є:

– Нейронні мережі – особливо ефективні для роботи з нелінійними залежностями та великими обсягами даних;

– Древа рішень та ансамблеві методи – дозволяють враховувати значну кількість факторів і забезпечують інтерпретацію результатів.

У роботі [5] показано, що ансамблі дерев забезпечують вищу точність прогнозування. Можна зробити висновок, що зазначені методи машинного навчання демонструють високу адаптивність та універсальність у задачах прогнозування генерації сонячної електроенергії.

Об'єктом дослідження є генерація сонячної енергії в електроенергетичних системах.

Предметом дослідження є застосування методів машинного навчання для короткострокового прогнозування сонячної генерації.

					КНУ КРБ.151.26.03.00.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Метою роботи є розробка та реалізація моделі прогнозування генерації сонячних електричних станцій із використанням методів машинного навчання для підвищення точності короткострокових прогнозів.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- Проаналізувати існуючі методи прогнозування сонячної генерації та визначити їхні обмеження;
- Розробити модель машинного навчання для короткострокового прогнозування сонячної генерації з використанням різних алгоритмів;
- Реалізувати та протестувати прогнозну модель машинного навчання із використанням реальних даних сонячних електростанцій;
- Порівняти результати моделей машинного навчання із застосуванням різних методів прогнозування.

Практична цінність роботи. Результати проведеного дослідження можуть знайти своє застосування у сфері управління та прогнозування виробітку електроенергії на сонячних електростанціях (СЕС). Це робить їх придатними для використання в оперативному управлінні енергетичними системами та комерційній експлуатації сонячних електростанцій.

					<i>КНУ КРБ.151.26.03.00.ПЗ</i>	Арк.
						8
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ I

ОСНОВИ СОНЯЧНОЇ ЕНЕРГЕТИКИ ТА ОСОБЛИВОСТІ ПРОГНОЗУВАННЯ СОНЯЧНОЇ ГЕНЕРАЦІЇ

Прогнозування із використанням машинного навчання дозволяє забезпечувати баланс між виробництвом та споживанням електроенергії, що, у свою чергу, дає можливість подавати точніші заявки на оптовий ринок та знижує штрафи, зумовлені різницею між фактичними та плановими обсягами генерації.

1.1 Децентралізація структури електроенергетичних систем

Із зростаючими введеннями нових установлених потужностей ВДЕ та їх інтеграцією в електроенергетичні системи, а також нерівномірним розподілом об'єктів на території, структура розподільних мереж електроенергетичної системи змінюється.

Історичний розвиток електроенергетичних систем ґрунтувався на будівництві великих центрів живлення та ліній електропередачі високої напруги і великої протяжності, призначених для передачі електроенергії до віддалених від великого центру живлення споживачів.

Поточні електроенергетичні системи, в умовах наявності безлічі електростанцій меншої потужності, розташовані нерівномірно, лінії електропередачі використовують нижчі класи напруги та мають меншу протяжність.

Таким чином формується децентралізована електроенергетична система з безліччю вузлів генерації середньої та малої потужності, що в окремих випадках працюють автономно для електрозабезпечення локальних

					КНУ КРБ.151.26.03.01.ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Галушка В. О.			РОЗДІЛ I			Літ.	Арк.	Аркушів	
Перевір.		Жосан А. А.								9	18
Н. Контр.		Маринич І.А.			АКНТ АКІТ-22						
Затверд.		Маринич І.А.									

споживачів.

Схема переходу від централізованої структури до децентралізованої представлена на рисунку 1.1.

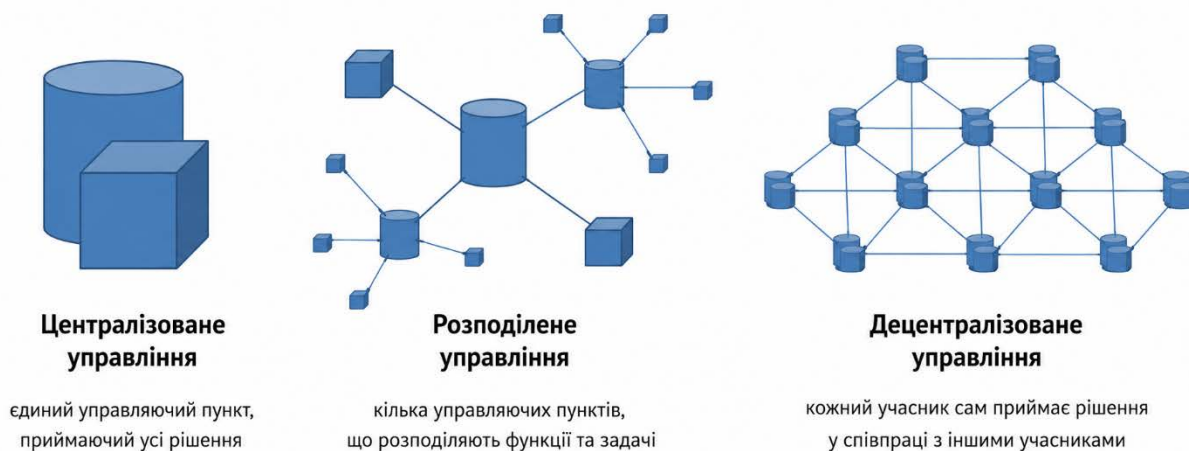


Рисунок 1.1 – Централізована та децентралізована електроенергетичні системи

На зміну великим електростанціям великої потужності, що призначалися для електро- та тепlopостачання великих виробничих кластерів, приходять об'єкти з розподіленою генерацією, гібридні станції, розраховані для живлення невеликих населених пунктів. Разом з цим мають бути реалізовані великі проєкти, що працюють на основі ВДЕ.

У зв'язку зі збільшенням кількості генеруючих енергооб'єктів середньої та малої потужності структура електричних мереж також перетворюється. Все менше зустрічаються проєкти довгих ліній електропередачі високого та надвисокого класів напруги. Замість них з'являються розподільні мережі з напругою до 220 кВ, які з'єднують споживачів та джерела енергії.

Сучасні децентралізовані електроенергетичні системи з високою часткою розподіленої генерації ВДЕ будуть перетворені від категорії «споживання» до категорії «споживання та генерація».

Такі системи, попри всі свої переваги, мають і недоліки. Основні, які можна виділити, це перевантаження, викликані нерівномірним розташуванням

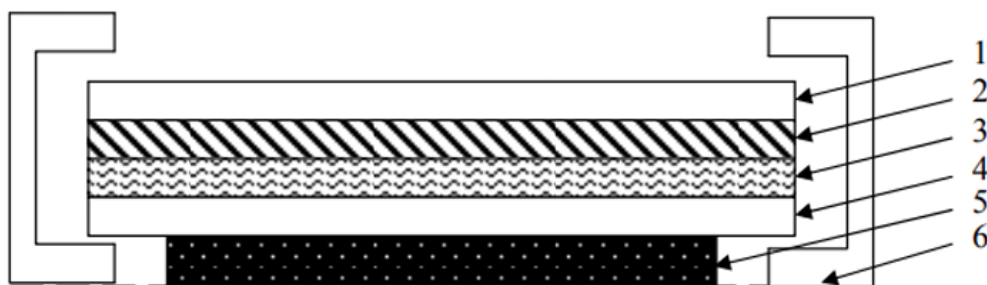
об'єктів генерації. Заходами для усунення незбалансованого розташування ВДЕ можна вважати розробку систем прогнозування виникнення перевантажень перерізів. Для успішного застосування таких підходів потрібно в першу чергу вдосконалювати інструменти прогнозування генерації ВДЕ.

1.2 Принципи роботи сонячних станцій та їх особливості

Історія винаходу сонячних панелей пов'язана з роботами французького фізика Александра Беккереля (1820–1891), який у 1839 році відкрив фотоелектричний ефект. Принцип дії сучасних сонячних панелей, заснований на цьому ефекті, описаний у роботі [6].

При потраплянні сонячних променів на поверхню фотоелементів фотони світла взаємодіють з атомами кремнію, вибиваючи електрони з їхніх орбіт. Цей рух електронів створює електричний струм. На початку процесу відбувається поглинання сонячного світла, коли фотоелементи вловлюють енергію сонячних променів. Після цього на межі позитивно і негативно заряджених шарів кремнію формується електричне поле, яке спрямовує рух електронів. Електрони ж генерують постійний електричний струм, який надходить до інвертора для перетворення на змінний струм.

Зазвичай сонячний модуль виконується у вигляді панелі, укладеної в каркас з анодованого алюмінієвого профілю, і представлений на рисунку 1.2.



1 – верхній шар покриття (скло або пластик); 2 – шар клею; 3 – полімерне ущільнення; 4 – герметизуюча плівка; 5 – кремнієвий сонячний елемент; 6 – каркас.

Рисунок 1.2 – Каркасний фотоелектричний модуль

Відомо, що генерація електроенергії та рух сонця тісно пов'язані між собою. Станції, що працюють на принципі фотоелектричного ефекту, знижують питому вартість електроенергії та мають значну гнучкість при виборі місця для встановлення, тим самим забезпечуючи домінуюче становище у сфері технологій ВДЕ [7–11].

Існує кілька основних типів фотоелектричних станцій [12], кожен з яких характеризується власними сферами застосування, перевагами та недоліками:

«Сонячні ферми»: великі установки, що складаються з об'єднаних сонячних панелей, розташованих на одній території. Вони призначені для масштабного отримання електроенергії та можуть розміщуватися як на відкритих майданчиках, так і на спеціально виділених ділянках. Основним недоліком є висока вартість будівництва та обслуговування системи.



Рисунок 1.3 – Приклад сонячної ферми

Гібридні або комбіновані СЕС: з іншими джерелами енергії, за типом турбін або дизельних генераторів. Від сонячних батарей енергія надходить одразу до споживача. Якщо не вистачає ресурсів, енергія буде отримана від акумуляторів [13]. Таким чином відбувається безперервне забезпечення

					КНУ КРБ.151.26.03.01.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

енергопостачанням, коли сонячна енергія недоступна. Перевагами цієї схеми можна вважати: надійне енергопостачання за рахунок комбінування різних джерел енергії, можливість використання різних джерел енергії залежно від умов та потреб у конкретний час, ефективне використання доступних ресурсів. До недоліків можна віднести складність проектування та налаштування гібридних систем через необхідність інтеграції різних джерел енергії, додаткові витрати на обслуговування та технічне обслуговування різних компонентів системи.

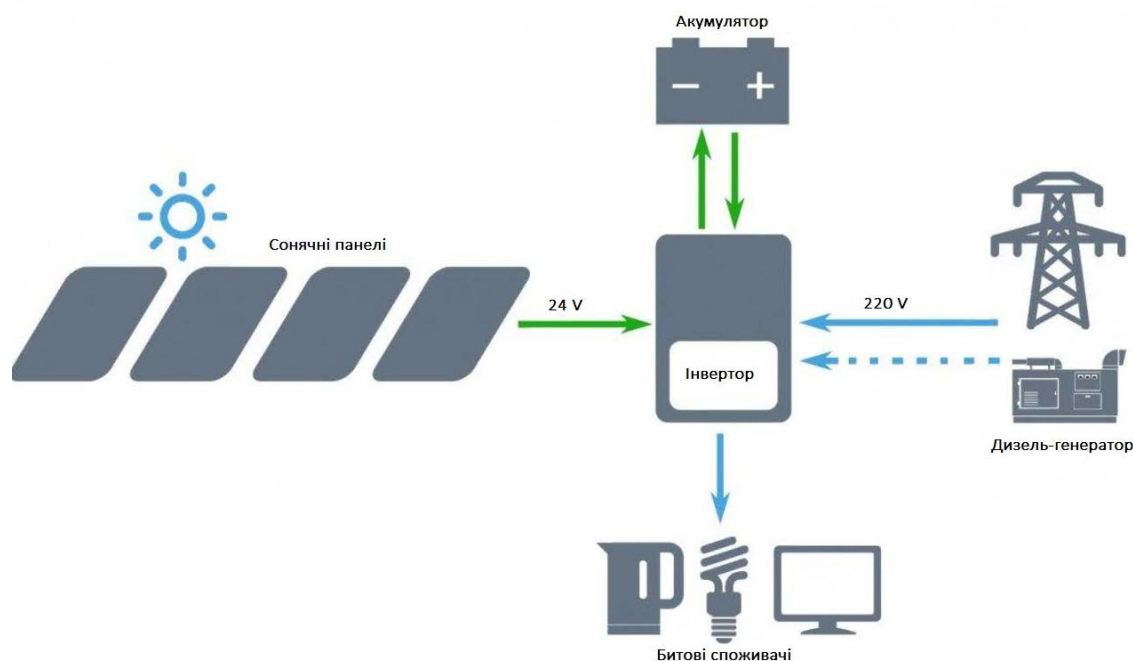
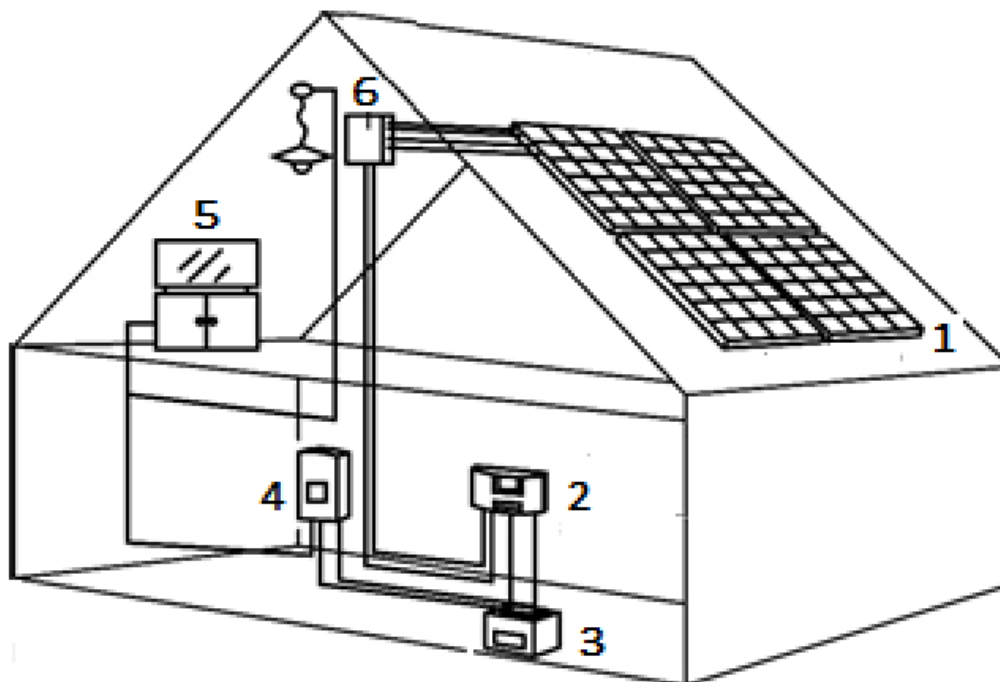


Рисунок 1.4 – Приклад гібридної системи

Окремі фотоелектричні станції. Дана схема використовується для забезпечення електроенергією окремих будівель або об'єктів. Зазвичай на даху будівлі встановлюються сонячні панелі, які генерують електрику для споживання всередині будівлі або для підключення до мережі. Перевагами таких станцій є: легке встановлення, мінімальні втрати енергії при передачі електрики на короткі відстані до споживача, можливість незалежного енергопостачання для окремих будівель або об'єктів. А недоліками – обмежена потужність і продуктивність порівняно з великими сонячними фермами, можливі труднощі з технічним обслуговуванням на висоті будівлі.



1- генератор; 2 – контролер сонячного заряду; 3- акумулятор; 4- інвертор;
5 – споживач (220В); 6 – з'єднувальна коробка

Рисунок 1.5 – Приклад окремої фотоелектричної станції

Різні типи фотоелектричних станцій дозволяють підібрати оптимальне рішення залежно від потреб та умов: «сонячні ферми» підходять для масштабного виробництва електроенергії, гібридні системи забезпечують стабільність за несприятливих погодних умов, а окремі станції є оптимальними для невеликих об'єктів з низьким енергоспоживанням.

1.3 Вплив зовнішніх факторів на вироблення електроенергії

Ключовими зовнішніми факторами, що визначають продуктивність та стабільність роботи сонячних панелей, є [14–17]:

- Непостійність сонячного випромінювання, викликана зміною інтенсивності та часу доби. Це вимагає розробки та впровадження систем акумулювання енергії або інтеграції з іншими джерелами в період низької або відсутньої сонячної активності.

- Погодні умови, такі як хмарність, дощі, снігопади та інші

метеорологічні явища, значно знижують отримані результати. Навіть часткове затінення сонячних панелей призводить до різкого падіння потужності.

– Сезонні коливання, коли рівень сонячної радіації змінюється залежно від пори року. У зимовий період кількість сонячних годин зменшується, що знижує середню потужність сонячних станцій.

