

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Архітектура комп'ютерної системи керування
мікрокліматом будинка з використанням засобів
машинного навчання
(тема)

Виконав:

студент II курсу, групи СПМ-23-1
Зубенко Д.Р.
(прізвище, ініціали)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва освітньої програми)

Керівник: доц. Іващенко Г.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

Коваленко А.А.
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту _____ Зубенку Денису Руслановичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Архітектура комп'ютерної системи керування мікрокліматом будинка з використанням засобів машинного навчання _____

затверджена наказом по університету від “ 22 ” листопада 2024 р. № 1236 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії _____ 20 січня 2025 р. _____

3. Вхідні дані до роботи _____

1) API системи Home Assistant для отримання даних; _____

2) часові ряди даних про температуру та вологість у приміщеннях; _____

3) документація бібліотеки ML.NET; _____

4) середовище розробки Microsoft Visual Studio; _____

5) набори даних для тренування та тестування моделей прогнозування; _____

6) методи обробки та нормалізації даних. _____

4. Перелік питань, що потрібно опрацювати у роботі _____

1) аналіз актуальних систем керування мікрокліматом; _____

2) огляд сучасних методів прогнозування часових рядів; _____

3) порівняння моделей ARIMA, SARIMA та LSTM для прогнозування мікроклімату; _____

4) реалізація комп'ютерної системи прогнозування мікроклімату; _____

5) інтеграція прогнозованих значень у Home Assistant для автоматизації; _____

6) оцінка точності моделей та порівняння їх результатів; _____

7) висновки та рекомендації щодо застосування обраних методів. _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) _____

Слайд-презентація – 13 слайдів _____

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	26.11.24-30.11.24	
2	Вибір та обґрунтування методики дослідження	02.12.24-05.12.24	
3	Вибір інструментальних засобів	06.12.24-10.12.24	
4	Розробка алгоритмічного забезпечення	11.12.24-21.12.24	
5	Проведення експериментів	23.12.24-03.01.25	
6	Оформлення матеріалів кваліфікаційної роботи	04.01.25-07.01.25	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	08.01.25-11.01.25	
8	Подання кваліфікаційної роботи на рецензування	13.01.25-17.01.25	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

доц. Іващенко Г.С.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 78 с., 9 рис., 4 табл., 2 дод., 27 джерел.

МІКРОКЛІМАТ, ТЕМПЕРАТУРА, ЕНЕРГОЕФЕКТИВНІСТЬ, АВТОМАТИЗАЦІЯ, ПРОГНОЗУВАННЯ, ЧАСОВІ РЯДИ, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, LSTM, ARIMA, SARIMA, HOME ASSISTANT, РОЗУМНИЙ ДІМ, СЕНСОРИ, ML.NET, ІНТЕГРАЦІЯ СИСТЕМ, АВТОМАТИЗАЦІЯ HVAC.

Метою кваліфікаційної роботи є дослідження ефективності застосування сучасних методів машинного навчання для короткострокового прогнозування часових рядів при побудові комп'ютерної системи керування мікрокліматом у житлових приміщеннях.

У ході виконання кваліфікаційної роботи розроблено комп'ютерну систему прогнозування температури на основі засобів машинного навчання, що включає використання моделей ARIMA, SARIMA та LSTM. Проведено детальний аналіз отриманих результатів, порівняння точності моделей за метриками MAE, RMSE та MAPE, а також виявлено переваги та обмеження кожного підходу.

Практичним результатом роботи стала інтеграція обраної моделі прогнозування в автоматизовану систему Home Assistant, що дозволило реалізувати адаптивні сценарії керування системами опалення, вентиляції та кондиціонування в реальному часі. Це забезпечує покращення енергоефективності, підвищення комфорту мешканців та створює можливості для подальшого вдосконалення розумних систем управління мікрокліматом.

ABSTRACT

Master's thesis: 78 pages, 9 figures, 4 tables, 2 appendices, 27 sources.

