

Київський національний університет імені Тараса Шевченка
Факультет радіофізики, електроніки та комп'ютерних систем
Кафедра комп'ютерної інженерії

Випускна кваліфікаційна робота бакалавра
**«Прогнозування споживання електроенергії побутовими
приладами будинку»**

Виконав студент 4 курсу
спеціальності 123 «Комп'ютерна інженерія»
Євгеній ІБРАГІМОВ

Науковий керівник,
к. ф.-м. наук, асистент
Андрій КОНОВАЛОВ

Рецензент
доцент кафедри квантової радіофізики та
наноелектроніки д. ф.-м. наук
Андрій ГОРЯЧКО

До захисту допускаю:

Завідувач кафедри, к.ф.-м.н.,
доцент **Юрій БОЙКО**

Ухвалено на засіданні кафедри “ ___ ” _____ 2022 р., протокол № ___

Київ 2022

РЕФЕРАТ

Дипломна робота: 32 с., 6 рис., 3 табл., 1 додаток, 19 джерел.

Мета роботи - розробка моделі багатошарового персептрону для прогнозування споживання електроенергії побутовими приладами будинку на прикладі набору даних “Appliances energy prediction”.

В першому розділі визначено мету та завдання прогнозування споживання електроенергії; досліджено методи прогнозування споживання електроенергії; визначені інформаційні джерела, які забезпечують прогнозування споживання електроенергії.

У другому розділі досліджено методику прогнозування споживання електроенергії та розглянуті показники, які використовуються для оцінки прогнозуючих моделей.

У третьому розділі проведено аналіз роботи прогнозуючих моделей, їх показників у порівнянні з опрацьованою літературою.

Встановлено, що використання моделі багатошарового персептрону дозволяє точніше прогнозувати споживання електроенергії побутовими приладами будинку за меншої кількості ознак.

ЛІЧИЛЬНИК ЕЛЕКТРОЕНЕРГІЇ, НЕЙРОННА МЕРЕЖА,
АЛГОРИТМ, ПРЕДИКТОР

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	4
ВСТУП	5
РОЗДІЛ 1. ОГЛЯД ЛІТЕРАТУРИ	6
1.1 Побутові прилади будинку	6
1.2 Збір інформації	6
1.3 Методи та процес збору інформації	8
1.4 Аналіз та фільтрація зібраних даних	10
1.5 Прогнозування та нейронні мережі	11
1.6 Оцінка раніше використаних моделей	11
1.7 Багатошаровий перцептрон	13
1.8 Постановка задач роботи	14
РОЗДІЛ 2. МЕТОДИКА ДОСЛІДЖЕНЬ	16
2.1 Збір даних	16
2.2 Інструментарій досліджень	18
2.3 Визначення параметрів	20
2.4 Метрики якості моделей	22
2.5 Відбір ознак	22
РОЗДІЛ 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ	24
3.1 Дослідження за допомогою багатошарового перцептрону	24
3.2 Аналіз отриманих результатів	26
ВИСНОВКИ	28
ПЕРЕЛІК ПОСИЛАНЬ	29
ДОДАТОК	31

ПЕРЕЛІК СКОРОЧЕНЬ

RFE - рекурсивне видалення ознак

CARET - навчальний пакет з класифікації та регресії

LM – лінійна модель

SVM - radial – машина опорних векторів

RF – випадковий ліс

GBM – машина градієнтного бустингу

ВСТУП

Наразі одним з найбільш поширених напрямів є використання моделей і алгоритмів машинного навчання, що здатне вирішувати задачі багатьох галузей і використовується в майже будь яких сферах. Виключенням не стала й область дослідження енергоефективності та електроспоживання. Саме тут є можливість дослідити витрати, спричинені різними видами побутових приборів. Споживання ними енергії будинку, що знаходяться в ньому і становлять 20 - 30 % від загальних витрат, є темою багатьох досліджень. Це надає можливість визначення необхідної кількості та якості енергоносіїв та фотоелектричних елементів при їх виготовленні та застосуванні, проектувати енергоменеджмент будинку та житлових споруд та розробляти паттерн оптимального використання побутових приладів.

Метою роботи є розробка моделі багат шарового перцептронну для прогнозування споживання електроенергії побутовими приладами будинку на прикладі набору даних “Appliances energy prediction”.

Практичною значимістю роботи є можливість використання моделі машинного навчання для прогнозування споживання електроенергії з метою контролю навантаження енергомережі та виявлення паттерну ненормального використання енергії.

РОЗДІЛ 1. ОГЛЯД ЛІТЕРАТУРИ

1.1 Побутові прилади будинку

Побутові прилади сягають значної частини (20 - 30 % [1,2]) від усього споживання електроенергії, що можна пояснити двома основними чинниками [3]: тип та кількість електроприборів та їх використання мешканцями. Під час своєї роботи, побутові прибори здатні залишати певні сліди в місцях свого використання як от, наприклад, зміни температури та вологості, світло, шум. Це надає можливість вирішувати проблему прогнозування споживання електроенергії побутовими приладами будинку в сукупності, за допомогою чого можна вирішувати ряд проблем, які виникають через надмірність витрат електроенергії. До цього можна віднести як проектування та виготовлення приладів та пристроїв, зменшення витоку енергії в мережі та вирішення ряду екологічних питань.

1.2 Збір інформації

В роботі [4] проводяться дослідження в сфері зборі даних, їх аналізу та прогнозування витрат електроенергії побутовими приладами. Даними, які були забрані для виконання роботи, є пункти, наведені в таблиці 1.1. Інформація в будинку була зібрана за допомогою бездротових датчиків ZigBee, а погода - з найближчої метеостанції в 12 кілометрах від будинку. Саме ці дані були взяті на основі досліджень [4], їх актуальність була більш детально описана в роботах [5,6].