– Ефективність роботи сонячних електростанцій залежить від розташування. Станції в регіонах з високою сонячною активністю, такі як пустелі, працюватимуть набагато стабільніше, ніж в областях, де спостерігаються часті опади та присутні короткі світлові дні.

Усі вище перелічені фактори роблять сонячну енергію менш прагматичною порівняно з «традиційними» джерелами. Це створює серйозні проблеми для операторів енергосистем, які повинні підтримувати баланс між попитом та пропозицією електроенергії в режимі реального часу.

Для зниження негативного впливу хмарності застосовуються різні методи прогнозування, засновані на даних метеорологічних супутників та математичних моделях. Це дозволяє заздалегідь оцінити можливі обсяги вироблення електроенергії та скоригувати роботу інших елементів енергосистеми.

Унаслідок залежності виробництва електроенергії від метеорологічних умов, об'єкти відновлюваної енергетики (ВДЕ) розміщуються на території України нерівномірно, з урахуванням найбільш сприятливих природно-кліматичних умов. Зокрема, найбільша концентрація потужностей вітрової та сонячної генерації спостерігається в південних і південно-східних регіонах країни (Одеська, Миколаївська, Херсонська, Запорізька області), де рівень сонячної радіації та вітровий потенціал є вищими. Водночас у центральних та північних регіонах частка таких потужностей є меншою через нижчу інтенсивність відповідних природних ресурсів та інші обмежувальні фактори розвитку енергетичної інфраструктури.

Подібне розміщення енергооб'єктів на основі ВДЕ характерне не лише для України, а й для країн Європейського Союзу. При розміщенні об'єктів

					КНУ КРБ.151.26.03.01.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

генерації ВДЕ на відносно невеликих територіях також враховуються кліматичні фактори.

Зі зростанням впливу ВДЕ на світовий енергетичний баланс світова енергетична спільнота розробила комплекс заходів для забезпечення стабільної та надійної роботи енергосистем. Коли внесок ВДЕ в загальне вироблення електроенергії залишається незначним (менше 1%), особливих заходів не потрібно. Однак, щойно частка ВДЕ досягає 1–10%, необхідно активізувати зусилля щодо вдосконалення методів прогнозування вироблення енергії з ВДЕ, а також посилити пропускну спроможність електромереж, що передають потужність від цих джерел. Це дозволить ефективно інтегрувати ВДЕ в існуючу енергосистему та забезпечити надійне електропостачання.

1.3.1 Великі дані та роль прогнозування в роботі сонячних електростанцій

Великі дані – це об'ємні масиви інформації, що характеризуються високою швидкістю надходження та різноманітністю своїх форматів. Джерелами великих даних для прогнозування роботи СЕС є різні джерела даних. Прикладами таких даних можна вважати дані із супутників, метеостанцій, дані з датчиків сонячних панелей тощо.

Використання великих даних можна описати як:

- Попередня обробка даних;
- Очищення даних від шумів;
- Аналіз та виявлення залежностей;
- Урахування сезонних та добових коливань;
- Побудова моделей машинного навчання.

Перевагами використання великих даних можна вважати підвищену точність, можливість побудови довгострокових прогнозів, а також адаптивність моделей до змін умов. Перспективами для розвитку можна вважати використання технологій штучного інтелекту для автоматизації та інтеграції даних з нових джерел.

					КНУ КРБ.151.26.03.01.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Крім великих даних, важливою частиною в роботі СЕС є сонячна енергія. Це невичерпний ресурс електроенергії, але його впровадження також ускладнене через низку технічних та операційних проблем. Сучасні алгоритми машинного навчання використовують потужні обчислювальні ресурси та можуть аналізувати великі обсяги даних, які впливають на вироблення [19]. Це дозволяє побудувати індивідуальні прогнози для кожної станції, що призводить до оптимізованого управління та знижує залежність від викопного палива.

У висновку можна зазначити, що використання великих даних та машинного навчання дозволяє значно підвищити точність прогнозів та забезпечити надійне енергопостачання. Однак потрібно розуміти, що для успішного впровадження таких технологій необхідно вирішувати питання, пов'язані з якістю отриманих даних.

1.4 Методи прогнозування вироблення електроенергії

Методи прогнозування вироблення сонячної електроенергії відіграють важливу роль в оптимізації роботи сонячних електростанцій, плануванні енергоспоживання та зниженні витрат на зберігання енергії. Нижче розглянуто основні методи прогнозування, їхні переваги та обмеження, а також їх застосування в реальних умовах [20–23].

Статистичні методи: засновані на аналізі історичних даних про погодні умови та вироблення сонячної електроенергії для передбачення майбутнього вироблення. Вони зазвичай вимагають наявності історичних, чистих та оброблених даних, кореляції між змінними тощо.

Фізичні моделі: засновані на фізичних законах та рівняннях, що описують процес перетворення сонячної енергії в електричну. Ці моделі використовують дані про погодні умови, такі як хмарність, температура, вологість тощо, щоб передбачити інсоляцію на поверхні землі, після чого обчислюється вироблення сонячної електроенергії. Фізичні моделі можуть бути досить точними, але вимагають великого обсягу обчислень і є складними в реалізації.

					КНУ КРБ.151.26.03.01.ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

Методи машинного навчання: включають алгоритми машинного навчання, такі як лінійна регресія, дерева рішень, випадковий ліс, нейронні мережі тощо [24]. Ці методи використовують історическі дані про погодні умови та вироблення сонячної електроенергії для навчання моделі, яка потім може використовуватися для прогнозування майбутнього вироблення. Методи машинного навчання можуть бути досить точними та гнучкими, але вимагають великого обсягу даних для навчання моделі.

Гібридні методи: комбінують у собі статистичні, фізичні та методи машинного навчання для прогнозування вироблення сонячної електроенергії. Гібридні методи можуть бути точнішими, ніж окремі методи, але вимагають більшого обсягу обчислень і можуть бути складними в реалізації.

Кожен із цих методів має свої переваги та недоліки, і вибір методу залежить від доступних даних, необхідної точності прогнозування та доступних обчислювальних ресурсів.

1.5 Математичний опис генерації сонячних панелей

Розглядаючи математичний опис генерації енергії сонячних панелей, необхідно враховувати такі фізичні закони, як фотоелектричний ефект та електронно-дірковий перехід, які перетворюють сонячну енергію. Ці закони стають основою для проєктування, оптимізації та експлуатації сонячних електростанцій, забезпечуючи високу точність розрахунків та можливість прогнозування.

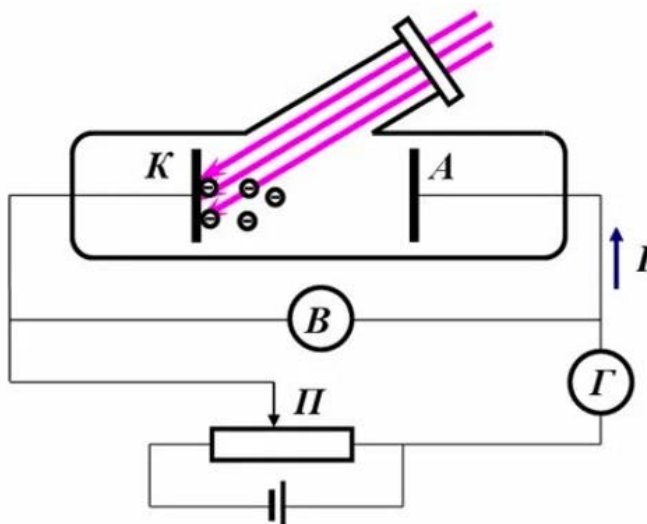
Фотоелектричний ефект був уперше пояснений Альбертом Ейнштейном у 1905 році. Суть явища полягає в тому, що при потраплянні фотонів (частинок світла) на поверхню напівпровідникового матеріалу їхня енергія передається електронам, які перебувають в атомах матеріалу. Якщо енергія фотона є достатньою для подолання енергетичного бар'єра, електрон переходить зі зв'язаного стану у вільний, створюючи електричний струм.

Енергія фотона може бути виражена через довжину хвилі або через імпульс фотона.

Формула з урахуванням довжини хвилі:

$$E = \frac{hc}{\lambda} \quad (1.1)$$

де E – енергія фотона; h – стала Планка ($6.63 \cdot 10^{-34}$ Дж·с); c – швидкість світла (м/с); λ – довжина хвилі (м).



К – катод; А – анод; П – потенціометр; В – вольтметр; Г – гальванометр

Рисунок 1.6 – Явище фотоефекту

Формула з урахуванням імпульсу:

$$E = pc \quad (1.2)$$

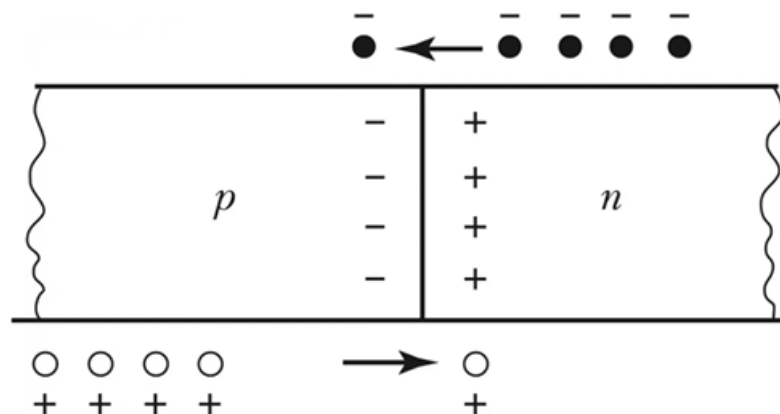
де E – енергія фотона; c – швидкість світла (м/с); p – імпульс фотона.

Електронно-дірковий перехід або р-п перехід у напівпровідниках. На межі між р- та п- шарами виникає електричне поле, яке розділяє фотони та електрони. Коли відбувається звільнення електронів в області р-п переходу, вони рухаються під дією електричного поля, що й створює електричний струм.

Напівпровідники мають два типи провідності:

- р-тип характеризується наявністю позитивно заряджених носіїв (дірки).
- п-тип характеризується надлишком негативно заряджених носіїв (електрон).

Дані принципи дозволяють зрозуміти, як світлова енергія перетворюється на електрику, і чому важливо враховувати такі параметри, як інтенсивність сонячного випромінювання, кут падіння світла, температура та характеристики матеріалів.



● - електрони; ○ - дірки

Рисунок 1.7 – Схема р-n-переходу

Розміри сонячних елементів можуть варіюватися, розглянемо сонячну панель із номінальною потужністю 250 Вт. Потужність, що отримується від сонця, залежить від:

- Інтенсивності сонячного випромінювання (G): це величина, яка характеризує кількість енергії, що падає на одиницю площі за одиницю часу.
- Кута падіння сонячних променів (θ): кут між нормаллю до поверхні панелі та напрямком сонячних променів.
- Коефіцієнта прозорості атмосфери (τ): урахування впливу атмосфери на поглинання та розсіювання сонячного випромінювання.

Потужність сонячного випромінювання можна записати як:

$$P_0 = G_0 * \cos(\theta) * \tau \quad (1.3)$$

де G_0 – інтенсивність сонячного випромінювання поза атмосферою; τ – коефіцієнт прозорості атмосфери; $\cos(\theta)$ – косинус кута падіння.

Виходячи з цього, можна знайти вихідну потужність сонячної панелі (P_{out}):

$$P_{out} = \eta * A * P_0 \quad (1.4)$$

					КНУ КРБ.151.26.03.01.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

де η – ККД сонячної панелі; A – площа панелі в квадратних метрах; P_0 – потужність сонячного випромінювання.

Не можна не враховувати температуру, яка постійно змінюється і призводить до похибки. Температурний коефіцієнт сонячної панелі (β) показує відсоток ефективності панелі при підвищенні температури повітря на 1 градус. Середній температурний коефіцієнт більшості панелей коливається в межах 0.005°C . Можна сказати, що при підвищенні температури на 20 градусів втрати становитимуть від 5.6 до 10%, для спрощення розрахунків буде використовуватися 10%.

Виразимо формулу ККД з урахуванням температури:

$$\eta_t = \eta * (1 - \beta(t - t_0)) \quad (1.5)$$

де η – номінальний ККД; β – коефіцієнт сонячної панелі; t_0 – середня температура (26°C); t – поточна температура.

На підставі перелічених формул можна отримати загальний вираз для вихідної потужності сонячної панелі:

$$P = \eta_t * A * P_0 \quad (1.6)$$

Нижче представлено приклад розрахунку сонячної панелі. Для панелей потужністю близько 250 Вт площа однієї панелі (A) становить 1.6 м^2 . Біля поверхні Землі величина інтенсивності сонячної радіації становить (P_0) 635 Вт/м^2 . У ясний сонячний день ця величина коливається від 950 до 1220 Вт/м^2 . Середнє значення можна взяти 1000 Вт/м^2 . Залежно від типу сонячних панелей їхній ККД може відрізнятися, середнє значення буде близько 20%. Середня температура сонячної панелі в реальних умовах становить 42.5°C . Необхідно враховувати втрати. Загальний коефіцієнт втрат $K_{\text{пот}}$ у середньому становить 0.8.

Підставивши всі наявні значення у фінальну формулу, можна отримати таке:

$$P = \eta * (1 - \beta(t - t_0)) * A * G_0 * \cos(\theta) * \tau * K_{\text{пот}} \quad (1.7)$$

$$P = (0.2 * (1 - 0.005(42.5 - 26)) * 1.6 * 1000) * 0.8 = 234.88 \text{ Вт.}$$

					КНУ КРБ.151.26.03.01.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

Математичний опис роботи сонячних панелей на основі фотоелектричного ефекту та характеристики р-п переходу, в якому враховано вплив інтенсивності сонячного випромінювання, кута падіння світла, температури та коефіцієнтів втрат, дозволив визначити різницю номінальної потужності 250 Вт і фактичної 234.88 Вт, що свідчить про важливість урахування реальних умов для точності прогнозування й оптимізації роботи системи.

Висновки за розділом:

У першому розділі роботи було розглянуто сучасний стан та тенденції розвитку сонячної енергетики, зокрема зростаючу роль фотоелектричних станцій у структурі енергосистем. Показано, що інтеграція СЕС у сучасні енергетичні мережі потребує врахування їхньої високої залежності від метеорологічних умов та природних факторів, що зумовлює необхідність використання сучасних методів аналізу даних і прогнозування. Обґрунтовано, що застосування технологій машинного навчання та методів обробки великих даних є ключовим інструментом підвищення ефективності управління «зеленою» генерацією. Це дозволяє зменшити залежність енергосистем від традиційних джерел генерації, забезпечити більш стабільний баланс між виробництвом і споживанням електроенергії та підвищити загальну енергетичну безпеку.

					КНУ КРБ.151.26.03.01.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

РОЗДІЛ 2

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ КОРОТКОСТРОКОВОГО ПРОГНОЗУВАННЯ СОНЯЧНОЇ ГЕНЕРАЦІЇ

2.1 Метрики для аналізу якості роботи алгоритмів

Аспекти машинного навчання, які є основною частиною технологій штучного інтелекту, включають у себе навчання на вибірках, підбір метрик та аналіз помилок. Перш ніж розглянути метрики, потрібно розібрати загальні терміни:

- Метрики – показники, які використовуються для моніторингу та аналізу продуктивності систем, процесів або алгоритмів. Вони дозволяють виявляти відхилення від заданих значень, визначати, що необхідно оптимізувати, та оцінити вплив внесених змін.
- Навчання – процес, коли модель налаштовує свої параметри, спираючись на дані, для мінімізації помилок і максимізації точності.
- Дані – є головним ресурсом для машинного навчання. Вони бувають структурованими (таблиці) та неструктурованими (текст, зображення).
- Модель – алгоритм або математичне представлення системи, яке навчається на даних.
- Ознаки – характеристики даних, які використовуються для навчання моделі. Прикладами ознак можна вважати температуру навколишнього середовища або час доби.
- Викиди (аномалії) – це точки даних, які відрізняються від загальної маси даних.

					КНУ КРБ.151.26.03.02.ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Галушка В. О.			РОЗДІЛ 2			Літ.	Арк.	Аркушів	
Перевір.		Жосан А. А.								28	30
Н. Контр.		Маринич І.А.			АКНТ АКІТ-22						
Затверд.		Маринич І.А.									

- Цільова змінна – передбачення моделі.
- Оцінка моделі – оцінка відбувається за допомогою метрик якості (точність або середньоквадратична помилка).
- Гіперпараметри – параметри моделі, які налаштовуються перед навчанням.
- Перенавчання – коли модель підлаштовується під тренувальні дані, що призводить до запам'ятовування зайвих деталей та «шумів».
- Недонавчання – модель занадто проста для врахування базових закономірностей у даних. Може відбуватися через брак складності або гнучкості, необхідних для представлення взаємозв'язків.

Побудова моделей машинного навчання неможлива без оцінювання якості їхньої роботи. Для задач навчання з вчителем – класифікації та регресії – існує кілька основних показників якості. При цьому кожен із них показує різні характеристики, тому необхідно точно знати спосіб розрахунку кожного з них.