MICROCLIMATE, TEMPERATURE, ENERGY EFFICIENCY, AUTOMATION, FORECASTING, TIME SERIES, MACHINE LEARNING, NEURAL NETWORKS, LSTM, ARIMA, SARIMA, HOME ASSISTANT, SMART HOME, SENSORS, CLIMATE FORECAST, ML.NET, SYSTEM INTEGRATION, HVAC AUTOMATION.

The major goal of this thesis is to investigate the efficiency of modern computational intelligence methods for short-term time series forecasting to optimize indoor climate control in residential buildings.

In order to achieve this goal, a comprehensive methodology for temperature forecasting based on machine learning was developed. This methodology includes the use of ARIMA, SARIMA, and LSTM models. A detailed analysis of the obtained results was conducted, comparing the accuracy of the models using metrics such as MAE, RMSE, and MAPE, as well as identifying the key advantages and limitations of each approach.

As a practical outcome, the selected forecasting model was integrated into the automated Home Assistant system. This integration enabled the implementation of adaptive control scenarios for heating, ventilation, and air conditioning systems in real-time. This approach improves energy efficiency, enhances the comfort of residents, and opens opportunities for further optimization of smart indoor climate control systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Актуальність теми дослідження	10
1.2 Огляд сучасних методів керування мікрокліматом.....	10
1.3 Машинне навчання в керуванні мікрокліматом	13
1.4 Прогнозування часових рядів даних мікроклімату	15
1.5 Аналіз існуючих досліджень.....	16
1.6 Постановка задачі.....	18
2 ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ МІКРОКЛІМАТУ	19
2.1 Засоби вирішення задачі прогнозування часових рядів.....	19
2.2 Використання ML.NET для прогнозування мікроклімату	22
2.3 Принципи роботи MLContext	23
2.3.1 Основні функції MLContext	24
2.3.2 Інтеграція з іншими компонентами ML.NET.....	25
2.4 Методи прогнозування часових рядів.....	25
2.4.1 Модель ARIMA	26
2.4.2 Модель SARIMA	26
2.4.3 Модель LSTM.....	27
2.4.4 Порівняння моделей ARIMA, SARIMA та LSTM	28
2.4.5 Довга короткострокова пам'ять для прогнозування часових рядів	29
2.5 Недоліки та обмеження при використанні нейронних мереж для прогнозування.....	31
2.6 Методи оцінки точності моделей прогнозування.....	33

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ВИКОРИСТАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ.....	36
3.1 Витяг даних із Home Assistant	36
3.2 Підготовка даних.....	38
3.3 Реалізація моделі ARIMA.....	39
3.4 Реалізація моделі довгої короткострокової пам'яті (LSTM)	41
3.4 Реалізація моделі SARIMA	42
3.6 Відправлення прогнозованих значень у Home Assistant та реалізація автоматизацій	45
4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ	48
4.1 Оцінювання точності моделей прогнозування	48
4.2 Графічний аналіз результатів.....	49
4.3 Графічне порівняння метрик точності	52
4.5 Аналіз переваг SARIMA та LSTM	54
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	57
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	60
ДОДАТОК Б Вихідний код розробленого програмного засобу	68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШНМ – штучна нейронна мережа

ARIMA – авторегресійна інтегрована середня ковзна (англ., Autoregressive Integrated Moving Average)

AutoML – автоматизоване машинне навчання (англ., Automated Machine Learning)

HVAC – системи опалення, вентиляція та кондиціювання (англ., Heating, Ventilation, and Air Conditioning)

IoT – Інтернет речей (англ., Internet of Things)

LSTM – довга короткострокова пам'ять (англ., Long Short-Term Memory)

MAE – середня абсолютна помилка (англ., Mean Absolute Error)

MAPE – середня абсолютна похибка у відсотках (англ., Mean Absolute Percentage Error)

ML – машинне навчання (англ., Machine Learning)

MLContext – середовище для машинного навчання у ML.NET

MLP – багат шаровий перцептрон (англ., Multilayer Perceptron)

PCA – метод головних компонент (англ., Principal Component Analysis)

RNN – рекурентна нейронна мережа (англ., Recurrent Neural Network)