Таблиця 1.1. Набір даних “Appliances energy prediction”

Номер	Назва змінної	Одиниці виміру
1	Споживання енергії	Ват на годину
2	Освітлення	Ват на годину

3	T1, Температура в кухонній зоні	градуси Цельсія
4	RH_1, Вологість в кухонній зоні	%
5	T2, Температура в зоні вітальні	градуси Цельсія
6	RH_2, Вологість в зоні вітальні	%
7	T3, Температура в зоні пральні	градуси Цельсія
8	RH_3, Вологість в зоні пральні	%
9	T4, Температура в офісному приміщенні	градуси Цельсія
10	RH_4, Вологість в офісному приміщенні	%
11	T5, Температура в ванній кімнаті	градуси Цельсія
12	RH_5, Вологість в ванній кімнаті	%
13	T6, Температура в поза будівлею	градуси Цельсія
14	RH_6, Вологість в поза будівлею	%
15	T7, Температура в приміщенні для прасування	градуси Цельсія
16	RH_7, Вологість в приміщенні для прасування	%
17	T8, Температура в дитячій зоні	градуси Цельсія
18	RH_8, Вологість в дитячій зоні	%

19	T9, Температура в кімнаті батьків	градуси Цельсія
20	RH_9, Вологість в кімнаті батьків	%
21	Температура на вулиці	градуси Цельсія
22	Тиск	мм рт. ст.
23	RH_out, вологість на вулиці	%
24	Швидкість вітру	м/с
25	Видимість	км
26	Точка роси	градуси Цельсія
27	Випадкова величина 1	невимірна
28	Випадкова величина 2	невимірна
29	Дата і час	Рік-місяць-день година:хвилина:секунда

1.3 Методи та процес збору інформації

Вся інформація про електроспоживання в будинку була зібрана за допомогою лічильників електроенергії M-BUS, з яких дані отримуються кожні 10 хвилин та передаються в базу даних за допомогою мережі Інтернет. Провівши аналіз, можна зробити висновок, що освітлювальні прилади займають від 1 до 4 відсотків, а побутові прилади - від 70 до 79% від загальної кількості електроспоживання [4]. Всі дані були зібрані протягом 137 днів починаючи з 1 січня 2016 року.

Для збору інформації про температуру та вологість в будинку, було використано бездротову систему вузлів ZigBee [7]. З їх розташуванням

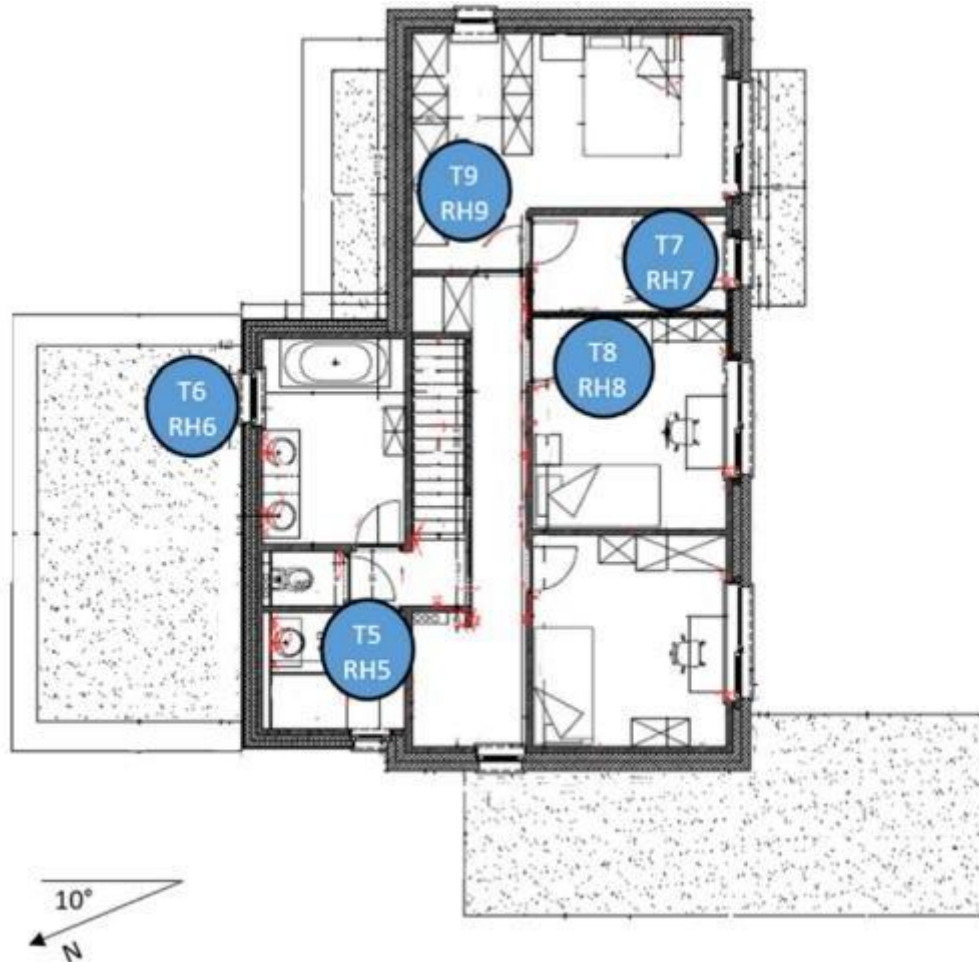
можна ознайомитись на рис. 1.1 та рис. 1.2. Точність датчиків кожного вузла для температури сягає 0.5 градусів Цельсія та 3 процента для вологості. Для отримання можливості зчитування інформації з датчиків, мікроконтролери були запрограмовані за допомогою Arduino IDE, а для передачі використано радіо XBee [8].

Дані про погоду біля будинку були отримані з найближчої метеостанції аеропорту Шевр, в Бельгії, що знаходиться в 12 кілометрах. В силу того, що дані завантажуються кожен годину, була проведена лінійна інтерполяція з інтервалом в 10 хвилин для отримання повного набору даних.