2.1.1 Бінарна класифікація класів

Розглянемо на прикладі бінарної класифікації двох класів: Р (позитивний результат) і N (негативний результат).

Матриця помилок – показник, який є не одним числом, а цілою матрицею.

Таблиця 2.1 – Матриця помилок

		Істинний клас (розмітка)	
		Р	N
Передбачений клас (результат моделі)	Р	ТР=число або % Істинно-позитивних результатів	ФР=число або % хибно-позитивних результатів
	N	FN =число або % хибно-негативних результатів	TN=число або % істинно-негативних результатів

$$A = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (2.1)$$

Точність є інтегрованим показником, що враховує помилки як хибно-негативні, так і хибно-позитивні. Але якщо вибірка не збалансована за класами, то показник може бути нерелевантним. Наприклад, розглянемо таку матрицю помилок:

Таблиця 2.2 – Приклад незбалансованої матриці помилок

		Істинний клас (розмітка)	
		P	N
Передбачений клас (результат моделі)	P	50	50
	N	100	5000

Якщо модель видала клас P, то насправді це може бути як P, так і N, ймовірність помилки 50%. Більше того, якщо насправді клас P, то модель скоріше віднесе його до класу N ($100/150 = 0.667$), ймовірність помилки 67%. Видно, що якість моделі дуже низька. Але у вибірці об'єктів класу N у рази більше, ніж об'єктів класу P, і вони класифікуються моделлю майже завжди правильно (5000 з 5050). У підсумку точність буде близько 97%.

Точність позитивного передбачення (PPV):

$$PPV = TP / (TP + FP) \quad (2.2)$$

Враховує хибно-позитивні помилки, виходячи з прикладу вище, можна порахувати отримане значення:

$$PPV = \frac{50}{50 + 50} = 50\%$$

Істинно-позитивний показник (TPR):

$$TRP = TP / (TP + FN) \quad (2.3)$$

Враховує хибно-негативні помилки, все також виходячи з прикладу вище, можна порахувати отримане значення:

$$TRP = \frac{50}{50 + 100} = 33\%$$

Видно, що в даному прикладі PPV та TRP, на відміну від точності, показують низьку якість.

Симетрично цим двом показникам є NPV та TNR:

$$NPV = TN / (TN + FN) \quad (2.4)$$

$$TNR = TN / (TN + FP) \quad (2.5)$$

Значення, що виходять у прикладі: NPV = 98%, TNR = 99%.

F_β-міра – цей показник балансує TRP та PPV, дозволяючи об'єднати їх в один, при цьому коефіцієнт β задає, у скільки разів TRP важливіший, ніж PPV. Якщо β = 1, то показники мають однакову значущість.

$$F_{\beta} = (1 + \beta)^2 \frac{PPV * TRP}{\beta^2 PPV + TRP} = (1 + \beta)^2 \frac{TP}{(1 + \beta)^2 TP + \beta^2 FN + FP} \quad (2.6)$$

2.1.2 Показники якості регресії

Нехай є ряд істинних значень випадкової величини $Y = y_1, y_2, \dots, y_n$ та передбачений регресійною моделлю ряд $U = u_1, u_2, \dots, u_n$.

Середній модуль помилки (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - u_i| \quad (2.7)$$

MAE вимірюється в тій самій розмірності, що й прогнозована величина. Її значення менш схильне до впливу викидів.

Середній квадрат помилки (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - u_i)^2 \quad (2.8)$$

Цей показник підвищує важливість великих відхилень. Але його недолік у тому, що він показує помилку в квадратних значеннях, а не в тих самих одиницях виміру, що й величини y та u .

Корінь із середнього квадрата помилки (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - u_i)^2} \quad (2.9)$$

Корінь із середнього квадрата помилки краще використовувати замість MSE, тому що він вимірює помилку в тій самій розмірності, в якій будується прогноз.

Середня по модулю відносна помилка (*MAPE*):

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - u_i}{y_i} \right| * 100\% \quad (2.10)$$

Показує середній відсоток відхилення прогнозованих значень від фактичних значень.

Коефіцієнт детермінації (R^2):

$$R^2 = 1 - \sum_{i=1}^n \frac{(X_i - Y_i)^2}{(X_i - Y)^2} \quad (2.11)$$

Вимірює, яка частка варіативності залежної змінної пояснюється незалежними змінними в моделі. Відмінний спосіб для оцінки адекватності моделі. Якщо показник буде близький до 1, це свідчить про хороше пояснення даних моделлю. Коефіцієнт детермінації найкраще підійде, якщо потрібно порівняти кілька моделей.

2.2 Особливості застосування методів машинного навчання

Успішне впровадження машинного навчання залежить від вибору правильного алгоритму, якості даних, їхнього налаштування та розуміння суті роботи моделі. Якість даних впливає на виробничу потужність та надійність моделі машинного навчання. Етап, на якому відбувається підготовка даних, включає в себе такі важливі кроки:

Очищення даних, де відбувається видалення викидів, які можуть суттєво спотворити результати моделювання. Викиди можуть бути результатом помилок вимірювання, аномалій або просто рідкісних подій. Також необхідно заповнити пропущені значення, використовуючи різні методи.

Нормалізація даних, що полягає в їхньому приведенні до єдиного масштабу, є важливим кроком, особливо для алгоритмів, чутливих до масштабу ознак.

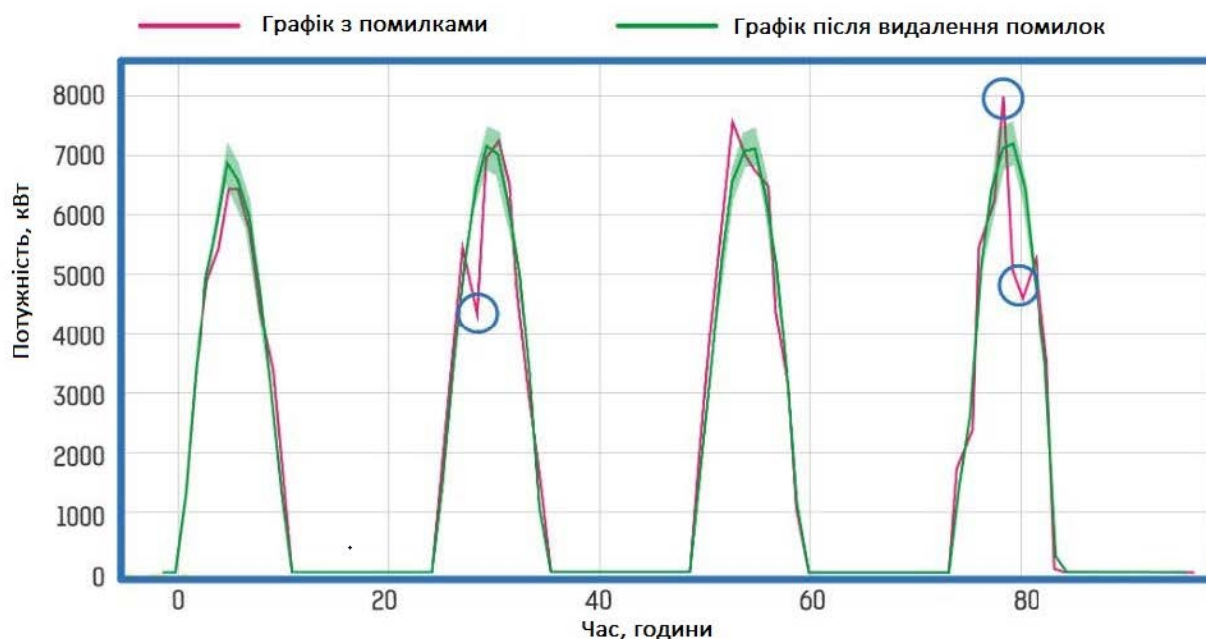


Рисунок 2.1 – Графік генерації СЕС із спотвореннями та після їх видалення

Як видно з графіка, представленого на рисунку 2.1, через спотворення значення функції втрат при навчанні помітна втрата точності, і таким чином погіршиться збіжність процесу навчання. Щоб уникнути описаних негативних ефектів, необхідно до початку навчання моделі виконати аналіз даних із використанням статистичних методів виявлення викидів та експертного аналізу результатів візуалізації даних на валідаційній вибірці.

Вибір та створення ознак – етап важливий і такий, що вимагає розуміння предметної області. Підбираються найбільш підходящі ознаки з існуючих даних, а також створюються нові ознаки, які сприяють покращенню продуктивності моделі. Успішний вибір ознак спрощує шлях до досягнення високої точності моделі. Дата буде чудовим прикладом, оскільки з неї можна взяти день тижня, час або місяць.

Існують методи налаштування гіперпараметрів, такі як перебір сітки (GridSearch) та випадковий пошук (RandomSearch). Перебір сітки має на увазі перевірку всіх можливих комбінацій гіперпараметрів, що може зайняти деякий час. Випадковий пошук вибирає випадкові комбінації гіперпараметрів, що дозволяє знайти оптимальні значення за менший час.

Кросс-валідація – метод оцінки, який дозволяє уникнути перенавчання. При використанні крос-валідації дані розділяються на кілька частин, і модель навчається на одних частинах, а оцінюється на інших. Після оцінки кожної частини відбувається усереднення результату для отримання більш надійної оцінки.

Розглянуті особливості вимагають повного розуміння специфіки вирішуваної задачі, об'єкта, який розглядається в рамках задачі, принципів роботи математичних моделей, статистичних показників та алгоритмів. Прогалини в цих параметрах із високою ймовірністю призведуть до збільшення трудомісткості створення системи та знизять показник якості роботи моделі під час її експлуатації.

2.3 Опис моделей машинного навчання

Штучний інтелект (ШІ) – це міждисциплінарна галузь, що широко спирається на інформатику, науку про дані, природничі та гуманітарні науки, математику, суспільні та інші науки. Машинне навчання (МО) – це один із типів ШІ, в якому модель формується під час навчання, спираючись на певний набір даних. Таке навчання має широке застосування в різних галузях, таких як енергетика, економіка, медицина тощо. Можна виділити деякі основні алгоритми, такі як метод опорних векторів (SVM), метод найближчих сусідів (kNN), беггінг (bagging) та бустинг (boosting), метод лінійної регресії (Linear Regression), метод дерева рішень (Decision trees), метод випадкового лісу (Random Forest), метод підбору оптимальних гіперпараметрів (GridSearch) та

метод екстремального градієнтного бустингу (XGBoost). Розглянемо їх детальніше.

2.3.1 Метод опорних векторів (SVM)

Полягає у визначенні, до якого класу з як мінімум двох спочатку відомих належить даний об'єкт. Зазвичай таким об'єктом є вектор у n -вимірному дійсному просторі R_n . Координати вектора описують окремі атрибути об'єкта.

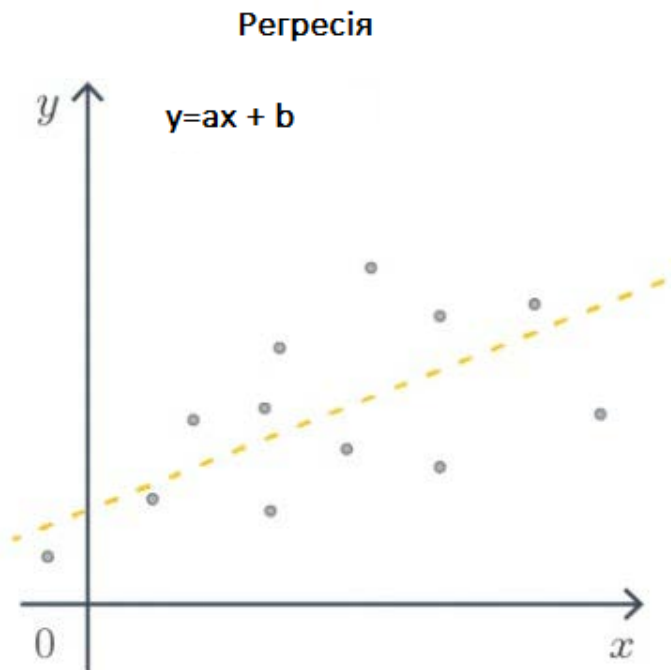
Простими словами, даний метод шукає, як провести дві прямі між категоріями, щоб між ними утворився найбільший зазор. Якщо класів усього два, то задача називається бінарною класифікацією. Якщо більше – це вважатиметься мультикласовою класифікацією. Також можуть бути наявні зразки кожного класу – об'єкти, про які заздалегідь відомо, до якого класу вони належать. Такі задачі називаються навчанням із учителем, а відомі дані називаються навчальною вибіркою. Метод опорних векторів належить до навчання з учителем. Отже, математичне формулювання задачі класифікації таке: нехай X – простір об'єктів (наприклад R_n), Y – класи. Дано навчальну вибірку: $(x_1, y_1), \dots, (x_m, y_m)$. Потрібно побудувати функцію $F = x \rightarrow y$ (класифікатор), що зіставляє клас y довільного об'єкта x . Ідею методу можна проілюструвати на такому простому прикладі: дано точки на площині, розбиті на два класи (рис. 2.2). Проведемо лінію, що розділяє ці два класи.

Далі всі нові точки (не з навчальної вибірки) автоматично класифікуються таким чином:

- точка вище прямої потрапляє в клас А,
- точка нижче прямої – в клас В.

Таку пряму назвемо розділяючою прямою. Однак у просторах високих розмірностей пряма вже не розділятиме наші класи, оскільки поняття «нижче прямої» або «вище прямої» втрачає будь-який сенс. Тому замість прямих необхідно розглядати гіперплощини простору, розмірність яких на одиницю менша, ніж розмірність вихідного простору. У прикладі існує кілька прямих, що розділяють два класи.

					КНУ КРБ.151.26.03.02.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		



x - (єдиний) признак; y - таргет

Рисунок 2.2 – Приклад роботи методу

З точки зору точності класифікації найкраще вибрати пряму, відстань від якої до кожного класу максимальна. Іншими словами, виберемо ту пряму, яка розділяє класи найкращим чином (червона пряма на рис. 2.3). Така пряма, а в загальному випадку гіперплощина, називається оптимальною розділяючою гіперплощиною. Вектори, що лежать найближче до розділяючої гіперплощини, називаються опорними векторами, на рисунку 2.3 вони позначені червоним.

2.3.2 Метод найближчих сусідів (KNN)

Досить простий, але потужний. Його можна застосовувати для вирішення задач регресії. KNN заснований на прикладах, які не вимагають припущень про розподіл даних, що лежать в його основі. Цей алгоритм працює на основі k числа найближчих сусідів точки, яка нас цікавить. K найближчими сусідами точки даних b є k точок, які розташовані на найкоротшій відстані від b . Вирахування для алгоритму KNN вимагає відстані між точками даних у

векторному просторі. Існує кілька показників, які можна використовувати для обчислення відстаней, одним із них є евклідова відстань.

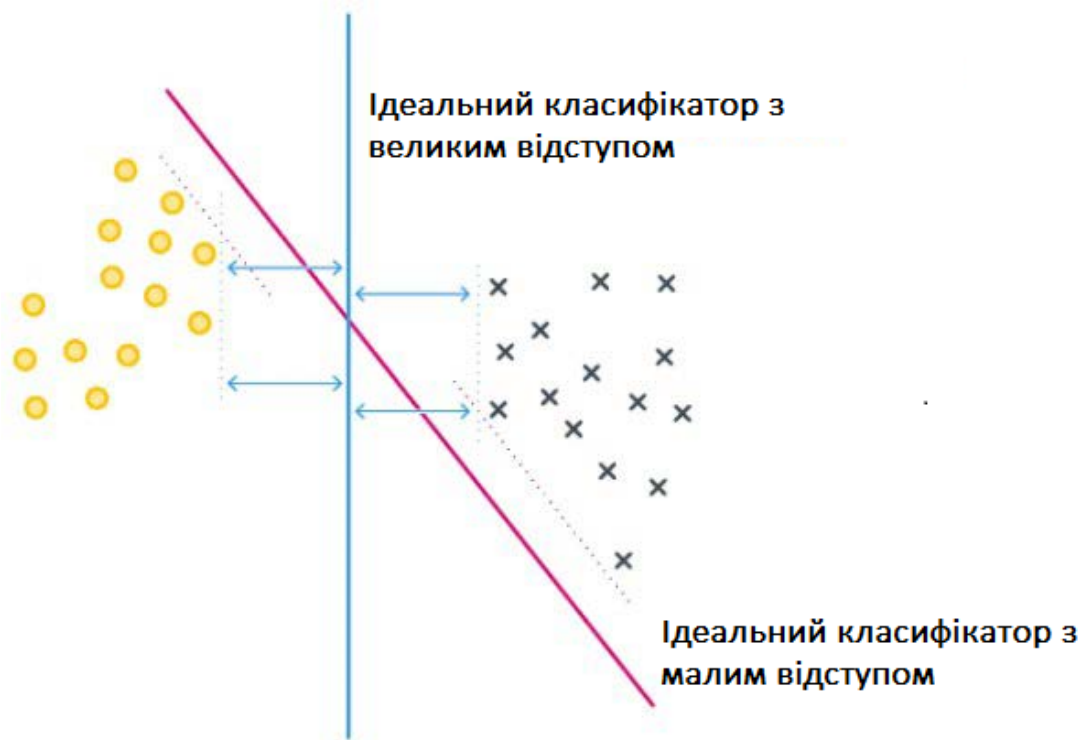


Рисунок 2.3 – Прямі, що розділяють на класи

Евклідова відстань між двома точками даних А та В визначається таким чином:

$$d(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (2.12)$$

Після обчислення відстані між відповідною точкою даних b та іншими точками даних відстані сортуються, і враховується тільки k точок даних із найменшою відстанню.