RMSE – корінь середньоквадратичної помилки (англ., Root Mean Squared Error)

SARIMA – сезонна авторегресійна інтегрована середня ковзна (англ., Seasonal ARIMA)

ВСТУП

Забезпечення комфортних умов проживання в приміщеннях є одним із головних завдань сучасного будівництва та управління будівлями. Мікроклімат у приміщенні, який включає температуру, вологість повітря та вентиляцію, безпосередньо впливає на здоров'я і самопочуття людей, а також на енергоефективність будівель. Сучасні вимоги до енергоощадності та автоматизації зумовлюють необхідність розробки інтелектуальних систем керування мікрокліматом, здатних контролювати поточні умови та прогнозувати їхні зміни для своєчасного реагування.

Технології інтернету речей (IoT) та штучного інтелекту (ШІ) відкривають нові можливості для створення систем керування мікрокліматом, що працюють на основі даних з численних сенсорів. Методи машинного навчання дозволяють обробляти ці дані та прогнозувати майбутні зміни кліматичних умов у приміщеннях, що забезпечує оптимальне регулювання роботи систем опалення, вентиляції та кондиціонування, а також мінімізує енергоспоживання.

Основним завданням таких систем є прогнозування часових рядів, зокрема температури та вологості, що дозволяє підтримувати мікроклімат на оптимальному рівні. Використання методів машинного навчання, таких як нейронні мережі, особливо моделей типу LSTM, є ефективним інструментом для вирішення цих завдань завдяки їхній здатності працювати з довгостроковими та короткостроковими залежностями в даних.

Робота присвячена дослідженню використання методів машинного навчання при створенні комп'ютерної системи керування мікрокліматом у будинках. Передбачається розробка моделі, яка буде здатна ефективно прогнозувати зміни кліматичних умов у приміщеннях і може бути використана для автоматизації керування системами опалення, вентиляції та кондиціонування для забезпечення комфорту та енергоефективності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність теми дослідження

В умовах сучасного розвитку інформаційних технологій та збільшення попиту на ефективні та енергоощадні рішення в галузі комфорту й екологічної безпеки, системи управління мікрокліматом у житлових будинках стають невід'ємною частиною смарт-інфраструктури. Такі системи дозволяють підтримувати комфортні кліматичні умови в приміщеннях, автоматизовано регулювати температуру, вологість, концентрацію CO₂ та інші показники повітря [1]. Це важливо не лише з точки зору комфорту, але й з погляду економії енергоресурсів та зниження впливу на довкілля.

Впровадження засобів машинного навчання в системи керування мікрокліматом дозволяє значно підвищити їхню ефективність, оскільки такі алгоритми здатні передбачати зміну кліматичних умов на основі минулих даних і своєчасно вносити корективи в роботу систем опалення, вентиляції та кондиціонування.

1.2 Огляд сучасних методів керування мікрокліматом

На сьогодні існує велика кількість підходів до регулювання мікроклімату в приміщеннях, які еволюціонували з простих механічних систем до складних інтелектуальних комплексів [2]. Традиційно для цього використовувалися механічні або електронні термостати.

Механічні системи керування мікрокліматом були найпростішими, реагуючи на зміну температури шляхом вмикання або вимикання опалювальних пристроїв. Хоча такі системи забезпечували базове регулювання мікроклімату, вони мали значні недоліки. Наприклад, механічні системи не здатні передбачати температурні коливання та враховувати

динаміку зміни вологості, що могло призводити до надмірного використання електроенергії [3].

З появою електронних термостатів можливості керування мікрокліматом збільшилися, оскільки ці пристрої могли здійснювати більш точний контроль за температурними параметрами. Вони дозволяли користувачам програмувати певні режими роботи в залежності від часу доби або присутності людей у приміщенні. Але ці вдосконалені системи не могли повністю задовольнити потреби сучасних користувачів. Вони не враховували зовнішні чинники, такі як зміни погоди, сонячна активність чи інші обставини, що впливають на внутрішній клімат приміщення.