Рис. 1.1. Перший поверх [4]



Рис. 1.2. Другий поверх [4]



1.4 Аналіз та фільтрація зібраних даних

В зв'язку з кількістю даних, а також враховуючи те, що дані про погоду біля будинку були отримані від метеостанції, яка знаходиться на відстані 12 кілометрів від будинку, є необхідність зрозуміти, які характеристики є найбільш важливими, а які не поліпшують прогнозування. В роботі [4] цей процес був виконаний за допомогою алгоритму Борута [9]. Для тестування цього пакету та зниження складності були введені дві випадкові змінні. Одним з недоліків алгоритму Борута є нездатність фільтрувати вхідні дані по відношенню до середньоквадратичної похибки. Для цього, після їх впорядкування, необхідно застосувати алгоритм RFE пакету CARET

[10,11]. Після застосування було виявлено, що 34 предикати є оптимальною кількістю для прогнозування.

Під час роботи стало зрозуміло, що алгоритм Борута має ще один недолік - неможливість його застосування з вибраною моделлю через те, що вона не має необхідної функції, на відміну від, наприклад, випадковий ліс.

1.5 Прогнозування та нейронні мережі

Прогнозування – це передбачення майбутніх подій. Як результат необхідно отримати певний набір даних, який сформований на основі вхідної інформації. Нейронні мережі – це розділ штучного інтелекту, в якому для обробки сигналів використовуються явища, аналогічні тим, що відбувається в нейронах живих істот [12]. Метою прогнозування можна назвати зменшення помилки при прийнятті рішень. Цього можна досягти наданням більше ресурсів для прогнозу, а також врахувати помилки або недоліки в більш ранніх дослідженнях [13].

1.6 Оцінка раніше використаних моделей

В роботі, яка була зроблена раніше, для прогнозування було використано чотири моделі:

- linear model,
- SVM - support-vector machine,
- random forest,
- GBM - gradient boosting machine.

Вхідні дані були розбиті на навчання та тестування у співвідношенні 75% на 25% відповідно [4].

Модель множинної лінійної регресії – це практична статистична модель для оцінки зв'язків між безперервною залежною змінною та змінними-предикторами. Зі збільшенням кількості вхідних параметрів, збільшується відсоток поясненої дисперсії вихідної змінної [14]. Тобто при додаванні ознак в модель мають зменшуватись метрики помилок, але на практиці модель має найгірші показники (таблиця 1.2).

Машини опорних векторів – модель, алгоритм тренування якої, отримуючи нові зразки ознак, відносить одної за декількох груп за допомогою бінарного класифікатора [15]. Після дослідження цього підходу при тестуванні, з'ясувалось, що модель є кращою за лінійну (таблиця 1.2), але гіршою за інші дві.

Дерева рішень починаються з базового запитання. Звідти можна задати серію запитань, щоб визначити відповідь. Ці питання складають вузли рішення в дереві, діючи як засіб для поділу даних. Кожне питання допомагає прийняти остаточне рішення, яке буде визначено листковим вузлом [16].

Машина збільшення градієнта працює шляхом послідовного додавання предикторів до ансамблю, при цьому кожен з них виправляє помилки свого попередника, де посилення градієнта тренується на залишкових помилках попереднього предиктора. Використовується назва «підвищення градієнта», оскільки воно поєднує в собі алгоритм градієнтного спуску та метод підвищення [17].

Таблиця 1.2. Показники моделей за різними оцінками

Модель	Тренування			Тестування		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
LM	93.21	53.13	61.32	93.18	51.97	59.93
GBM	17.9	12.24	16.66	66.21	35.24	38.65
SVM	17.56	11.97	16.27	66.65	35.22	38.29
RF	29.61	13.75	13.43	68.48	31.85	31.39

1.7 Багатошаровий перцептрон

Багатошаровий перцептрон - це множина вхідних вузлів або вузлів джерела, що складаються собою вхідний шар, де вхідні дані рухаються в прямому напрямі по мережі від шару до шару, одного або більше прихованих шарів обчислювальних нейронів і одного вихідного шару.

Варто зазначити особливості цієї моделі, які в свою чергу дають розуміння причини її вибору для прогнозування:

1. Кожен нейрон мережі має нелінійну функцію активації. Сигмоїдальна, гіперболічний тангенс, логістична або ReLu функції виступають в якості активаційних.
2. В мережі є один або більше прихованих шарів нейронів. Вони дають можливість отримувати рішення складних задач, послідовно беручи послідовно найважливіші ознаки з вхідного вектора.
3. Мережа має високий ступінь зв'язності, отримана від синоптичних з'єднань.

Архітектурний граф багатошарового перцептронну з двома прихованими та одним вихідним шаром, яка продемонстрована на рис. 1.3, є повнозв'язною. В свою чергу це характеризує те, що кожен нейрон в будь-якому шарі пов'язаний з усіма іншими вузлами попереднього шару.

- Вхідний шар має в собі вхідні дані моделі
- Приховані шари знаходяться між вхідним і вихідним шарами, саме в них відбуваються всі обрахунки, а їх кількість визначає глибину нейронної мережі.
- Вихідний шар містить в собі результати роботи функції, змодельованої нейронною мережею і застосованої до вхідних даних.