На рисунку 2.4 показано ілюстрацію трьох найближчих сусідів точок даних b_1 і b_2 .

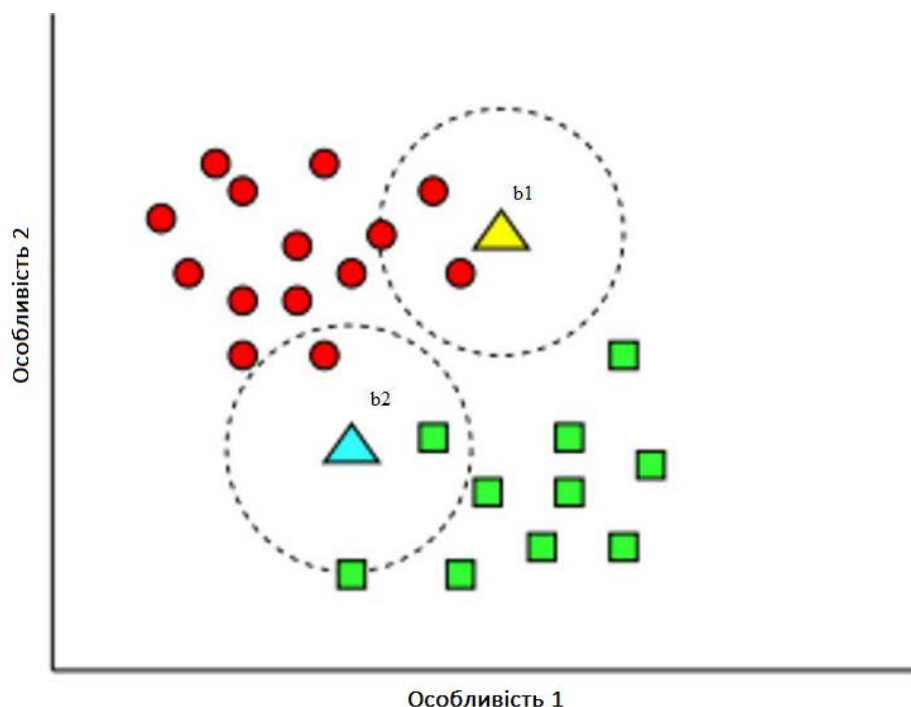


Рисунок 2.4 – Приклад трьох найближчих сусідів точок даних b1 і b2

Для задачі регресії алгоритм KNN обчислює вихідне значення на основі середнього значення k-найближчих сусідів таким чином:

$$y_b = \frac{1}{k} \sum_{i=1}^k Y_i(x) \quad (2.13)$$

де Y_b – вихідне значення відповідної точки даних b , а Y_i – вихідне значення точки даних a_i , яка знаходиться в k найближчих сусідів.

2.3.3 Метод лінійної регресії

Лінійна регресія – це різновид алгоритму машинного навчання з учителем, який навчається за допомогою наборів даних і описує шукану залежність у вигляді лінійної функції, яку можна використовувати для прогнозування на нових наборах даних. Він обчислює лінійну залежність між залежною та кількома незалежними змінними, підбираючи лінійне рівняння до спостережуваних даних.

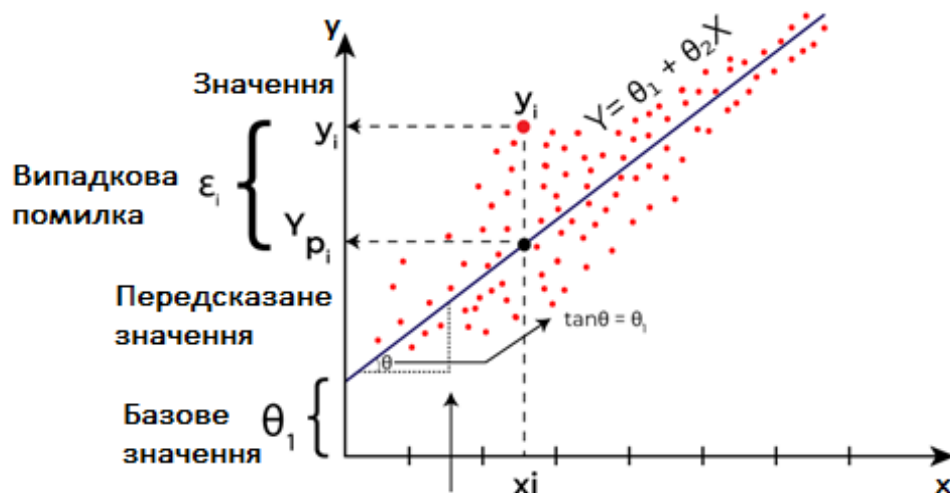


Рисунок 2.5 – Метод лінійної регресії. Де Y називається залежною змінною, а X незалежною змінною.

Основна мета при використанні лінійної регресії – знайти найбільш підходящу лінію, похибка між прогнозованими та фактичними значеннями має бути зведена до мінімуму. Рівняння найкращої апроксимації прямої являє собою пряму лінію, яка відображає взаємозв'язок між X і Y . Нахил прямої говорить про те, як змінюється залежна змінна при зміні незалежної змінної.

2.3.4 Метод дерев рішень

Являє собою ієрархічні моделі у вигляді правил «ЯКЩО...ТО...ІНАКШЕ». При навчанні дерева рішень для задач регресії вони групуються так, що навчальна вибірка ділиться на дві частини, а структура описується зверху вниз. Для побудови кожного чергового правила робиться статистична оцінка – яке правило найсильніше знизить рівень невизначеності. Дерева рішень часто об'єднують в ансамблі для підвищення точності та стійкості до перенавчання окремого регресійного дерева.

Є два найбільш поширених ансамблевих методи: беггінг (bagging) та бустинг (boosting). Бустинг дозволяє підвищити точність базової моделі та знижує помилку. Беггінг же корисний, якщо потрібна стабільність моделі, а також зниження її чутливості до перенавчання. При використанні беггінгу

береться одна модель, екземпляри якої безліч разів навчаються на випадкових вибірках на основі вихідних даних, після чого усереднює отриманий результат на виході в якості відповіді. Метод Random Forest заснований на принципах беггінгу. Він дозволяє побудувати безліч дерев рішень на випадкових вибірках і наборах ознак з наявних даних, після чого усереднює результат. Усе це дозволяє знизити ризик перенавчання моделі.

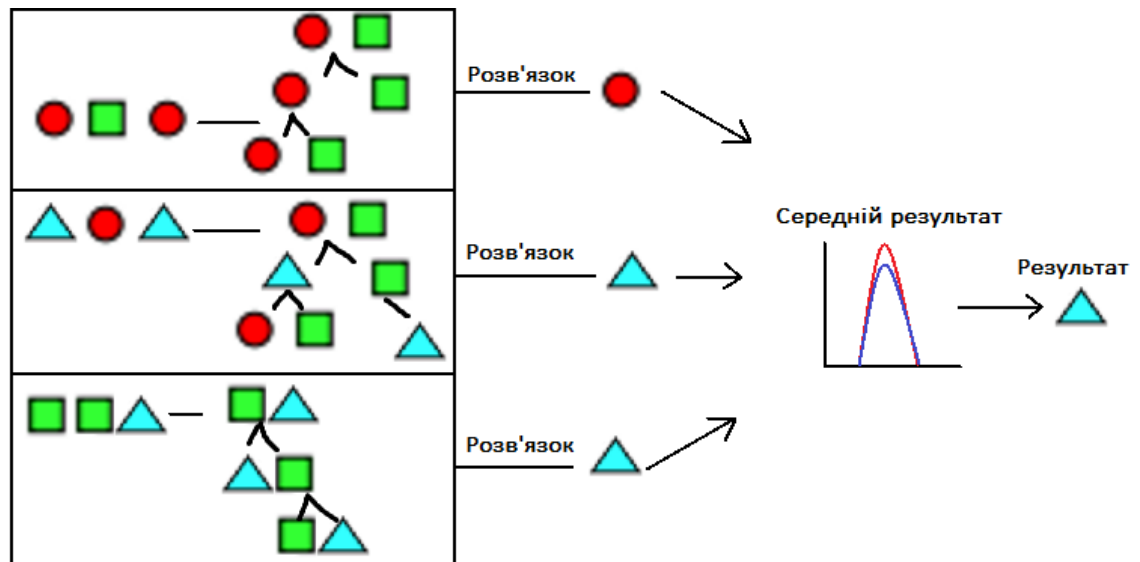


Рисунок 2.6 – Принцип роботи беггінгу

Бустинг же здатний навчити алгоритм, зберігаючи послідовність. У кожену нову вибірку беруться дані з минулої, в яких були помилки. Виходить навчання нової моделі на помилках попередніх. XGBoost (Extreme Gradient Boosting) – реалізація метода бустингу, в якому використовуються дерева рішень в якості основи для моделі.



Рисунок 2.7 – Принцип роботи бустингу

При аналізі даних можна знайти найбільш підходящі гіперпараметри. Бібліотеки в Python мають певний набір гіперпараметрів, які добре працюють

з більшістю моделей, но вони можуть виявитися не найкращими. Метод "ручного" перебору – це спосіб знайти найбільш підходящі гіперпараметри, що і є основою оптимізації гіперпараметрів (GridSearch).

2.3.5 GridSearch

Це метод, який передбачає пошук у діапазоні значень, щоб знайти підмножини результатів, забезпечуючи найкращу продуктивність на необхідному наборі даних. У якості прикладу можна сказати, що при перевірці матриці 3×3 даний метод пройшовся по кожному значенню цієї матриці і вибрав найбільш підходящий варіант.

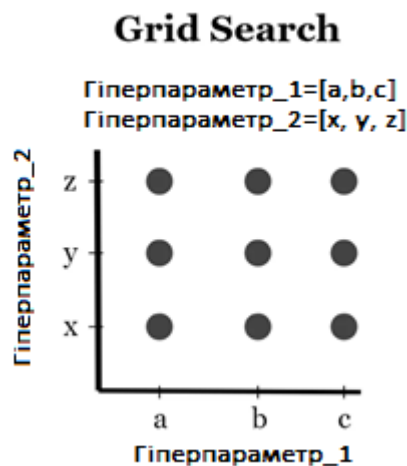


Рисунок 2.8 – Принцип роботи GridSearch

2.3.6 RandomSearch

Для кожного гіперпараметра задається розподіл, з якого вибирається його значення, і комбінація гіперпараметрів складається семплюванням з цих розподілів. Завдяки випадковому вибору чергової комбінації гіперпараметрів можна знайти оптимальну комбінацію за менше число ітерацій. Щоб не перевіряти всі точки в матриці 3×3 , даний метод випадковим чином вибирає кілька комбінацій із цієї матриці, навчає модель тільки на цих комбінаціях і вибирає найкращий результат.

Random Search

```
Гіперпараметр_1=random.num(range)
Гіперпараметр_2=random.num(range)
```

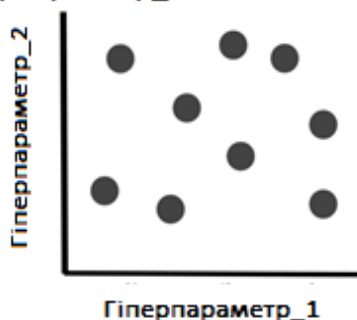


Рисунок 2.9 – Принцип роботи RandomSearch

Вибір між розглянутими методами ґрунтується на їхніх перевагах та недоліках, залежно від задачі та типу наявних даних. Використання машинного навчання дозволяє використовувати широкі можливості для підвищення ефективності роботи сонячних електростанцій та спрощення управління відновлюваними джерелами енергії.

2.4 Вибір програмного забезпечення для реалізації

Щоб реалізувати поставлене завдання, було обрано мову програмування Python, а також буде використано MATLAB.

MATLAB є високорівневою мовою програмування та інтерактивним середовищем для чисельних обчислень. Він широко використовується для аналізу та візуалізації даних, розробки алгоритмів, моделювання.

Особливості MATLAB:

- Інтеграція з Simulink для моделювання динамічних систем;
- Підтримка графіків та діаграм;
- Велика кількість спеціалізованих пакетів розширення;
- Синтаксис, який дозволяє швидко реалізовувати складні математичні операції;

- Підтримка взаємодії з іншими мовами програмування, такими як Python;
- Інтерактивне середовище, що дозволяє виконувати команди покроково та аналізувати результати в реальному часі.
- Python використовується в більшій частині цієї роботи і був обраний з таких причин:
 - Простота синтаксису;
 - Наявність безлічі бібліотек з широким функціоналом для машинного навчання.

Для Python існує безліч середовищ розробки, в межах даної роботи буде використано PyCharm. PyCharm – це потужне інтегроване середовище розробки (IDE), яке може забезпечити автономну роботу та широке налаштування конкретних завдань програмування. Дане середовище розробки можна встановити локально на комп'ютер, що дозволяє виключити залежність від інтернету. PyCharm також пропонує просунуті інструменти для управління проєктами, включаючи інтеграцію з системами контролю версій, підтримку віртуальних середовищ та автоматизоване вирішення залежностей. Така гнучкість робить його популярним серед розробників не лише у сфері аналізу даних, а й створення додатків та веб-розробки.

Можна виділити такі ключові особливості:

- Інтелектуальне автодоповнення та рефакторинг коду, які прискорюють написання програм.
- Інтеграція з відладчиком, що дозволяє покроково виконувати скрипти та аналізувати змінні.
- Вбудований термінал та інструменти для роботи з базами даних, які спрощують взаємодію з зовнішніми системами.
- Підтримка простих скриптів та enterprise-додатків.
- Відладка, профілювання, інтеграція з CI/CD.
- Незалежність від інтернету та хмарних обмежень.

PyCharm підходить для реалізації готових рішень, де важливі структурованість, документація та керованість кодової бази.

2.5 Опис вибірки та об'єкта даних

Об'єктом дослідження в роботі є сонячна електростанція (СЕС), розташована на території України - Нікопольська СЕС, яка знаходиться в Нікопольському районі Дніпропетровської області.

Нікопольський район належить до одного з найбільш перспективних регіонів України для розвитку відновлюваних джерел енергії. Клімат території характеризується як помірно континентальний із великою кількістю сонячних днів протягом року (у середньому 200–240 днів), значним рівнем сонячної інсоляції та відносно низькою хмарністю в теплий період. Такі умови є сприятливими для ефективного функціонування фотоелектричних станцій.

Нікопольська СЕС є однією з найбільших сонячних електростанцій в Україні та Європі. Вона введена в експлуатацію у 2019 році та має встановлену потужність близько 246 МВт [19]. Станція розташована на території колишнього промислового кар'єру площею близько 400 гектарів та використовує сотні тисяч фотоелектричних модулів для перетворення сонячної енергії в електричну [20].

Розвиток подібних об'єктів у регіоні є складовою частиною стратегії України щодо переходу до «зеленої» енергетики, підвищення енергетичної незалежності та зменшення викидів парникових газів. Введення в експлуатацію великих СЕС дозволяє інтегрувати відновлювану генерацію в об'єднану енергосистему України та підвищувати її стійкість.

Станція оснащена сучасними фотоелектричними модулями (переважно кремнієвими), інверторними станціями та автоматизованою системою моніторингу, яка забезпечує контроль параметрів роботи в реальному часі, діагностику відхилень та прогнозування генерації електроенергії.

					<i>КНУ КРБ.151.26.03.02.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

База даних, що використовується для аналізу, представляє собою часові ряди з годинною дискретизацією та містить інформацію про фактичну генерацію електроенергії СЕС, а також супутні метеорологічні параметри.

До основних параметрів відносяться:

- година доби;
- номер дня в році;
- кут сонячного схилення (КСС);
- сонячний час;
- щільність потоку сонячного випромінювання (ЩПСВ);
- хмарність, %;
- коефіцієнт прозорості атмосфери (КПА), %;
- температура повітря, °С;
- вологість повітря, %;
- швидкість вітру, м/с;
- фактична генерація електричної енергії, кВт·год.