З впровадженням технологій розумних будинків та систем Інтернету речей (IoT) стали можливими нові підходи до керування мікрокліматом, які ґрунтуються на аналізі великих обсягів даних. Сучасні IoT-системи здатні збирати інформацію з численних сенсорів, розташованих у приміщенні, таких як датчики температури, вологості, освітленості, вмісту вуглекислого газу тощо. Ці дані аналізуються в реальному часі для прийняття оптимальних рішень щодо керування мікрокліматом.

Зібрані дані можуть бути використані для прогнозування. На основі історичних даних про температуру, вологість, енергоспоживання та інші параметри, системи керування мікрокліматом можуть передбачати майбутні зміни. Це дозволяє, наприклад, заздалегідь ввімкнути систему охолодження у спекотні дні або опалення під час різкого похолодання. Такий підхід допомагає не лише підтримувати комфортний мікроклімат, але й суттєво знижує витрати на енергію.

Методи машинного навчання дозволяють системам не лише реагувати на поточні показники сенсорів, але й робити прогнози та приймати рішення на основі аналізу великих обсягів даних. Завдяки цьому, система керування мікрокліматом стає більш автономною, адаптуючи свою роботу в залежності від змін у навколишньому середовищі та індивідуальних уподобань користувачів.

Методи прогнозування дозволяють не лише оптимізувати роботу систем опалення та охолодження, але й інтегрувати інші аспекти управління будинком. Наприклад, інтелектуальні системи можуть контролювати вологість, освітленість, забезпечувати вентиляцію або моніторинг рівня вуглекислого газу для забезпечення здорового клімату в приміщенні.

Однією з важливих особливостей інтелектуальних систем є їхня інтеграція з мобільними застосунками, що дозволяє користувачам контролювати мікроклімат віддалено. Наприклад, є можливість налаштувати температуру у своєму будинку перед тим, як повернутися додому, або перевірити стан систем опалення під час відсутності.

Сучасні системи керування мікрокліматом можуть адаптуватися до звичок користувача. Вони аналізують поведінку мешканців будинку, їхні уподобання щодо температури і графік присутності, і на основі цього приймають рішення про те, коли і як регулювати системи опалення або кондиціонування. Це значно підвищує рівень комфорту і робить систему більш ефективною з точки зору енергоспоживання.

Серед сучасних підходів також можна виділити використання гібридних систем, які поєднують різні методи регулювання мікроклімату. Наприклад, системи можуть одночасно використовувати термостати, датчики і машинне навчання для досягнення максимальної точності в регулюванні температури та інших параметрів. Такий підхід дозволяє враховувати всі фактори, що впливають на мікроклімат, і забезпечує оптимальні умови для проживання.

Таким чином, сучасні методи керування мікрокліматом спрямовані на забезпечення максимальної автономії та ефективності. Завдяки використанню машинного навчання, аналізу великих обсягів даних та інтеграції з IoT-системами, стало можливим створювати розумні будинки, які не лише забезпечують комфорт, але й сприяють зниженню витрат на енергоресурси та захисту навколишнього середовища [4].

1.3 Машинне навчання в керуванні мікрокліматом

Машинне навчання дозволяє реалізацію «розумних» систем управління мікрокліматом, які можуть не лише реагувати на поточні зміни, але й передбачати їх [5]. Використання алгоритмів машинного навчання дозволяє аналізувати великі обсяги зібраних історичних даних, отриманих від датчиків, і на основі цих даних створювати моделі для прогнозування змін у середовищі. Це забезпечує точніше управління системами вентиляції, опалення та кондиціонування, знижуючи витрати енергії та підвищуючи комфорт користувачів.