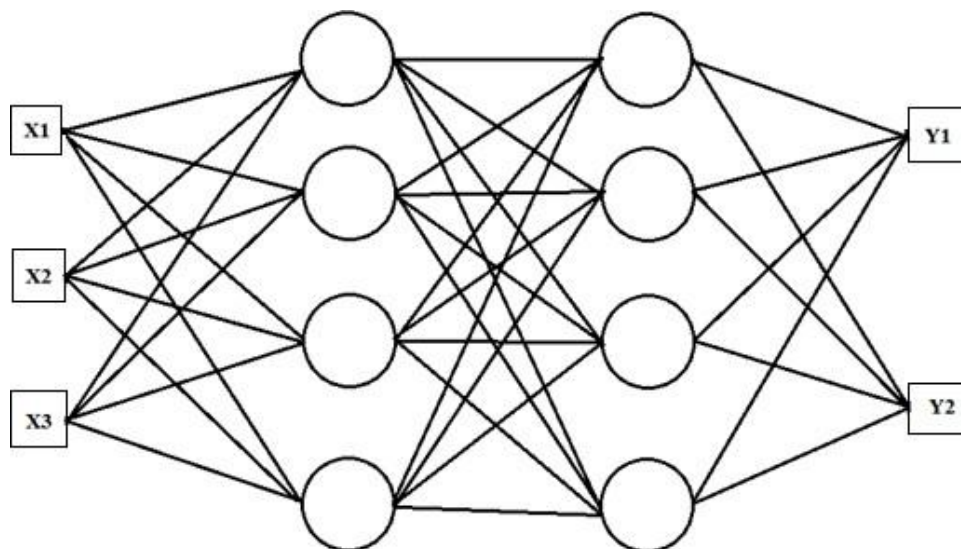


Рис.1.3. Багатошаровий перцептрон

Вибір моделі багатошарового перцептрону також обумовлюється тим, що її не було використано в роботі [4].

1.8 Постановка задач роботи

Метою випускної кваліфікаційної роботи є розробка моделі багатошарового перцептрону для прогнозування споживання електроенергії побутовими приладами будинку на прикладі набору даних “Appliances energy prediction”.

Під час виконання **випускної кваліфікаційної роботи** поставлені такі задачі:

1. Провести аналіз літератури щодо прогнозування споживання електроенергії побутовими приладами будинку.

2. Опанувати методику прогнозування неперервних кількісних величин на основі моделі багат шарового перцептрон за допомогою мови Python.
3. Використовуючи набір даних “Appliances energy prediction”, побудувати модель багат шарового перцептрон для прогнозування споживання електроенергії побутовими приладами будинку.
4. Провести дослідження із підвищення точності прогнозування побудованої моделі.

РОЗДІЛ 2. МЕТОДИКА ДОСЛІДЖЕНЬ

2.1 Збір даних

Будинок, в якому відбувався збір інформації, знаходиться в місті Стамбург, що в 24 кілометрах від Монса, в Бельгії. Його будівництво було завершено в грудні 2015 року. Згідно до сертифікації, за якою була побудована будівля, його щорічне теплове та охолоджуюче навантаження має бути не більше $15 \text{ кВт} \cdot \text{год}$ на один метр квадратний [18]. Дерев'яний димохід забезпечує більшу частину теплового навантаження будинку. Загальна кількість деревини та її сорт реєструвалась в кілограмах щомісячно вручну, а витік повітря в будинку була виміряна в вересні 2016 року і сягала 0.6 повітрообміну за годину при тиску в 50 Па. Будинок було спроектовано, таким чином, щоб густина теплового потоку сягала не більше 0.1 Вт на метр квадратний для зовнішніх стін, даху та підлоги. Вентиляція забезпечувалась за допомогою вентиляційної установки з рекуперацією тепла. Загальна площа будинку становить 280 метрів квадратних, з яких отоплюємими є 220 метрів квадратних. В будинку зазвичай знаходяться четверо людей: два дорослих та двоє підлітків, один з дорослих регулярно працює в домашньому офісі. Для більш детального розуміння техніки, яка безпосередньо впливає на дані, зібрані з будинку, наведено таблицю 2.1.

Таблиця 2.1. Перелік обладнання в кімнатах

Кімната	Обладнання
Пральня	Холодильник, морозильна камера, винний погреб, пральна машина, сушка, інтернет - маршрутизатор, інтернет - центр, мережеве сховище
Гараж	Насос для дощової води, електричні гаражні ворота
Кухня	Холодильник, індукційна плита, кухонна витяжка, мікрохвильова піч, духовка, посудомийна машина, кавоварка
Їдальня	Підсилювач Wi - Fi, координатор ZigBee, електричні жалюзі
Вітальня	Телевізор з діагоналлю 138 см, корпус для жорсткого диска, DVD-плеєр, кабельна коробка, ноутбук, струменевий принтер, електричні жалюзі
Офіс	Два персональних комп'ютери, три монітори, один роутер, один ноутбук, один копіювальний апарат, електричні жалюзі
Кімната для прасування	Будильник, радіо, праска, електричні жалюзі
Кімната 1	Будильник, радіо, електричні жалюзі, дві лампи
Кімната 2	Настільний комп'ютер, монітор, будильник, електричні жалюзі
Кімната 3	Ноутбук, будильник
Кімната для ігор	Телевізор з діагоналлю 93 см, інтернет-роутер, DVD-плеєр
Ванна 1	Електричні зубні щітки, PlayStation 2, фен
Ванна 2	Дві електричні зубні щітки

2.2 Інструментарій досліджень

Python став загальноприйнятою мовою для багатьох сфер застосування науки про дані. Він поєднує в собі переваги мов програмування з простотою використання предметно орієнтованих скриптових мов. У Python є бібліотеки для завантаження даних, візуалізації, статистичних обчислень, обробки зображень та багато чого іншого. Цей великий набір інструментів пропонує спеціалістам з роботи з даними великий набір інструментів загального та спеціального призначення. Одним з основних переваг використання Python є можливість безпосередньо працювати з програмним кодом за допомогою терміналу чи інших інструментів типу Jupyter Notebook. Машинне навчання та аналіз даних – це в основному ітераційні процеси, у яких дані задають перебіг аналізу. Вкрай важливо для цих процесів мати інструменти, які дозволяють оперативно та легко працювати з ними. Як мова програмування загального призначення Python дозволяє створювати складні графічні інтерфейси і веб-сервіси, а також легко інтегруватися в вже існуючі системи [19].