2.6 Предобробка та підготовка даних для навчання

Вихідні дані не дозволяють одразу ж застосувати до них методи машинного навчання через такі причини, представлені нижче:

- Числові дані можуть бути представлені в нечисловому форматі, наприклад, замість числа «5» у полі таблиці вказано «5 кВт» або очікується число з десятковим роздільником «.», але у файлі використано роздільник «,»;
- Різні роздільники стовпців, наприклад, стовпці відокремлені один від одного символом «;», але якісь два різних параметри записані в один стовпець і розділені пробілом або комою;
- Пропущено значення, замість числа стоїть прочерк «-», знак питання «?», пробіл « » або будь-який інший символ;
- Викиди, помилкові значення, які записані в коректному форматі, але через помилку запису або інші причини дуже сильно вибиваються і

					<i>КНУ КРБ.151.26.03.02.ПЗ</i>	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

спотворюють статистичні показники, наприклад, у стовпці записані потужності в кВт, але окремі значення випадково були записані в МВт.



Рисунок 2.10 – Процес очищення даних

Процес очищення даних.

Даний процес складається з таких кроків:

- читання вихідного файлу;
- приведення стовпців до потрібних форматів;
- виявлення та видалення пропусків;
- виявлення та виправлення викидів;
- запис отриманого файлу.

Було проведено передобробку даних, спираючись на все вище перелічене. На рисунку нижче представлено фрагмент наявних даних до передобробки із пропусками та неоднорідними роздільниками.

Номер доби в році	Кут сонячного схилання	Сонячний час	ППСВ на Гр.Атм	Хмарність, %	Коефіцієнт прозорості, %	Температура, °C
269	-2,217	14,380	787,434	10,00	76,00	18,0
269	-2,217	15,380	644,730	11,0	73,98	18,00
269	-2,217	16,380	455,487		69,48	18,00
269	-2,217	17,380	232,600		59,41	17,50
269	N/A	18,380	56,947		0,00	16,75
269	N/A	19,380	398,948		-	15,75
269	N/A	20,380	597,383		-	14,75
269	N/A	21,380	752,032		-	13,75
269	N/A	22,380	398,948		-	13,00
270	N/A	-0,614	597,383		-	12,75
270	N/A	0,386	752,032		-	
270	N/A	1,386	398,948		-	-
270	N/A	2,386	597,383		-	
270	N/A	3,386	752,032		-	9,00
270	-2,618	4,386	852,356		0,00	-

Рисунок 2.11 – Фрагмент вихідних даних

Вбачаються такі проблеми:

- Є пропуски, причому пропущені значення можуть бути замінені різними символами (прочерк, знак питання), можуть бути просто відсутні значення або N/A;
- Є рядки з точкою як десятковим роздільником, а є з комою.

Як уже згадувалося раніше, вручну виправляти наявний вихідний файл буде нелегко. Щоб спростити даний етап, можна використовувати MATLAB. У ньому можна автоматизувати процес обробки файлу та внести коригування.

При вивченні файлу видно, що частина роботи виконана автоматично: частина стовпців із числовими даними прочитана одразу в потрібному числовому форматі, і навіть різні символи замість пропущених значень переведені в єдиний формат NaN. З метою спрощення роботи з файлом можна прибрати заголовки та крайній правий стовпець, оскільки в розглянутій вибірці у всіх записів одне й те саме значення. Це можна виконати за допомогою такого рядка коду:

```
[num,txt] = xlsread('solar_raw', 'A2:J180');
```

Для подальшої роботи необхідно перетворити стовпці з числовими даними в текстовому форматі в числовий формат. Найпростіше написати цикл, який йтиме рядок за рядком і перетворюватиме потрібні стовпці з матриці txt в числовий формат та додаватиме їх до матриці num. Даний цикл можна відобразити у вигляді такого коду:

```

for i = 1:length(txt)

    tmp = strsplit(string(txt(i, 9)), ';');
    tmp = double(strrep(tmp(1), ',', '.'));
    num(i, count_num_columns + 1) = tmp;

    tmp = strsplit(string(txt(i, 10)), ';');
    tmp = double(tmp(1));
    num(i, count_num_columns + 2) = tmp / 3.6;

    dt = datetime( ...
        string(txt(i, 1)) + " " + string(txt(i, 2)), ...
        'InputFormat', 'dd.MM.yyyy HH:mm:ss');

    num(i, count_num_columns + 3) = day(dt);
    num(i, count_num_columns + 4) = month(dt);
    num(i, count_num_columns + 5) = hour(dt);
    num(i, count_num_columns + 6) = minute(dt);

end

```

У середині циклу є дві частини, розділені порожнім рядком. У першій виконується переведення зі значення вигляду «10» у число (10). Для цього робиться таке:

- береться значення з *i*-го рядка та 9 стовпця матриці *txt* (*txt(i,9)*);
- указується, що його значення потрібно обробляти як рядок, тобто текстове значення (*string(txt(i,9))*);
- це значення потрібно розбити за пробілом і записати результат у тимчасову змінну (*tmp=strsplit(string(txt(i,9)), ' ');*);
- потрібно взяти першу частину (ліворуч від пробілу) та замінити кому на точку (*strrep(tmp(1), ',', '.')*);
- перевести отриманий поки ще текст у число (*tmp=double(strrep(tmp(1), ',', '.'))*);
- записати це число в *i*-й рядок нового стовпця матриці *num*.

Аналогічні дії виконуються в другій частині, тільки тут не потрібно замінювати кому на точку.

Для виявлення та видалення викидів можна використовувати стандартні функції MATLAB. Знайти середнє значення по кожному стовпцю, потім для кожного елемента матриці знайти модуль відхилення від середнього та

					КНУ КРБ.151.26.03.02.ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

порахувати медіани модулів відхилень. Якщо відхилення елемента дуже сильно відрізняється від медіанного відхилення, воно може бути викидом і знадобиться видалення. Код, що відповідає за це, представлений нижче:

```
cleared_data = rmmissing(num);

% Середні значення по стовпцях
mean_values = mean(cleared_data);

% Відхилення за модулем
absolute_deviations = abs(cleared_data - mean_values);

% Обчислюємо медіани відхилень
% (медіани майже нечутливі до викидів)
mean_abs_devs = median(absolute_deviations);

% Якщо якесь значення відхиляється дуже сильно
% (на два порядки), запам'ятовуємо цей рядок
outliers = absolute_deviations > 100 * mean_abs_devs;

% Знаходимо, у яких рядках є викиди
outlier_row_ids = sum(outliers, 2);

% Встановлюємо NaN замість викидів
% (неважливо в який стовпець, позначаємо весь рядок)
for i = 1:length(outlier_row_ids)

    if outlier_row_ids(i) > 0
        cleared_data(i,1) = NaN;
    end
end
```

На рисунку 2.12 представлено фрагмент файлу вже з очищеними даними, більше немає пропусків і весь файл приведений до загального формату запису даних.

Прогнозована модель не може зрозуміти циклічності безпосередньо, вона не знає, що години йдуть як 23:00 - 00:00. Тому потрібно зробити переведення з годин у кут від 0 до 2π радіан (повне коло). Додавання $\sin_{год}$ та $\cos_{год}$ дозволить представити час як точку на колі, а не як лінійну шкалу.

Формули для переведення синус години:

$$\sin_{год} = \sin\left(2\pi * \frac{год}{24}\right) \quad (2.14)$$

					КНУ КРБ.151.26.03.02.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Datetime	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number
Дата й час	Номер доби	Кут сонячн..	Сонячний ..	ППСВ на Гр..	Хмарність	Коефіцієнт	Температура	Вологість	Швидкість	Генерація ..
26- вер -20...	269	-2.2169	14.3799	787.4338	0	75.9963	18	30	6.9444	10206
26- вер -20...	269	-2.2169	15.3799	644.7302	0	73.9827	18	30	6.9444	8.1438e+03
26- вер -20...	269	-2.2169	16.3799	455.4867	0	69.4872	18	30	6.9444	5.2382e+03
26- вер -20...	269	-2.2169	17.3799	232.6001	0	59.4142	17.5000	31	6.5278	1.9841e+03
26- вер -20...	269	-2.2169	18.3799	56.9466	0	0	16.7500	32.5000	5.5556	141.9600
26- вер -20...	269	-2.2169	19.3799	0	0	0	15.7500	34.5000	3.8889	0
26- вер -20...	269	-2.2169	20.3799	0	0	0	14.7500	36.5000	4.4444	0
26- вер -20...	269	-2.2169	21.3799	0	0	0	13.7500	39.5000	6.3194	0
26- вер -20...	269	-2.2169	22.3799	0	0	0	13	44	6.4583	0
26- вер -20...	270	-2.6184	-0.6141	0	0	0	12.7500	44.7500	5	0
26- вер -20...	270	-2.6184	0.3859	0	0	0	11.7500	47	5.2778	0
26- вер -20...	270	-2.6184	1.3859	0	0	0	10.7500	47.7500	6.1111	0
27- вер -20...	270	-2.6184	2.3859	0	0	0	9.7500	51	6.1111	0
27- вер -20...	270	-2.6184	3.3859	0	0	0	9	55	5.8333	0
27- вер -20...	270	-2.6184	4.3859	0	0	0	8.7500	59	5.2778	0
27- вер -20...	270	-2.6184	5.3859	0	0	0	8	62	5.8333	0
27- вер -20...	270	-2.6184	6.3859	24.8370	0	5.6633	7.7500	63	4.7222	35.2800
27- вер -20...	270	-2.6184	7.3859	170.2501	0	58.8984	7	66	4.1667	1.4406e+03
27- вер -20...	270	-2.6184	8.3859	398.9479	0	68.1969	7.2500	65	5	4.6276e+03
27- вер -20...	270	-2.6184	9.3859	597.3828	0	73.9618	8.5000	60	5	7.7868e+03
27- вер -20...	270	-2.6184	10.3859	752.0317	0	75.2533	10.5000	52.2500	5.2778	9.9380e+03
27- вер -20...	270	-2.6184	11.3859	852.3556	12.5000	76.2772	12.2500	46.2500	6.1111	1.0985e+04
27- вер -20...	270	-2.6184	12.3859	891.5174	50	73.3174	13	43.2500	6.1111	1.0599e+04
27- вер -20...	270	-2.6184	13.3859	866.8485	50	65.1533	13.5000	39.2500	6.1111	9.0888e+03

Рисунок 2.12 – Фрагмент очищених даних

Переведення косинус години:

$$\cos_{\text{год}} = \sin\left(2\pi * \frac{\text{год}}{24}\right) \quad (2.15)$$

2.7 Побудова моделей машинного навчання в середовищі MATLAB

Щоб наблизити поставлене завдання до реального, в MATLAB і Python перші 80% даних використовуватимуться для навчання моделі, а решта 20% – для тесту [24]. Після очищення даних були побудовані регресійні моделі, які використані для аналізу, результати деяких представлені нижче.

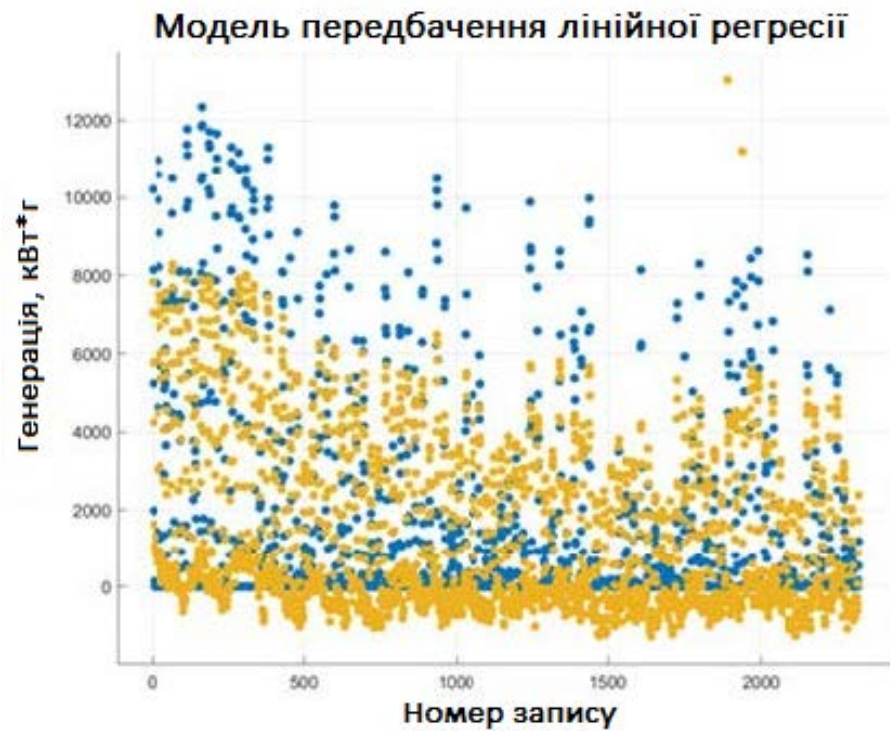


Рисунок 2.13 – Зіставлення прогнозу моделі лінійної регресії з фактичним виробітком



Рисунок 2.14 – Зіставлення прогнозу моделі випадкового лісу з фактичним виробітком



Рисунок 2.15 – Зіставлення прогнозу моделі випадкового лісу за допомогою дод. параметрів з фактичним виробітком

Випадковий ліс з використанням розширених налаштувань:

Minimum Leaf size – мінімальна кількість зразків у листку. Цей параметр регулює складність дерев в ансамблі. Значення до 5 зазвичай використовується для задач класифікації, 5-10 і більше – для задач регресії. При занадто малому значенні дерева стають глибшими та складнішими. Підвищується здатність моделі вловлювати тонкі закономірності, але зростає ризик перенавчання, особливо при шумних даних. При великих значеннях дерева спрощуються і підвищується стійкість до шуму.

Number of Learners – кількість дерев, що використовуються в ансамблі. Мала кількість дає низьку обчислювальну складність, високу дисперсію передбачень. Велика кількість призводить до зниження дисперсії за рахунок усереднення передбачень та підвищує стабільність моделі. Для задач із регресією рекомендують використовувати діапазон від 50-100.

На основі отриманих даних можна побудувати таблицю, яка відображатиме кількість помилок на тестовій вибірці.

					<i>КНУ КРБ.151.26.03.02.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

Таблиця 2.3 – Результати регресій MATLAB

Назва методу	RMSE	MSE	MAE	Час на навчання
Лінійна регресія	1375	1.38	812	1.9417 сек.
Випадковий ліс	423.87	1.79	173	1.6914 сек.
Випадковий ліс з розширеними налаштуваннями	388.65	1.5	149.28	2.54 сек.

Виходячи з отриманих даних, можна помітити, що метод випадкового лісу з підкоригованими показниками Minimum Leaf size та Number of Learners видав найменший середній модуль помилки (MAE), квадрат помилки (MSE), а також корінь із середнього квадрата помилки (RMSE).

Таким чином, випадковий ліс з оптимізованими параметрами демонструє високий потенціал для вирішення задач регресії, забезпечуючи стабільність, точність та стійкість до перенавчання.

2.8 Навчання та налаштування моделей з використанням Python

Перед початком навчання моделі необхідно перевірити отримані дані ще раз, але вже в середовищі розробки PyCharm. Для цього слід перетворити всі дані в тип float, розбити стовпець дата і час на різні стовпці та замінити роздільник «.» на «.». В оригіналі файлу є порожній рядок між даними, необхідно видалити і його, щоб уникнути помилок. Усе це можна виконати за допомогою бібліотеки Pandas.