Важливою перевагою засобів машинного навчання є здатність адаптуватися до нових умов, ґрунтуючись на зібраних даних. Системи на основі машинного навчання здатні «вчитися» на основі інформації, отриманої з датчиків температури, вологості, освітленості та інших параметрів, і поступово вдосконалювати свої прогнози. Це робить їх більш точними та надійними в процесі роботи. Окрім того, такі системи дозволяють враховувати індивідуальні вподобання мешканців будинку. Наприклад, система може автоматично налаштовувати температурні режими відповідно до звичок користувачів, забезпечуючи комфорт без необхідності ручного втручання. Якщо один із мешканців надає перевагу прохолоднішій температурі під час сну, система, на основі забраних даних про раніше встановлені користувачем параметри опалення, здатна навчитися враховувати ці індивідуальні потреби та забезпечувати відповідний мікроклімат.

Завдяки використанню машинного навчання, системи управління мікрокліматом можуть не лише підтримувати комфортні умови, але й прогнозувати зміни в навколишньому середовищі. Наприклад, система може передбачити, що температура в приміщенні підвищиться через сонячне випромінювання в певний час доби, і заздалегідь скоригувати роботу кондиціонера, щоб уникнути перегріву. Крім того, машинне навчання

допомагає виявляти аномальні ситуації. Якщо, наприклад, температура стрімко зростає або падає, це може свідчити про несправності в системах опалення, вентиляції чи кондиціонування. Системи можуть аналізувати великі обсяги історичних даних і виявляти патерни, що ведуть до таких аномалій. Це дозволяє уникнути багатьох проблем, автоматично виправляючи ситуацію або сповіщаючи користувача про необхідність втручання.

Однією з найбільш важливих переваг використання машинного навчання в системах керування мікрокліматом є здатність значно знизити енергоспоживання. За рахунок точного прогнозування змін температури та інших кліматичних факторів система може завчасно знижувати або підвищувати інтенсивність роботи кондиціонера або обігрівача, щоб уникнути надмірного використання енергії. Наприклад, якщо прогнозується підвищення температури через сонячну активність, система може заздалегідь знизити потужність обігрівача або вентиляційної системи. Це не лише допоможе зекономити енергію, але й підвищить загальний комфорт у приміщенні, оскільки зміни температури будуть поступовими і менш відчутними для користувачів. Крім того, системи на основі машинного навчання можуть оптимізувати свою роботу з огляду на споживання енергії в попередні періоди. Аналізуючи історичні дані, представлені у вигляді часових рядів, система може навчитися працювати ефективніше, мінімізуючи витрати без шкоди для комфорту мешканців.

Використання сучасних моделей машинного навчання та засобів прогнозування, таких як ARIMA, LSTM або SARIMA, дає змогу ефективніше обробляти великі обсяги даних та враховувати як лінійні, так і нелінійні залежності в часових рядах.

Ще однією перевагою машинного навчання є можливість адаптації до зовнішніх умов. Системи керування мікрокліматом можуть враховувати дані про зовнішню температуру, вологість, рівень сонячного освітлення та інші фактори. Наприклад, система може автоматично налаштувати роботу

опалення в залежності від прогнозу погоди, щоб знизити споживання енергії, коли очікується теплий день. Також можливість інтеграції з іншими компонентами розумного будинку, такими як датчики руху або приводи штор, дозволяє створювати комплексні моделі управління. Система може не лише реагувати на зміни в кліматі, але й враховувати, чи є люди в приміщенні, чи відкриті вікна, або навіть відрегулювати інтенсивність опалення в залежності від часу доби або активності мешканців.

Інтелектуальні системи, що базуються на машинному навчанні, постійно вдосконалюються шляхом аналізу нових даних. Це дозволяє системам відстежувати сезонні патерни, адаптуватися до нових умов та підвищувати точність своїх прогнозів. Таким чином, системи управління мікрокліматом поступово досягають більш високої автономності та ефективності, що сприяє оптимізації енергоспоживання та забезпеченню постійного комфорту для користувачів.

1.4 Прогнозування часових рядів даних мікроклімату

Прогнозування часових рядів даних мікроклімату є одним із найефективніших підходів для визначення майбутніх змін температури або інших параметрів мікроклімату. Часовий ряд – це послідовність даних, зібраних протягом певного періоду часу з однаковими інтервалами. Ці інтервали можуть бути щохвилинними, щогодинними або щоденними вимірами, наприклад, температури, вологості, рівня вуглекислого газу або інших екологічних факторів. Для системи керування мікрокліматом важливо враховувати не лише поточні параметри, але й історичні дані для прогнозування майбутніх умов і забезпечення комфортного середовища.