Мова програмування Python була використана як інструмент для розробки програмного коду. Вона має ряд переваг для розвитку нейромереж:

1. Python - об'єктно-орієнтована мова програмування високого рівня, що означає роботу з об'єктами згідно з задачами, рішення яких необхідно отримати.
2. Структури даних високого рівня роблять її зручною для швидкої розробки програм, а також легко зв'язувати компоненти, необхідні для реалізації.
3. Достатня кількість бібліотек для машинного навчання нейронних мереж та роботи з даними.

В якості середовища розробки було використано середовище розробки Google Colaboratory. Цей сервіс є безкоштовним та зручним, для машинного навчання, оскільки робота відбувається в браузері. Google Colaboratory поширений серед експертів та дослідників в галузі штучного навчання та нейронних мереж.

Ресурсами середовища Google Colaboratory є Ubuntu 17.10 64 біт та процесор Intel Xeon з двома ядрами 2,3 ГГц.

Scikit-learn – проект із відкритим вихідним кодом, що дозволяє вільно використовувати і поширювати, і будь-яка людина може легко отримати вихідний код, щоб побачити, що відбувається «за лаштунками». Scikit-learn містить низку сучасних алгоритмів машинного навчання, а також повну документацію щодо кожного алгоритму. Scikit-learn – дуже популярний інструмент і найвідоміша Python бібліотека для машинного навчання. Вона широко використовується в промисловості та науці, а в інтернеті є багатий вибір навчальних матеріалів та прикладів програмного коду [19].

Для навчання нейронної мережі є можливість скористатись рядом бібліотек мови Python:

1. Pandas – бібліотека для обробки та аналізу даних. Вона побудована на основі структури даних DataFrame та змодельованого за принципом дата фреймів статистичного середовища програмування R. Інакше кажучи, DataFrame бібліотеки pandas є таблицею, схожою на електронну таблицю Excel. Бібліотека pandas пропонує великий спектр методів роботи з цією таблицею, зокрема, вона дозволяє виконувати SQL-подібні запити та приєднання таблиць.
2. NumPy — містить функціональні можливості для роботи з багатовимірними масивами, високорівневими математичними функціями (операції лінійної алгебри, перетворення Фур'є, генератор псевдовипадкових чисел). Scikit-learn приймає дані як масивів

NumPy. Будь-які дані, які ви використовуєте, повинні бути перетворені на масив NumPy. Базовий функціонал NumPy – це клас ndarray, багатовимірний масив. Всі елементи масиву мають бути одного й того самого типу.

3. Seaborn – бібліотека для створення статистичних графіків на Python . Вона побудована на основі matplotlib і тісно інтегрується із структурами даних Pandas. Seaborn допомагає вам вивчити та зрозуміти дані.

2.3 Визначення параметрів

Для роботи з моделлю багатошарового перцептронного було досліджено ряд параметрів цього методів та проаналізовано за різними показниками. До початку застосування прихованих шарів, було досліджено наступні параметри функції багатошарового перцептронного:

1. **random_state** - визначає генерацію випадкових чисел для ініціалізації ваг і зміщення. Застосування параметру можливе, якщо параметр solver = 'sgd' або 'adam';
2. **alpha** - параметр штрафу (термін врегулювання);
3. **learning_rate_init** - початкова швидкість навчання. Контролює розмір кроку при оновленні ваг. Використовується лише тоді, коли solver='sgd' або 'adam'.
4. **solver** - алгоритм оптимізації, що використовується для оптимізації ваг і має чотири показники:
 - 'lbfgs' - оптимізатор із сімейства квазіньютонівських методів;
 - 'sgd' - відноситься до стохастичного градієнтного спуску;
 - 'adam' - відноситься до оптимізатора на основі стохастичного градієнта;

5. **max_iter** - максимальна кількість ітерацій. Виконує ітерації до збіжності або до цієї кількості ітерацій. Для стохастичних вирішувачів визначає кількість епох (скільки разів буде використана кожна точка даних), а не кількість кроків градієнта;
6. **n_iter_no_change** - максимальна кількість епох, які не мають покращення. Ефективний лише тоді, коли solver='sgd' або 'adam'.

Важливим є те, що залежність параметрів є нелінійною, що означає необхідність комбінації їх між собою. Для дослідження необхідно дослідити кожний параметр і застосовувати їх разом. Для знаходження оптимальних показників було автоматизовано підбір значень параметрів за допомогою вкладених циклів. Також було досліджено використання прихованих шарів разом з рядом параметрів:

1. **activation** - функція активації для прихованого шару. Має чотири показники:
 - 'identity' - активація без операцій, $f(x) = x$;
 - 'logistic' - логістична сигмовидна функція, повертає $f(x) = 1 / (1 + \exp(-x))$;
 - 'tanh' - гіперболічна функція, повертає $f(x) = \tanh(x)$;
 - 'relu' - лінійна одинична функція, повертає $f(x) = \max(0, x)$.
2. **hidden_layer_sizes** - і-ий елемент представляє кількість нейронів в і-му прихованому шарі;
3. **early_stopping** - визначає, чи використовувати ранню зупинку для припинення навчання, якщо оцінка перевірки не покращується. Якщо встановлено значення true, він автоматично відкладе 10% навчальних даних як перевірку та припинить навчання, коли оцінка перевірки не покращиться принаймні протягом n_iter_no_change послідовних епох. Ефективний лише тоді, коли solver='sgd' або 'adam'.

Багатошаровий перцептрон навчається ітераційно, оскільки на кожному кроці часу обчислюються часткові похідні функції втрат щодо параметрів моделі для оновлення параметрів. Ця реалізація працює з даними, представленими у вигляді щільних і розріджених масивів значень з плаваючою комою.