Після всіх перетворень отримуємо дані, представлені на рисунку 2.16.

```
import pandas as pd

# Завантаження Excel-файлу
df = pd.read_excel('Random_Forest.xlsx', header=0)

# Видалення перших двох стовпців
df = df.iloc[:, 2:]

# Видалення порожніх рядків
df = df.dropna(how='all').reset_index(drop=True)

# Перетворення даних:
# заміна ком на крапки та конвертація в числовий формат
df = df.apply(
    lambda col: col.astype(str)
                    .str.replace(',', '.', regex=False)
                    .astype(float)
)
```

```
0      2017-09-26  12:00:00  ...      6.944444      10206.00
1      2017-09-26  13:00:00  ...      6.944444      8143.80
2      2017-09-26  14:00:00  ...      6.944444      5238.24
3      2017-09-26  15:00:00  ...      6.527778      1984.08
4      2017-09-26  16:00:00  ...      5.555556       141.96
...      ...      ...      ...      ...      ...
2321   2018-01-01  05:00:00  ...      1.944444        0.00
2322   2018-01-01  06:00:00  ...      3.055556       10.08
2323   2018-01-01  07:00:00  ...      3.055556       168.84
2324   2018-01-01  08:00:00  ...      3.472222       544.32
2325   2018-01-01  09:00:00  ...      3.055556      1193.64

[2326 rows x 12 columns]
```