Основне завдання при роботі з часовими рядами – це навчання моделі на основі попередніх значень для прогнозування майбутніх. Наприклад, маючи дані про зміну температури в приміщенні протягом останнього тижня, можна спрогнозувати, як температура змінюватиметься протягом наступних

кількох годин. Це дозволяє системі мікроклімату діяти на випередження, регулюючи роботу систем опалення, вентиляції та кондиціонування таким чином, щоб уникнути різких змін температури або надмірного енергоспоживання [16]. У контексті прогнозування мікроклімату важливими є такі параметри, як:

- температура: один із головних факторів, що впливає на комфорт у приміщенні;
- вологість: висока або низька вологість може негативно впливати на самопочуття людей і стан матеріалів у будівлі;
- CO₂: рівень вуглекислого газу є індикатором якості повітря, особливо в приміщеннях з великим скупченням людей;
- тиск і швидкість вітру: ці показники можуть мати значення для великих або відкритих приміщень, де природні фактори відіграють важливу роль у забезпеченні оптимального мікроклімату.

Часові ряди для вказаних параметрів збираються за допомогою датчиків у реальному часі, що дає змогу системі керування мікрокліматом аналізувати історичні дані та враховувати їх при прогнозуванні. Цей підхід дозволяє вчасно реагувати на зміни та підтримувати постійний комфортний рівень у приміщенні.

1.5 Аналіз існуючих досліджень

Для формування задачі та вибору методів обчислювального інтелекту для подальшої реалізації та вдосконалення було проаналізовано низку досліджень у сфері прогнозування часових рядів, зокрема стосовно використання сучасних засобів прогнозування, таких як ARIMA, LSTM та SARIMA, для вирішення задач, пов'язаних із мікрокліматом та іншими релевантними завданнями [6].

Дослідження, проведене у 2020 році, порівнює моделі ARIMA та LSTM для прогнозування мікроклімату в будинках. Згідно з результатами, модель

LSTM перевершила ARIMA за точністю у передбаченні змін температури та вологості в умовах швидкої динаміки клімату, що підтверджує ефективність нейронних мереж для роботи зі складними часовими рядами з лінійними та нелінійними залежностями [7]. Основна перевага LSTM полягає у здатності враховувати довготривалі залежності, що є важливим для точного прогнозування в мінливих умовах середовища. Водночас, модель ARIMA також демонструє надійність у прогнозуванні стаціонарних процесів і може бути корисною для короткострокового прогнозування за стабільних умов.

У дослідженні 2019 року розглянуто застосування моделі SARIMA для прогнозування сезонних кліматичних змін у будинках [11]. Результати показали, що SARIMA ефективно працює з даними, які мають виражену сезонність, наприклад, із річними температурними коливаннями. Однак метод SARIMA виявив певні обмеження у випадках, коли дані мають складнішу структуру або нерегулярні сезонні патерни, тоді як LSTM показала здатність краще адаптуватися та зберігати точність прогнозів завдяки рекурентним особливостям [11].

Розглянуто дослідження 2021 року присвячено можливості застосування глибоких нейронних мереж для прогнозування параметрів мікроклімату [25]. Встановлено, що модель LSTM може забезпечувати високу точність при прогнозуванні змін температури та здатна адаптуватися до нових даних у реальному часі, що особливо важливо для інтеграції з іншими компонентами «розумного будинку». Дослідження підтверджує переваги LSTM для точного прогнозування в умовах динамічних змін завдяки її гнучкості й здатності до безперервного навчання.

З огляду на розглянуті дослідження, можна зробити висновок, що методи машинного навчання, зокрема LSTM та SARIMA, є перспективними для вирішення завдань прогнозування мікроклімату. Використання цих підходів дозволяє розробляти системи з високою точністю прогнозування, особливо для динамічних середовищ, де важливо враховувати різні типи залежностей у даних.