2.4 Метрики якості моделей

Для порівняння продуктивності цих регресійних моделей, використано різні показники оцінки:

- RMSE - середньоквадратична помилка, відповідає очікуваному значенню квадратичної помилки,

$$RMSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

- MAE - середня абсолютна похибка, середнє арифметичне з абсолютних похибок усіх вимірювань,

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i|$$

- MAPE - середня абсолютна помилка у відсотках.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i - \hat{Y}_i|}{|Y_i|}$$

2.5 Відбір ознак

Для відбору ознак використано функцію перестановки. Ця техніка перевірки моделі, яка може бути використана для будь-якої метрики якості або набору метрик якості у випадку, коли дані представлені таблицею. Це означає, що є можливість застосувати необхідні метрики, наприклад в даній роботі це RMSE, MAE та MAPE. Процедура перемішування значення ознаки випадковим порушує зв'язок між ознакою та величиною, що прогнозується, тому падіння оцінки моделі вказує на те, наскільки модель

залежить від ознаки. Перевагою цього методу є те, що він не залежить від моделі і його можна обчислити багато разів з різними перестановками ознаки. Цей метод реалізований в проекті scikit-learn під назвою **permutation_importance**. Разом з використанням методів метрик якості, цей інструмент є дуже зручним для підбору ознак.

РОЗДІЛ 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

3.1 Дослідження за допомогою багатошарового перцептронну

Для проведення досліджень, створено модель, яка використовує ознаки, вибрані для перевірки моделей в роботі [4], що було досліджено. Для порівняння використано три метрики. На рисунку 3.1 синім кольором зазначені результати роботи моделі з опрацьованої роботи [4], жовтим – розроблена модель багатошарового перцептронну. Всі наведені значення отримані на тестовій вибірці. За метриками оцінки помилок, багатошаровий перцептрон переважає найкращу модель, отриману в опрацьованій роботі, тобто GBM, яка використовувала ті самі ознаки. З цього можна зробити висновки, що розроблена модель здатна робити менше помилок при прогнозуванні споживання електроенергії. Результати наведено на рис 3.1.