Стовпці: Дата, Час, Швидкість вітру, м/с, Генерація, кВт·год

Рисунок 2.16 – Фрагмент даних після оптимізації

За допомогою бібліотеки matplotlib можна створити візуалізацію отриманих статистичних параметрів, таких як хмарність, коефіцієнт прозорості, вологість та швидкість вітру, представивши все це у вигляді гістограм. Код для побудови цих даних представлений нижче.

```
import matplotlib.pyplot as plt

columns_of_interest = [
    'Хмарність, %',
    'Коефіцієнт прозорості, %',
    'Вологість, %',
    'Швидкість вітру, м/с'
]

# Перевірка наявності стовпців у DataFrame
valid_columns = [
    col for col in columns_of_interest
    if col in df.columns
]

# Побудова гістограм
plt.figure(figsize=(12, 8))
df[valid_columns].hist(
    bins=20,
    color='skyblue',
    edgecolor='black',
    alpha=0.7
)

plt.suptitle('Гістограми параметрів', fontsize=16)
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```

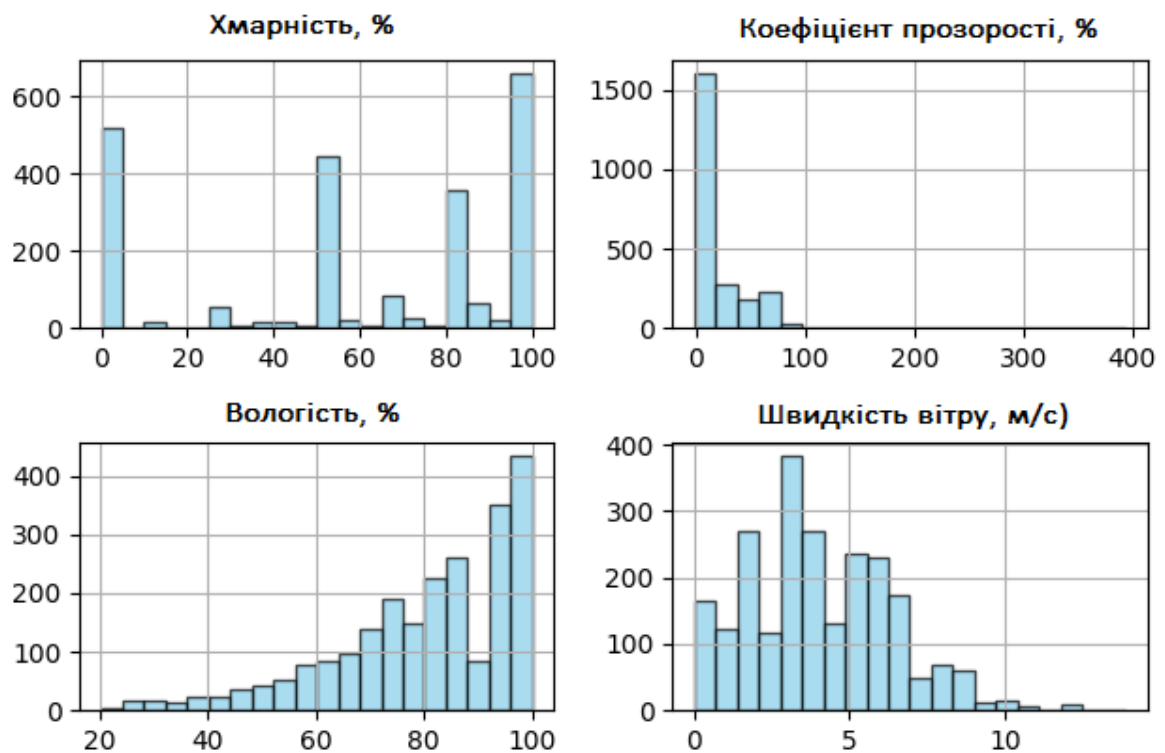


Рисунок 2.17 – Гістограми розподілу ознак

За допомогою Python було реалізовано лінійну регресію, дерево рішень, випадковий ліс, випадковий ліс із застосуванням оптимізації гіперпараметрів, Gradient boosting та XGBoost. Код реалізації завдання буде представлено в Додатку А.

2.8.1 Метод лінійної регресії

Через свою простоту модель погано навчилася, це можна помітити, якщо подивитися не лише на графіки, а й на показники середньоквадратичних помилок, результати яких представлені нижче.

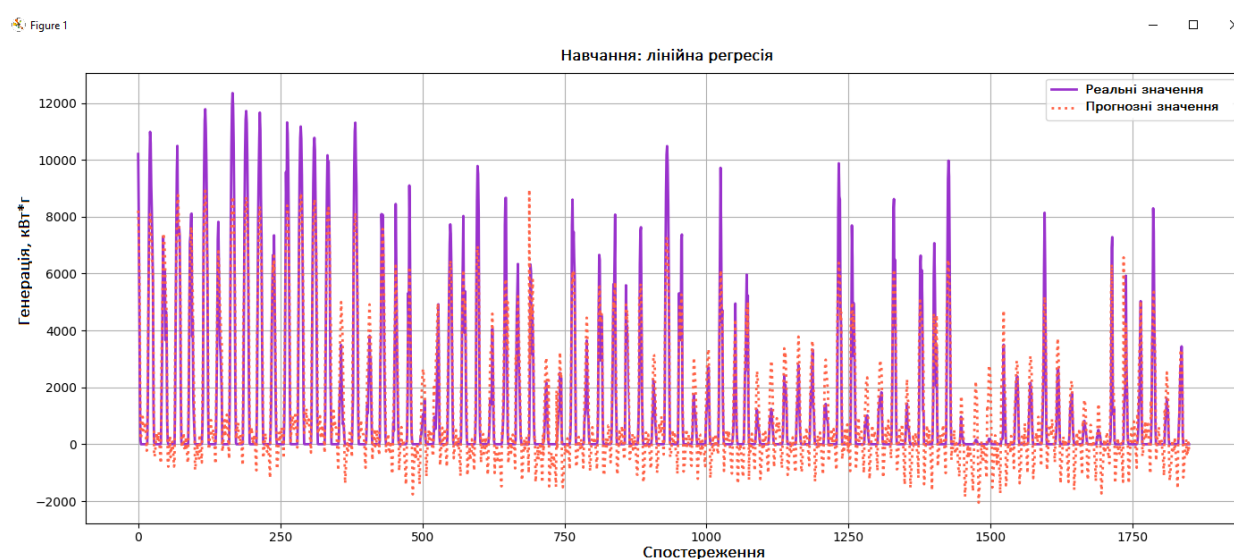


Рисунок 2.18 – Зіставлення прогнозу моделі лінійної регресії з фактичним виробітком (Навчання)

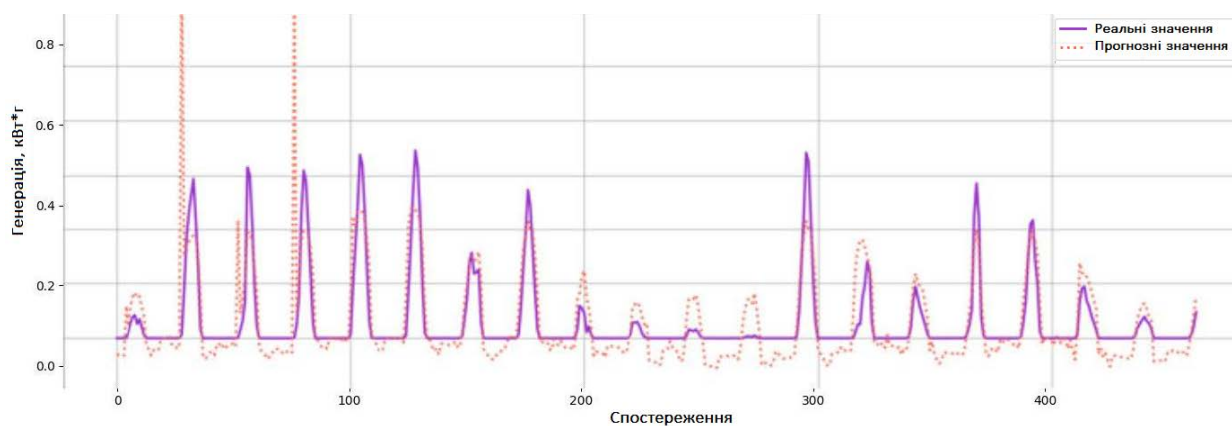


Рисунок 2.19 – Зіставлення прогнозу моделі лінійної регресії на тестовій вибірці

2.8.2 Дерево рішень

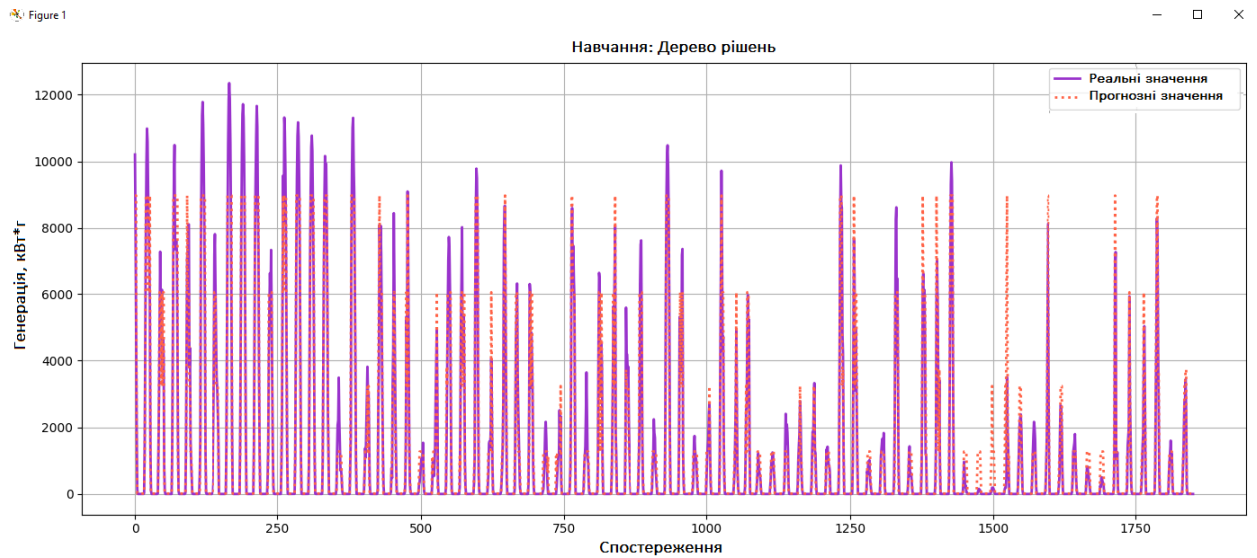


Рисунок 2.20 – Зіставлення прогнозу моделі дерева рішень з фактичним виробітком (Навчання)

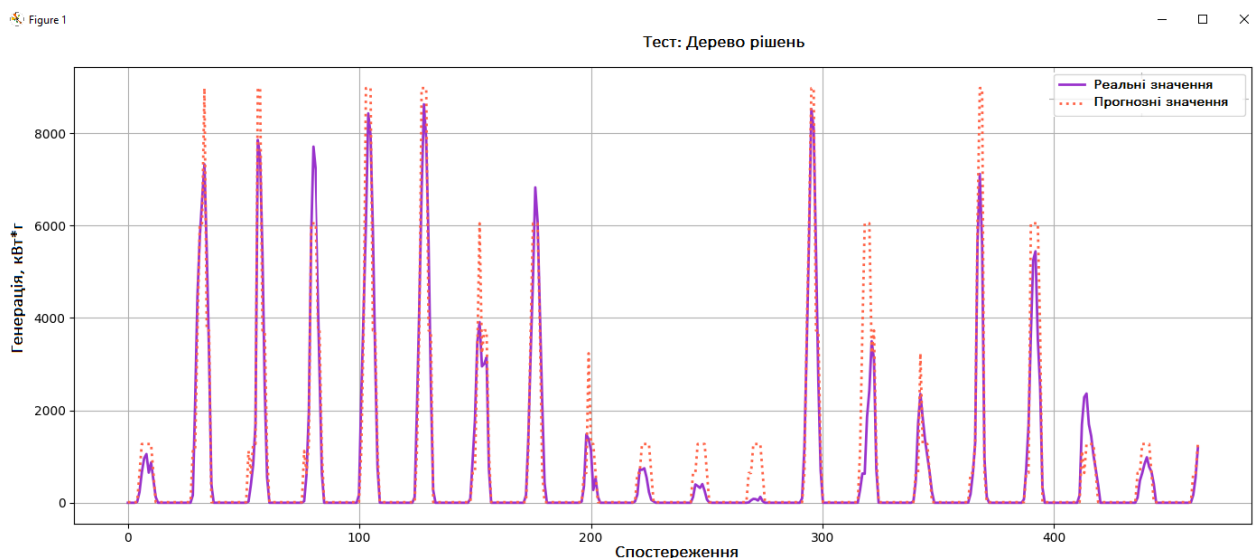


Рисунок 2.21 – Зіставлення прогнозу моделі дерева рішень на тестовій вибірці

2.8.3 Випадковий ліс

Точність результатів збільшилася, показник помилки порівняно з лінійною регресією зменшився майже в два рази.

					КНУ КРБ.151.26.03.02.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

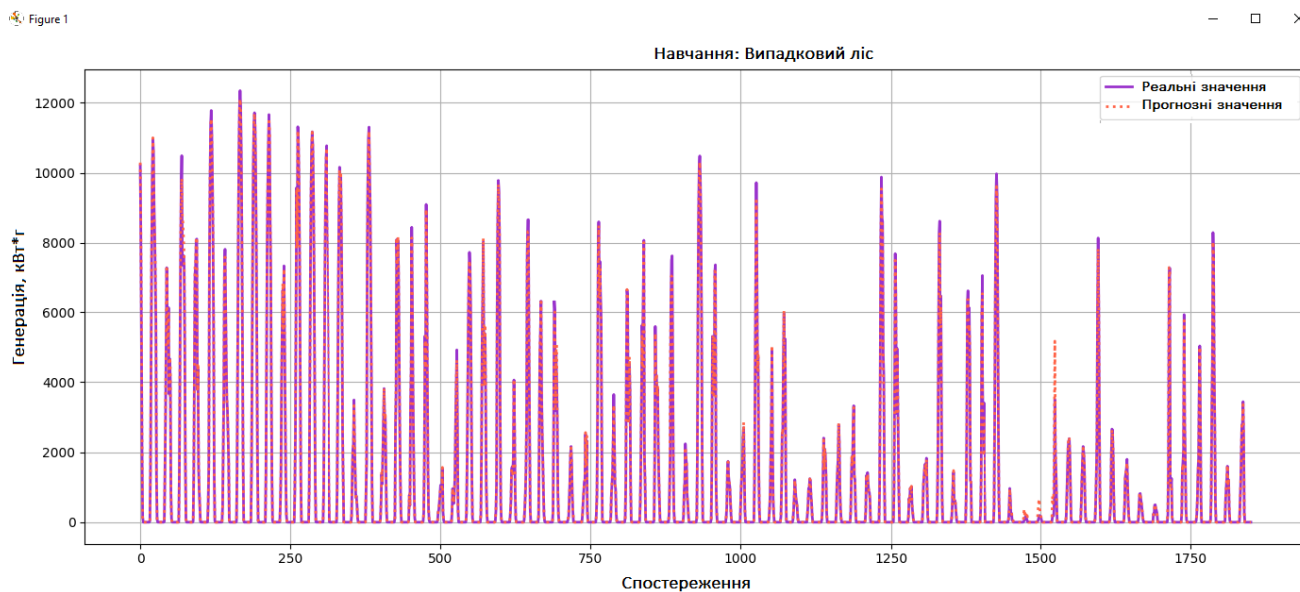


Рисунок 2.22 – Зіставлення прогнозу моделі випадкового лісу з фактичним виробітком

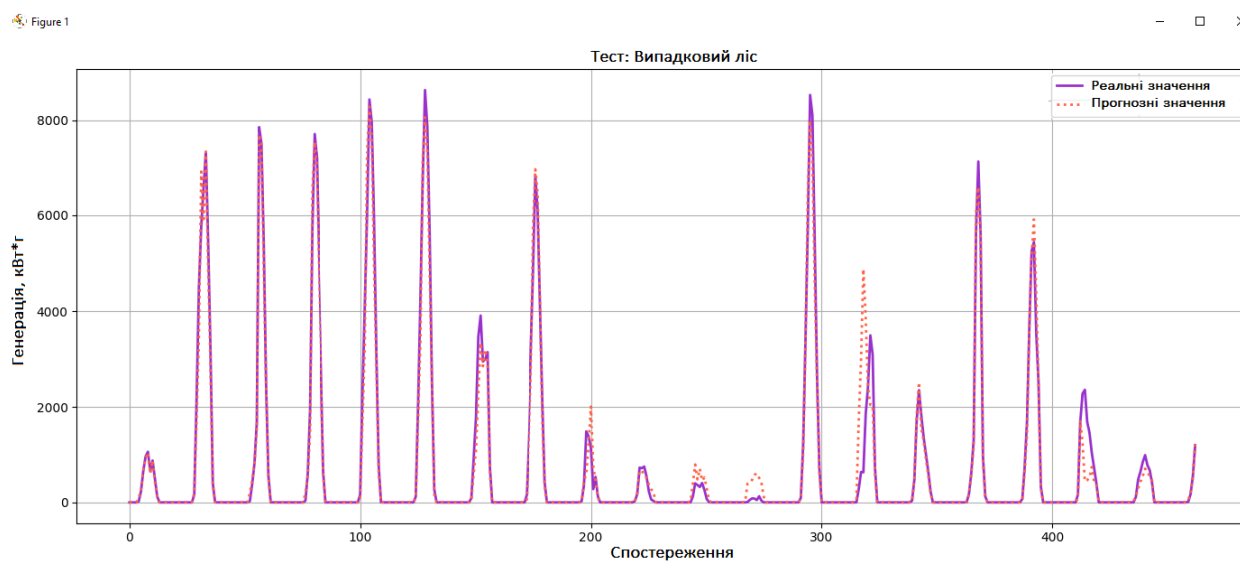


Рисунок 2.23 – Зіставлення прогнозу моделі випадкового лісу на тестовій вибірці

2.8.4 Gradient Boosting

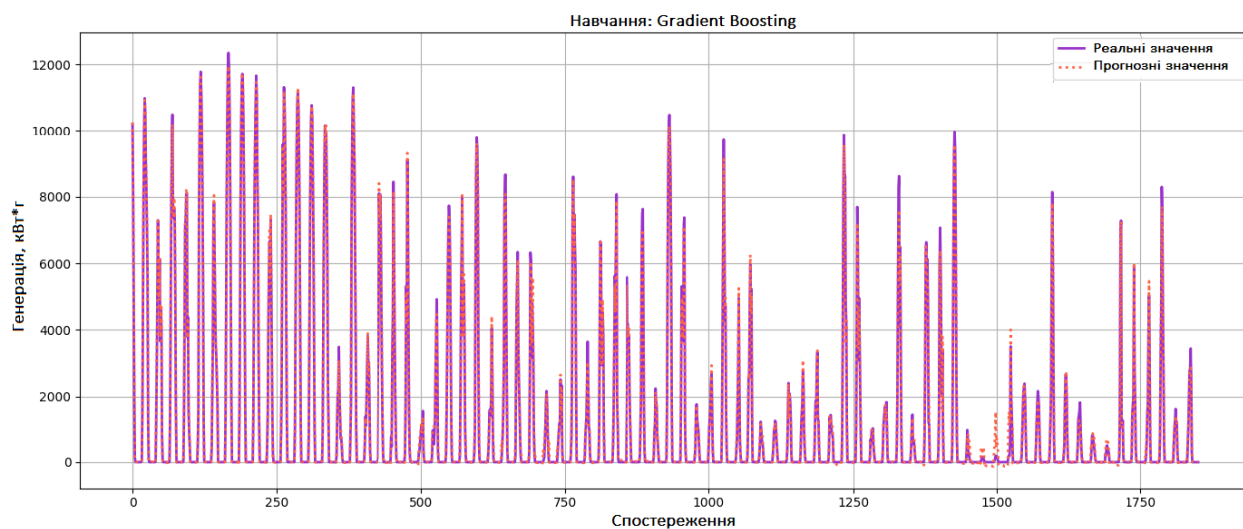


Рисунок 2.24 – Зіставлення прогнозу моделі Gradient Boosting з фактичним виробітком

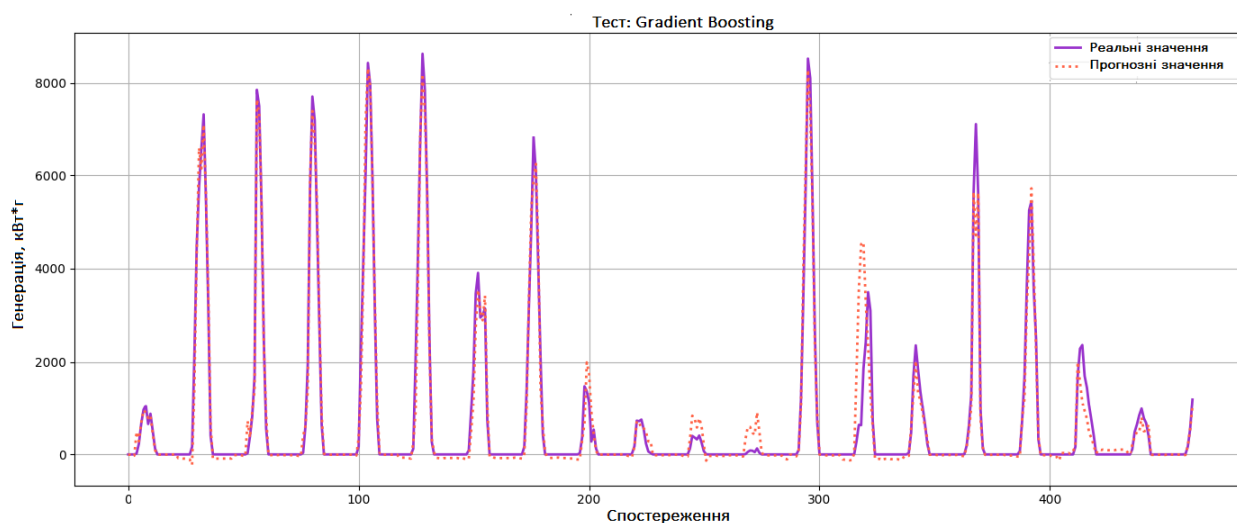


Рисунок 2.25 – Зіставлення прогнозу моделі Gradient Boosting на тестовій вибірці

2.8.5 Випадковий ліс за допомогою GridSearch

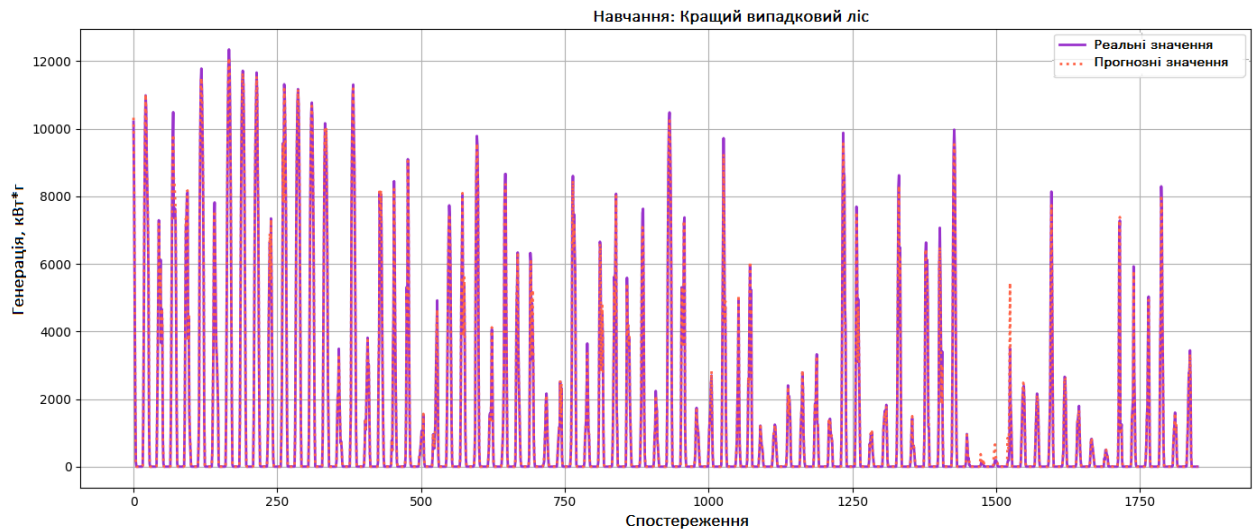


Рисунок 2.26 – Зіставлення прогнозу моделі випадкового лісу та GridSearch з фактичним виробітком

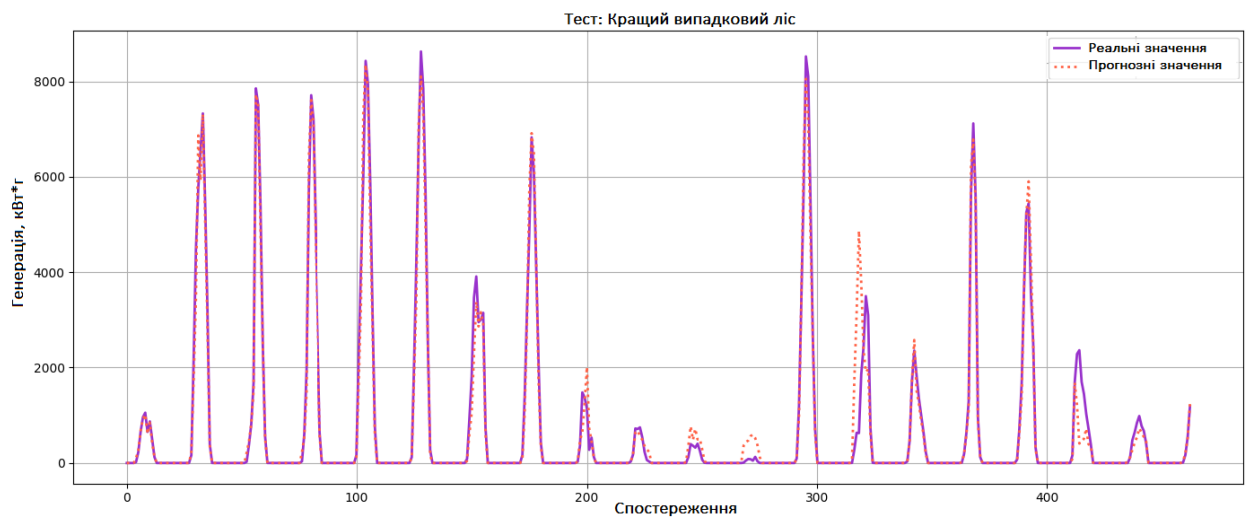


Рисунок 2.27– Зіставлення прогнозу моделі випадкового лісу та GridSearch на тестовій вибірці

2.8.6 XGBoost та GridSearch:

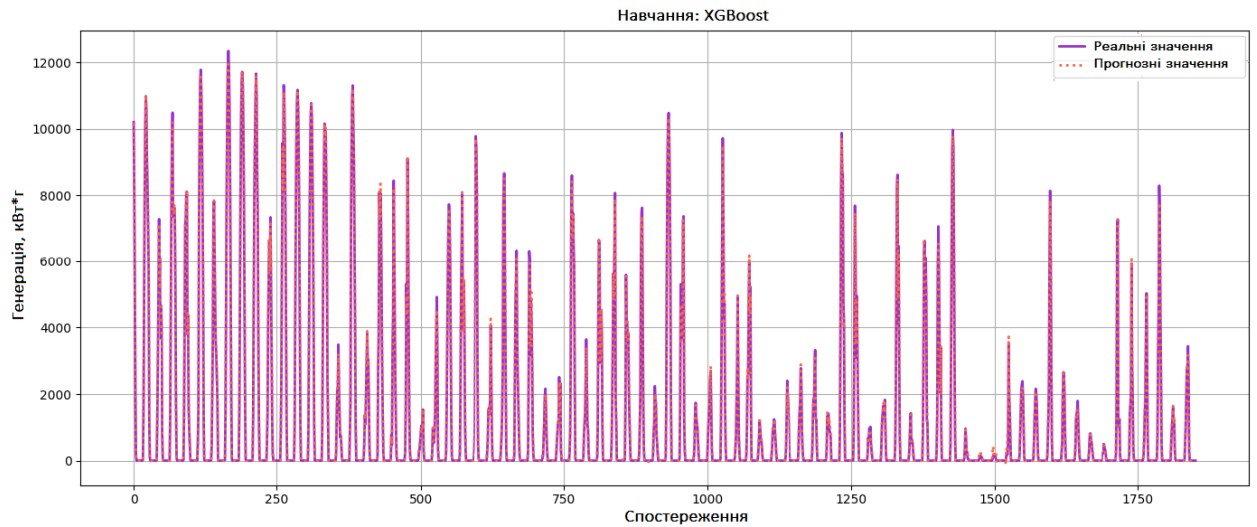


Рисунок 2.28– Зіставлення прогнозу моделі XGBoost та GridSearch з фактичним виробітком

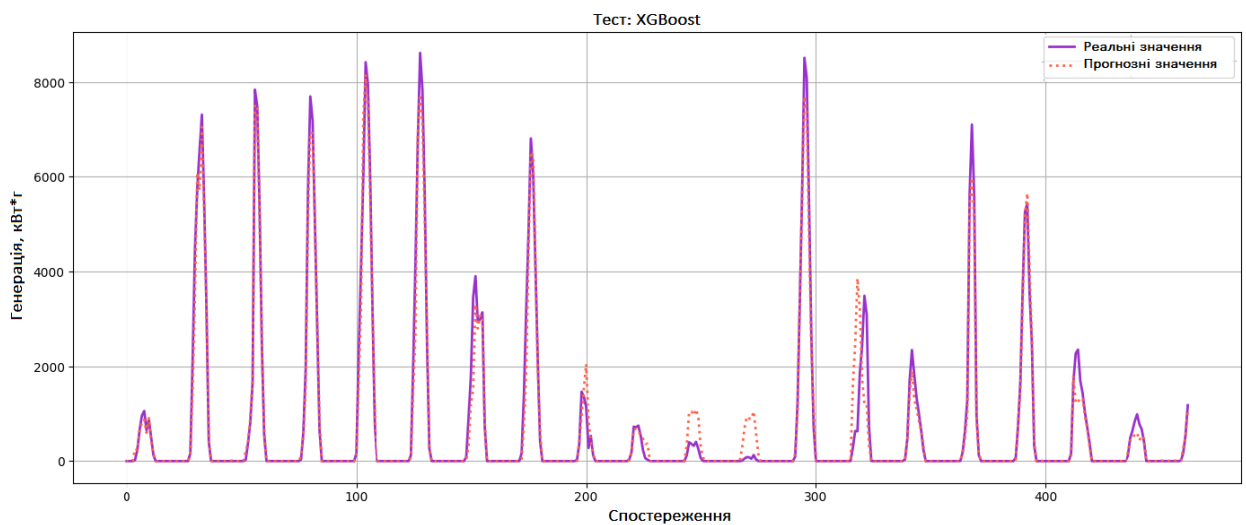


Рисунок 2.29 – Зіставлення прогнозу моделі XGBoost та GridSearch на тестовій вибірці

2.9 Оцінка якості моделей та висновки

Для оцінки якості перелічених моделей можна скласти таблицю всіх вищезгаданих методів для наочного розуміння різниці та подальшої роботи.

					КНУ КРБ.151.26.03.02.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Таблиця 2.4 – Результат моделей машинного навчання в Python

Назва моделі	RMSE помилка під час навчання	RMSE помилка під час тесту
Лінійна регресія	1123 кВт·год	1084 кВт·год
Дерево рішень	677 кВт·год	666 кВт·год
Випадковий ліс	148 кВт·год	371 кВт·год
Gradient boosting	190 кВт·год	387 кВт·год
Випадковий ліс і GridSearch	147 кВт·год	368 кВт·год
XGBoost і GridSearch	99 кВт·год	362 кВт·год

Тепер можна порівняти показники помилок у MATLAB та Python. Результати прогнозування вироблення електроенергії в MATLAB та Python показали близькі значення за точністю прогнозу. У Python додатково реалізовані складніші ансамблеві методи: градієнтний бустинг, XGBoost із застосуванням GridSearch.

Подібність отриманих прогнозів у різних програмних середовищах і при використанні різних підходів до навчання свідчить про коректність підготовки даних, стабільність обраних моделей та правильність напрямку дослідження.

Виходячи з показників, найкращим результатом можна вважати метод випадкового лісу та XGBoost, до них були застосовані додаткові налаштування моделі (GridSearch). Середньоквадратична помилка (RMSE) на тесті становила 362 та 368 кВт·год відповідно, від середнього значення виробітку 6347 кВт·год, що становить 6% від установленної потужності розглянутої електричної станції.

Таблиця 2.5 – Порівняння показників точності моделей машинного навчання, реалізованих на MATLAB та Python

Модель	MATLAB			Python		
Модель	RMSE, кВт·год	MAE, кВт·год	MSE, (кВт·год) ²	RMSE, кВт·год	MAE, кВт·год	MSE, (кВт·год) ²
Лінійна регресія	1375	812	1.89×10^6	1084	743.72	1.171×10^6

Випадковий ліс	423	173	0.17×10^6	371	127.43	0.137×10^6
Випадковий ліс з дод. параметрами	388	149	0.15×10^6	368	124.01	0.135×10^6

Висновки за розділом:

У другому розділі представлені фізико-математичні основи перетворення сонячної енергії, а також опис методів машинного навчання, що застосовуються для прогнозування виробітку електроенергії. Розглянуто етапи підготовки даних, описано метрики якості моделей, які забезпечують об'єктивну оцінку точності прогнозу. Наведено основні моделі машинного навчання, застосування яких дозволяє побудувати точні моделі прогнозування генерації енергії.

У ході реалізації прогновної моделі для оцінки вироблення електроенергії було виконано етапи підготовки та аналізу даних із застосуванням машинного навчання. Дані пройшли етап передобробки: усунуто пропуски, видалено викиди, стандартизовано формати числових значень та виконано обробку циклічних ознак, таких як час доби.

В MATLAB було реалізовано регресійні моделі, включаючи лінійну регресію та випадковий ліс із ручним налаштуванням гіперпараметрів. В Python з використанням бібліотек Pandas, Scikit-learn та Matplotlib проведено подальше навчання та оптимізацію моделей, включаючи використання ансамблевих методів. Найкращі значення показали моделі Випадкового лісу та XGBoost з налаштованими гіперпараметрами, демонструючи стабільність та високу точність прогнозу. Середньоквадратична помилка становила близько 360-370 кВт·год, що відповідає 6% від установленної потужності, це підтверджує їхню практичну застосовність

ВИСНОВКИ

У межах даної роботи було проведено комплексне дослідження, спрямоване на розробку та аналіз моделей короткострокового прогнозування вироблення електроенергії фотоелектричними станціями з використанням методів машинного навчання. Актуальність дослідження зумовлена стрімким зростанням частки ВДЕ в енергетичних системах, необхідністю підвищення надійності та ефективності управління «зеленою» генерацією, а також збільшенням обсягів даних, що потребують автоматизованої обробки.

У першому розділі роботи було розглянуто сучасний стан та тенденції розвитку сонячної енергетики, зокрема зростаючу роль фотоелектричних станцій у структурі енергосистем. Показано, що інтеграція СЕС у сучасні енергетичні мережі потребує врахування їхньої високої залежності від метеорологічних умов та природних факторів, що зумовлює необхідність використання сучасних методів аналізу даних і прогнозування. Обґрунтовано, що застосування технологій машинного навчання та методів обробки великих даних є ключовим інструментом підвищення ефективності управління «зеленою» генерацією. Це дозволяє зменшити залежність енергосистем від традиційних джерел генерації, забезпечити більш стабільний баланс між виробництвом і споживанням електроенергії та підвищити загальну енергетичну безпеку.

У другому розділі наведено фізико-математичні основи процесів перетворення сонячної енергії в електричну, що дозволило сформулювати теоретичне підґрунтя для подальшого моделювання. Детально розглянуто основні підходи машинного навчання, які застосовуються для задач короткострокового прогнозування генерації електроенергії, включаючи регресійні та ансамблеві методи.

					<i>КНУ КРБ.151.26.03.00.ПЗ</i>		
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>ВИСНОВКИі</i>		
<i>Розроб.</i>		<i>Галушка В. О.</i>					
<i>Перевір.</i>		<i>Жосан А. А.</i>					
<i>Н. Контр.</i>		<i>Маринич І.А.</i>					
<i>Затверд.</i>		<i>Маринич І.А.</i>			<i>АКНТ АКІТ-22</i>		
					<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
						58	4

Окрему увагу приділено етапам підготовки даних, що є критично важливими для якості моделей: очищення від пропусків і викидів, нормалізація числових значень, а також трансформація часових і циклічних ознак. Також описано метрики оцінки якості моделей, які дозволяють об'єктивно визначати точність та стабільність прогнозів. У сукупності це створює цілісну методологічну основу для побудови ефективних прогнозних моделей.

У процесі практичної реалізації прогнозної моделі для оцінки вироблення електроенергії було виконано повний цикл роботи з даними та моделями машинного навчання. Початковий етап включав ретельну передобробку даних, що передбачала усунення пропусків, фільтрацію аномальних значень, стандартизацію форматів числових показників та коректну обробку циклічних ознак, таких як година доби та сезонні коливання. Це дозволило підвищити якість вхідного набору даних та забезпечити стабільність подальшого навчання моделей.

Для реалізації та первинного тестування моделей було використано середовище MATLAB, де побудовано регресійні моделі, зокрема лінійну регресію та модель випадкового лісу з базовим налаштуванням параметрів. Подальший етап дослідження здійснювався в середовищі Python із використанням бібліотек Pandas, Scikit-learn та Matplotlib, що дозволило розширити набір інструментів аналізу, виконати більш глибоку оптимізацію моделей та провести порівняльне дослідження різних підходів машинного навчання.

Результати експериментів показали, що найвищу точність та стабільність демонструють ансамблеві методи, зокрема модель випадкового лісу та градієнтний бустинг XGBoost із підібраними гіперпараметрами. Дані моделі забезпечили найменші значення середньоквадратичної помилки на рівні приблизно 360–370 кВт·год, що становить близько 6% від встановленої потужності станції. Отримані результати підтверджують високу ефективність застосованих методів машинного навчання та їхню практичну придатність для задач короткострокового прогнозування генерації електроенергії.

					КНУ КРБ.151.26.03.00.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

Узагальнюючи, можна стверджувати, що проведене дослідження підтверджує доцільність використання інтелектуальних методів аналізу даних для підвищення точності прогнозування роботи фотоелектричних станцій. Розроблена методика може бути використана як основа для створення систем підтримки прийняття рішень в енергетиці, оптимізації режимів роботи СЕС та підвищення ефективності інтеграції відновлюваних джерел енергії в енергосистему України.

Подальший розвиток роботи може бути спрямований на використання глибинного навчання та моделей часових рядів для ще більш точного прогнозування в умовах змінної генерації.

					<i>КНУ КРБ.151.26.03.02.ПЗ</i>	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Sustainable Development Goals [United Nations] URL: <https://sdgs.un.org/goals/goal7> (дата звернення: 06.04.2026).
2. Wood D.A. German solar power generation data mining and prediction with transparent open box learning network integrating weather, environmental and market variables // *Energy Conversion and Management*. 2019. P. 354–369.
3. Hourly predictions of direct normal irradiation using an innovative hybrid LSTM model for concentrating solar power projects in hyper- arid regions / A. Djaafari [et al.] // *Energy Reports*. 2022. V. 8. P. 15548–15562.
4. Zufferey T., Lepouze A., Hug G. Inadequacy of standard algorithms and metrics for short-term load forecasts in low-voltage grids. *IEEE Milan PowerTech, Milan, Italy*. 2019. P. 1–6.
5. Ali J., Khan R., Ahmad N., Maqsood I. Random Forests and Decision Trees // *International Journal of Computer Science Issues (IJCSI)*. 2012. Vol. 9.
6. Biau G. Analysis of a Random Forests Model // *Journal of Machine Learning Research*. 2010. № 13.
7. Delgado M., Sirsat M., Cernadas E., Alawadi S., Barro S., Febrero-Bande M. An extensive experimental survey of regression methods // *Neural Networks*. 2019. Vol. 111. P. 11–34.
8. Gambella C., Ghaddar B., Naoum-Sawaya J. Optimization Models for Machine Learning: A Survey // *European Journal of Operational Research*. 2019. Vol. 290. P. 807–828.
9. Child M., Bogdanov D., Breyer C. The Role of Storage Technologies for the Transition to a 100% Renewable Energy System in Europe // *Energy Procedia*. 2018. Vol. 155. P. 44–60. doi: 10.1016/j.egypro.2018.11.067.

					КНУ КРБ.151.26.03.00.ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Галушка В. О.			СПИСОК ЛІТЕРАТУРИ			Літ.	Арк.	Аркушів	
Перевір.		Жосан А. А.								66	3
								АКНТ АКІТ-22			
Н. Контр.		Маринич І.А.									
Затвердив		Маринич І.А.									

10. Delgado M. An extensive experimental survey of regression methods // *Neural Networks*. 2019. Vol. 111. P. 11–34.
11. Gopi A., Sudhakar K., Keng N., Krishnan A., Priya S. Performance Modeling of the Weather Impact on a Utility-Scale PV Power Plant in a Tropical Region // *International Journal of Photoenergy*. 2021.
12. Atyia T., Qasim M. Evaluating the Impact of Weather Conditions on the Effectiveness and Performance of PV Solar Systems and Inverters // *NTU Journal of Renewable Energy*. 2023 doi: 10.56286/ntujre.v5i1.551.
13. Voskanyan A., Borden D., Yakimovich B., Barmani A. Studies of the Operation of a Solar Power Plant // *Applied Solar Energy*. 2019. Vol. 55. P. 235–241. doi: 10.3103/S0003701X19040091.
14. Cheng S., Xie Y., Shu Z., Wang Y., Ning Z., Yang Y. Effect of Different Solar Irradiance Parameters on Reliability Evaluation of the Grid // *IOP Conference Series: Earth and Environmental Science*. 2019. doi: 10.1088/1755-1315/300/4/042115.
15. Наземна сонячна електростанція Нікопольська СЕС. URL: https://www.energo.ua/uk/assets/nikopolska_pv_plant (дата звернення: 06.04.2026).
16. Абетка енергетики: Нікопольська СЕС. URL: <https://ua-energy.org/uk/posts/abetka-enerhetyky-nikopolska-ses>. (дата звернення: 06.04.2026).
17. Tuvshin O., Rusina A., Bayasgalan Z., Matrenin P.V. Machine Learning Application for Renewable Energy Forecasting. 2023. P. 1–2.
18. Jurj D.I., Micu D.D., Muresan A. Overview of Electrical Energy Forecasting Methods and Models in Renewable Energy // *2018 International Conference and Exposition on Electrical and Power Engineering (EPE)*. 2018. P. 87–90. doi: 10.1109/ICEPE.2018.8559807.
19. Cheung K.W. New forecasting and modeling tools for operating power systems with variable energy resources // *2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*. 2014. P. 186–191.

					КНУ КРБ.151.26.03.00.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

20. Negnevitsky M., Mandal P., Srivastava A.K. An overview of forecasting problems and techniques in power systems // *2009 IEEE Power & Energy Society General Meeting*. 2009. P. 1–4.

21. Aburiyana G., El-Hawary M.E. An overview of forecasting techniques for load, wind and solar powers // *2017 IEEE Electrical Power and Energy Conference (EPEC), Saskatoon, SK*. 2017. P. 1–7. doi: 10.1109/EPEC.2017.8286192.

22. Kuzlu M., Cali U., Sharma V., Güler O. Gaining Insight into Solar Photovoltaic Power Generation Forecasting Utilizing Explainable Artificial Intelligence Tools // *IEEE Access*. 2020. Vol. 8. P. 187814–187823.

23. CandianSolar_Low temperature_coefficient. URL: <https://www.csisolar.com/new-n-type-high-power-182-pro-topbihiku6/> (дата звернення: 06.04.2026).

24. Моркун Н. В., Маринич І. А. Методичні вказівки до виконання кваліфікаційної роботи бакалавру для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”. Кривий Ріг : Видавничий центр КНУ, 2019. 50 с.

25. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).

26. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).

27. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація)

28. ДСТУ 3651.0-97 Метрологія. Одиниці фізичних величин. Основні одиниці фізичних величин Міжнародної системи одиниць. Основні положення, назви та позначення Київ, Держстандарт України, 1998. 27 с. (Інформація та документація)

					КНУ КРБ.151.26.03.00.ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Фрагменти ісходного коду програми на мові Python