1.6 Постановка задачі

Метою роботи є дослідження використання методів короткострокового прогнозування часових рядів на основі засобів машинного навчання для оптимізації керування мікрокліматом у житлових приміщеннях. Зокрема, задача полягає у прогнозуванні температурних показників на основі історичних даних про температуру, вологість та інші параметри внутрішнього середовища, що дозволяє своєчасно адаптувати роботу систем опалення, вентиляції та кондиціонування для підтримки комфортних умов і оптимізації енергоспоживання.

Як дані для дослідження використовуються часові ряди температури і вологості повітря у приміщенні, які можуть коливатися залежно від часу доби, погодних умов, зовнішньої температури та діяльності мешканців. Прогнозування цих параметрів дозволить створити систему, яка реагує на потенційні зміни в мікрокліматі до їх настання, тим самим знижуючи витрати енергії на обігрів або охолодження приміщення.

Для вирішення задачі були обрані кілька моделей машинного навчання, серед яких моделі часових рядів, такі як ARIMA та SARIMA, а також нейронні мережі, зокрема модель довгої короткострокової пам'яті (LSTM), яка доцільна для роботи з довготривалими процесами. Необхідно проаналізувати ефективність цих моделей для прогнозування змін мікроклімату на основі доступних даних, що дозволить обрати найбільш точну і надійну модель для впровадження в реальну систему керування.

Таким чином, основна мета роботи полягає у побудові моделі прогнозування мікроклімату, яка дозволить покращити якість життя мешканців приміщень шляхом ефективного керування температурним режимом та забезпечення енергоефективності системи.

2 ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ МІКРОКЛІМАТУ

2.1 Засоби вирішення задачі прогнозування часових рядів

Часові ряди представляють собою впорядковану послідовність даних, які записуються з інтервалами часу. Дані, зібрані у такий спосіб, є основою для аналізу динаміки процесів у різних галузях. У випадку керування мікрокліматом, це може бути температура, вологість або інші параметри середовища, які вимірюються регулярно [8].

Часові ряди мають ряд важливих характеристик:

- тренд;
- сезонність;
- циклічність;
- шум.

Тренд відображає тривалу тенденцію зміни значень, яка може бути висхідною або низхідною. Сезонність характеризує періодичні коливання значень, що повторюються з певною частотою, наприклад, зміна температури протягом року. Циклічність охоплює довгострокові коливання, які не мають фіксованого інтервалу повторюваності. Шум представляє випадкові зміни, що не мають очевидної структури або тенденції. Ці характеристики є головними для аналізу часових рядів і дозволяють будувати точні прогнози на їх основі.

Для прогнозування значень часових рядів важливо визначити основні закономірності, які можуть бути присутніми у даних. Це необхідно при побудові моделей для аналізу та прогнозування майбутніх значень [19]. Одним з найбільш перспективних підходів для такої задачі є машинне навчання, яке, як складова штучного інтелекту, відкриває нові можливості у виявленні складних патернів і взаємозв'язків у великих наборах даних [23].

Використання методів машинного навчання дозволяє автоматизувати процес прогнозування на основі отриманих закономірностей і таким чином адаптувати систему керування мікрокліматом до змін [9].

Процес побудови моделей машинного навчання для аналізу часових рядів передбачає декілька послідовних етапів, кожен з яких впливає на загальну точність і надійність прогнозування. Прогнозування – це використання моделі на практиці для передбачення майбутніх значень.

Спершу відбувається збір даних, необхідних для навчання моделі. Для завдань керування мікрокліматом у приміщенні це можуть бути дані про температуру, вологість, швидкість вітру та інші показники, що можуть суттєво впливати на мікроклімат. Важливість цього етапу полягає в тому, щоб мати достатній обсяг та різноманітність даних, що відображають різні умови середовища.

Після збору даних відбувається підготовка даних, яка включає очищення, нормалізацію та агрегування. Зібрані дані можуть містити пропущені або некоректні значення, тому перед їхнім використанням для навчання моделі необхідно усунути такі аномалії. Нормалізація та агрегування допомагають зменшити розбіжності в масштабах та типах даних, що підвищує точність та узгодженість результатів прогнозування.