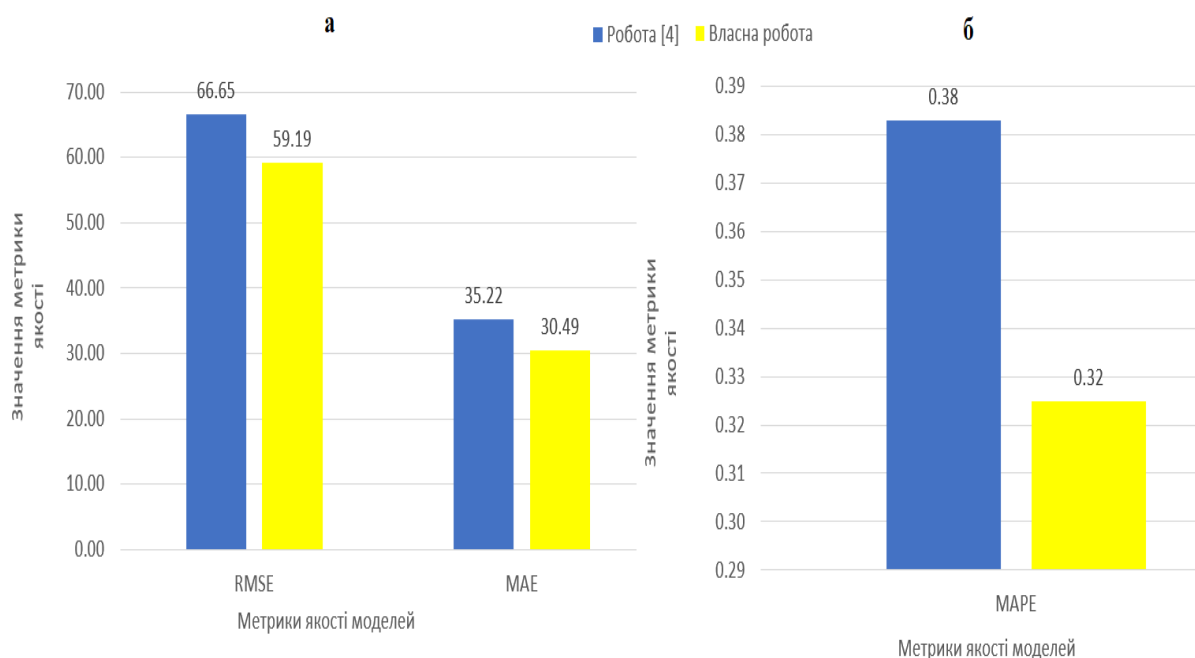


Рис 3.1. Ілюстрація порівняння моделей за трьома метриками при виборі всіх ознак

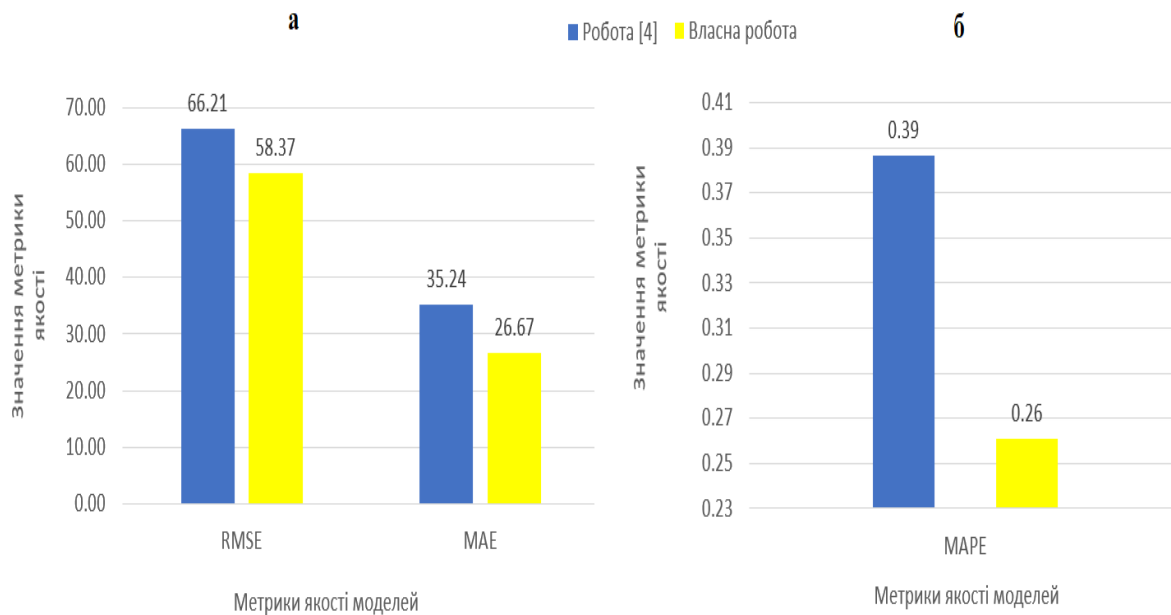


Рис 3.2. Ілюстрація порівняння моделей за трьома метриками за відсутності світла як ознаки

З метою покращення показників метрики в опрацьованій роботі [4] були змінені ознаки, які використовувались для навчання моделі GBM. Найкращі показники метрики якості досягнуті без використання світла, яка ознаки. Для моделі багат шарового персептронну вибір аналогічного набору вхідних даних також дозволив покращити показники метрик якості. Так показники помилок – RMSE, MAE та MAPE – стали меншими, як в порівнянні з моделлю багат шарового персептронну при всіх ознаках, так і відносно моделі GBM. Результати наведено на рис 3.2.

Досягти найкращих значень метрик вдалось після підбору власної комбінації ознак – тиску, температуру та вологості всередині будинку, а також день тижня. Так було виявлено ті вхідні дані, за яких побудована модель багатошарового перцептронну має найкращі показники метрик якості, що продемонстровано на рис 3.3. Таким чином після проведення досліджень, вдалось досягти найнижчих показників метрик помилок - RMSE, MAE, MAPE – в порівнянні з моделлю опрацьованої роботи [4] та розробленою моделлю багатошарового перцептронну з будь яким іншим набором вхідних ознак.

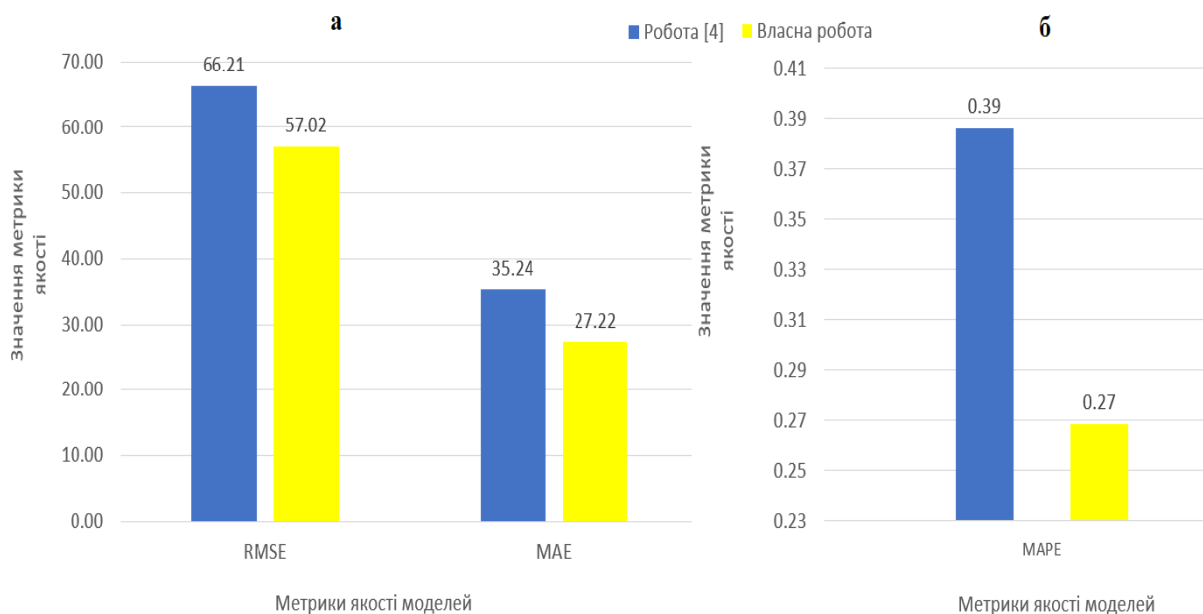


Рис 3.3. Ілюстрація порівняння моделей за трьома метриками з використанням власної комбінації ознак

3.2 Аналіз отриманих результатів

При розробці багатошарового перцептронну для прогнозування споживання електроенергії було досягнуто нижчих значень метрик RMSE, MAE та MAPE, ніж у роботі [4], опрацьованій мною, що демонструє здатність розробленої моделі допускати менше помилок при прогнозуванні, ніж модель з найнижчими метриками помилок з опрацьованої роботи [4]. Ці результати свідчать про те, що багатошаровий перцептрон здатний

точніше прогнозувати вихідну величину і є доцільнішим для цього набору ознак. Також після відбору ознак вдалось з'ясувати, що використання показників погоди є зайвими. Це дозволяє обмежитись лише ознаками температури та тиску всередині будинку, світлом та днем тижня, що дозволяє суттєво полегшити збір ознак.