```
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, GridSearchCV, TimeSeriesSplit
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error
```

```
# =====
# Завантаження та підготовка даних
# =====

file_path = 'RRRandom_Forest.xlsx'
df = pd.read_excel(file_path)

df = df.drop(index=0).reset_index(drop=True)

df['Дата та час'] = pd.to_datetime(df['Дата и время'], format='%d.%m.%y %H:%M')
df['Дата'] = df['Дата та час'].dt.date
df['Час'] = df['Дата та час'].dt.time
df = df.drop('Дата та час', axis=1)

new_columns_order = [
    'Дата', 'Час',
    'Номер дня в році',
    'Кут сонячного схилення',
    'Сонячний час',
    'ППСІ на Гр Атм',
    'Хмарність, %',
    'Коефіцієнт прозорості, %',
    'Температура, °C',
    'Вологість, %',
    'Швидкість вітру, м/с',
    'Генерація, кВт·год'
]
```

```
df = df[new_columns_order]

numeric_cols = df.columns[2:-1]
for col in numeric_cols:
    df[col] = df[col].astype(str).str.replace(',', '.').str.extract(r'([-+]?\d*\.\d+)', expand=False)
    df[col] = pd.to_numeric(df[col], errors='coerce')

df = df.dropna(subset=numeric_cols)
```

```
# =====
# Фічі часу
# =====
df = df[df['Генерация, кВтч'].notna()]

df['datetime'] = pd.to_datetime(df['Дата'].astype(str) + ' ' + df['Время'].astype(str))

df['year'] = df['datetime'].dt.year
df['month'] = df['datetime'].dt.month
df['day'] = df['datetime'].dt.day
df['hour'] = df['datetime'].dt.hour
df['day_of_week'] = df['datetime'].dt.dayofweek
df['is_weekend'] = df['day_of_week'].isin([5, 6]).astype(int)
df['day_of_year'] = df['datetime'].dt.dayofyear

df['sin_hour'] = np.sin(2 * np.pi * df['hour'] / 24)
df['cos_hour'] = np.cos(2 * np.pi * df['hour'] / 24)

df = df.drop(['Дата', 'Время', 'datetime'], axis=1)

X = df.drop('Генерация, кВтч', axis=1)
y = df['Генерация, кВтч']
```

```
# =====
# Train/Test split (time-based)
# =====
split = int(len(df) * 0.8)

X_trainval = X.iloc[:split]
y_trainval = y.iloc[:split]

X_test = X.iloc[split:]
y_test = y.iloc[split:]

# =====
# Метрики
# =====
def calc_rmse(model, X, y):
    return np.sqrt(mean_squared_error(y, model.predict(X)))

def calc_mse(model, X, y):
    return mean_squared_error(y, model.predict(X))

def calc_mae(model, X, y):
    return mean_absolute_error(y, model.predict(X))
```

```

# =====
# Візуалізація
# =====
def visualize(model, X, y, title="Predictions vs Actual"):
    y_pred = model.predict(X)
    idx = np.arange(len(y))

    mse = calc_mse(model, X, y)
    mae = calc_mae(model, X, y)

    plt.figure(figsize=(14, 6))
    plt.plot(idx, y.values, label="Фактичні значення", linewidth=2)
    plt.plot(idx, y_pred, label="Прогноз", linestyle=':', linewidth=2)

    plt.title(f"{title}\nMSE: {mse:.2f}, MAE: {mae:.2f}")
    plt.xlabel("Індекс")
    plt.ylabel("Генерація")
    plt.legend()

    plt.grid(True)
    plt.tight_layout()
    plt.show()

```

```

# =====
# Моделі
# =====

print("\nЛінійна регресія")
model_lr = LinearRegression()
model_lr.fit(X_trainval, y_trainval)

visualize(model_lr, X_test, y_test, "Лінійна регресія (тест)")

print("\nДерево рішень")
model_tree = DecisionTreeRegressor(max_depth=3)
model_tree.fit(X_trainval, y_trainval)

visualize(model_tree, X_test, y_test, "Дерево рішень (тест)")

print("\nВипадковий ліс")
model_rf = RandomForestRegressor(n_estimators=100, random_state=42)
model_rf.fit(X_trainval, y_trainval)

visualize(model_rf, X_test, y_test, "Random Forest (тест)")

```

```
print("\nПідбір параметрів Random Forest (GridSearchCV)")
param_grid = {
    'n_estimators': [100, 150],
    'max_depth': [None, 5, 10],
    'min_samples_split': [2, 4]
}
```

```
tscv = TimeSeriesSplit(n_splits=10)

grid_search = GridSearchCV(
    estimator=RandomForestRegressor(random_state=42),
    param_grid=param_grid,
    scoring='neg_root_mean_squared_error',
    cv=tscv,
    verbose=1,
    n_jobs=-1
)

grid_search.fit(X_trainval, y_trainval)

best_rf = grid_search.best_estimator_
print("Найкращі параметри RF:", grid_search.best_params_)

visualize(best_rf, X_test, y_test, "Best Random Forest (тест)")
```

```
print("\nGradient Boosting")
model_gb = GradientBoostingRegressor(
    n_estimators=100,
    learning_rate=0.1,
    max_depth=3,
    random_state=42
)

model_gb.fit(X_trainval, y_trainval)

visualize(model_gb, X_test, y_test, "Gradient Boosting (тест)")

print("\nXGBoost + GridSearchCV")

param_grid_xgb = {
    'n_estimators': [150],
    'learning_rate': [0.01, 0.05],
    'max_depth': [5],
    'subsample': [0.8],
    'colsample_bytree': [0.6, 0.8]
}
```

```
tscv = TimeSeriesSplit(n_splits=10)

grid_search_xgb = GridSearchCV(
    estimator=XGBRegressor(random_state=42, tree_method='hist'),
    param_grid=param_grid_xgb,
    scoring='neg_root_mean_squared_error',
    cv=tscv,
    verbose=1,
    n_jobs=-1
)
```

```
grid_search_xgb.fit(X_trainval, y_trainval)

best_xgb = grid_search_xgb.best_estimator_
print("Найкращі параметри XGB:", grid_search_xgb.best_params_)

visualize(best_xgb, X_test, y_test, "XGBoost (тест)")

print("\nГотово ✓")
```

					КНУ КРБ.151.26.03.00.ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