ВИСНОВКИ

В результаті проведення досліджень в рамках випускної кваліфікаційної роботи:

1. Проаналізовано літературу щодо прогнозування споживання електроенергії побутовими приладами будинку;
2. Опановано методику прогнозування неперервних кількісних величин на основі моделі багатошарового перцептронів за допомогою мови Python.
3. Використовуючи набір даних “Appliances energy prediction”, побудовано модель багатошарового перцептронів для прогнозування споживання електроенергії побутовими приладами будинку.
4. Досліджено, що архітектура, за якої багатошаровий перцептрон має два шари по 80 і 60 нейронів у кожному, дозволяє досягти метрик якості кращих, ніж у опрацьованій роботі [4].
5. Для збільшення точності прогнозування, проведено відбір ознак, який дозволяє значно полегшити збір вхідних даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. A. Kavousian, R. Rajagopal, M. Fischer, Ranking appliance energy efficiency in households: Utilizing smart meter data and energy efficiency frontiers to estimate and identify the determinants of appliance energy efficiency in residential buildings, *Energy Build.* 99 (2015) 220–230.
2. K.S. Cetin, P.C. Tabares-Velasco, A. Novoselac, Appliance daily energy use in new residential buildings: Use profiles and variation in time-of-use, *Energy Build.* 84 (2014) 716–726.
3. S. Firth, K. Lomas, A. Wright, R. Wall, Identifying trends in the use of domestic appliances from household electricity consumption measurements, *Energy Build.* 40 (5) (2008) 926–936.
4. Luis M. Candanedo, Véronique Feldheim, Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. 140 (2017) 81–97
5. W.F. Sandusky, E.W. Pearson, N.E. Miller, R.S. Crowder, G.B. Parker, R.P. Mazzucchi, G.M. Stokes, J.J. Thomas, R.G. Pratt, G.J. Schuster, M.A. Halverson, J.L. Stoops, F.J. Peterson, R.A. Gillman, R.A. Stokes, S.G. Hauser, ELCAP operational experience, *Energy Build.* 19 (3) (1993) 167–178.
6. C.-L. Hor, S.J. Watson, S. Majithia, Analyzing the impact of weather variables on monthly electricity demand, *IEEE Trans. Power Syst.* 20 (4) (2005) 2078–2085.
7. Z. Alliance, Zigbee Specification, 2006.
8. R. Faludi, Building wireless sensor networks: with ZigBee, XBee, arduino, and processing, O'Reilly Media, Inc., 2010.
9. M.B. Kursu, W.R. Rudnicki, Feature selection with the boruta package, *J. Stat. Softw.* 36 (11) (2010) 1–13.
10. C. Fan, F. Xiao, S. Wang, Development of prediction models for next-

- day building energy consumption and peak power demand using data mining techniques, *Appl. Energy* 127 (2014) 1–10.
11. M. Kuhn, caret: Classification and Regression Training, 2015.
 12. Gruau F. Automatic Definition of Modular Neural Networks. // *Adaptive Behavior*. – 1995. – №3(2). – P. 151–183.
 13. Солдатов О.П., Семенов В.В. Применение нейронных сетей для решения задач прогнозирования, 2006.
 14. Multiple linear regression // IBM. URL: <https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=tests-multiple-linear-regression> (дата звернения 12.02.2022)
 15. Radial Kernel SVM // Amazon. URL: https://rstudio-pubs-static.s3.amazonaws.com/296261_b5acc05b2b0e41879c917033b1497543.html (дата звернения 13.02.2022)
 16. Random Forest // IBM. URL: <https://www.ibm.com/docs/en/cognos-analytics/11.1.0?topic=tests-multiple-linear-regression> (дата звернения 12.02.2022)
 17. Boosting // IBM. URL: <https://www.ibm.com/cloud/learn/boosting> (дата звернения 12.02.2022)
 18. Андреас Мюллер, Сара Гвидо. Введение в машинное обучение с помощью Python, 2016 - 2017.
 19. Colaboratory // Google. URL: <https://research.google.com/colaboratory/faq.html> (дата звернения 13.02.2022)

ДОДАТОК

Додаток А Код програми

```
from sklearn.neural_network import MLPRegressor
from sklearn.datasets import make_regression
from numpy import genfromtxt
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from math import sqrt

df =
pd.concat([pd.read_csv('https://raw.githubusercontent.com/LuisM78/Appliance-s-
energy-prediction-data/master/training.csv'),

pd.read_csv('https://raw.githubusercontent.com/LuisM78/Appliances-energy-
prediction-data/master/testing.csv')])
df = df.sort_values(by=['date'])
df = df[['date', 'Appliances', 'T1', 'T2', 'RH_2', 'T3', 'RH_3', 'RH_5',
'RH_6', 'T7', 'T8', 'RH_8', 'T9', 'RH_9', 'Day_of_week']]
train_size = 14803 # round(len(df) * 0.75)
test_size = len(df) - train_size

df.Day_of_week.replace({'Monday': 0, 'Tuesday': 1, 'Wednesday': 2,
'Thursday': 3, 'Friday': 4, 'Saturday': 5,
'Sunday': 6}, inplace=True)

window_size = 10
x_train = []
for i in range(window_size, train_size):
    x = np.concatenate([np.concatenate(df.iloc[i - window_size : i ,
1:].values), df.iloc[i, 2:].values.astype('float64')])
    x_train.append(x)
x_train = np.array(x_train)
y_train = df.iloc[window_size : train_size, 1].values
train_size = 14803 # round(len(df) * 0.75)
test_size = len(df) - train_size

x_test = []
for i in range(window_size + train_size, len(df)):
    x = np.concatenate([np.concatenate(df.iloc[i - window_size : i ,
1:].values), df.iloc[i, 2:].values.astype('float64')])
    x_test.append(x)
x_test = np.array(x_test)
y_test = df.iloc[window_size + train_size: , 1].values

mlpr = MLPRegressor(hidden_layer_sizes=[80, 60], alpha=7,
learning_rate_init=0.001,
early_stopping=True, max_iter=1_000_000, n_iter_no_change=100,
validation_fraction=0.3, random_state=5)
mlpr.fit(x_train, y_train)
print(regr.score(X_test, y_test))

result = permutation_importance(mlpr, x_test, y_test, n_repeats=30,
random_state=5)
```

```
mse = mean_squared_error(y_test, regr.predict(X_test))
mse, np.sqrt(mse)
mean_absolute_error(y_test, regr.predict(X_test))
mse = mean_squared_error(y_test, regr.predict(X_test))
```