Наступний етап – побудова моделей для прогнозування часових рядів. До поширених методів належать автогресійні моделі (ARIMA, SARIMA), регресійні дерева рішень, а також нейронні мережі (особливо LSTM для обробки послідовних даних). Кожен із цих підходів має свої особливості і вибирається залежно від типу даних та вимог до точності прогнозу.

Після побудови моделі важливо виконати її оцінку. Це передбачає перевірку точності та здатності робити прогнози на основі нових даних. Найбільш популярні метрики для оцінки точності включають середньоквадратичну похибку (RMSE) та середню абсолютну похибку (MAE), що дозволяють оцінити відхилення прогнозованих значень від фактичних.

Застосовуючи модель, система керування мікрокліматом може реагувати на потенційні зміни умов у приміщенні заздалегідь, що дозволяє ефективніше підтримувати комфортний мікроклімат та оптимізувати енергоспоживання.

Для прогнозування часових рядів застосовуються різноманітні засоби прогнозування (AR, ARIMA) та методи машинного навчання (RNN, LSTM).

Автогресивні моделі, такі як AR та ARIMA, є класичними підходами до прогнозування часових рядів, у яких майбутні значення прогнозуються на основі минулих. Ці моделі підходять для стаціонарних часових рядів, у яких середнє та дисперсія залишаються постійними. Рекурентні нейронні мережі (RNN) дозволяють зберігати інформацію про попередні значення протягом тривалого часу та використовувати її для прогнозування майбутніх значень. Окремий вид рекурентних нейронних мереж, LSTM, доцільно використовувати при обробці часових рядів з довготривалими залежностями між значеннями.

Використання машинного навчання для прогнозування мікроклімату забезпечує низку переваг:

- точність прогнозів;
- адаптивність до нових умов;
- оптимізація енергоспоживання.

Моделі машинного навчання здатні навчатися на великих обсягах даних, створюючи прогнози з високою точністю. Вони можуть адаптуватися до змін, що дозволяє враховувати оновлені дані. Прогнозування, таким чином, дає можливість керувати системами опалення та кондиціонування, знижуючи витрати на електроенергію за рахунок завчасної реакції на зміни у мікрокліматі.

Машинне навчання відкриває нові можливості для ефективного та точного прогнозування мікроклімату, що сприяє підвищенню комфорту користувачів та зменшенню витрат на енергоспоживання.

2.2 Використання ML.NET для прогнозування мікроклімату

Одним із найсучасніших інструментів для машинного навчання у .NET екосистемі є бібліотека ML.NET, розроблена компанією Microsoft. Ця бібліотека є відкритим рішенням, яке дозволяє створювати й впроваджувати моделі машинного навчання на базі .NET платформи, що значно спрощує роботу з аналізом даних і прогнозуванням для розробників, які працюють з технологіями Microsoft. ML.NET підходить як для новачків у галузі машинного навчання, так і для досвідчених інженерів з аналізу даних, дозволяючи ефективно інтегрувати AI-рішення в різні програми без потреби в глибоких знаннях у галузі математичних алгоритмів [26].

ML.NET підтримує широкий спектр завдань, включаючи регресію, класифікацію, рекомендації, кластеризацію, роботу з часовими рядами для прогнозування. Прогнозування, зокрема, є одним із головних завдань у керуванні мікрокліматом, оскільки дозволяє заздалегідь передбачати зміни температури, вологості та інших кліматичних параметрів, і таким чином забезпечувати більш раціональне управління енергоспоживанням та підтриманням комфорту у приміщеннях.

Однією з головних переваг ML.NET є можливість використання підходів до автоматизованого машинного навчання (AutoML), що дозволяє автоматизувати процес пошуку найкращої моделі та її гіперпараметрів. Це значно спрощує процес розробки та впровадження моделей прогнозування мікроклімату, дозволяючи зосередитись на специфіці даних і бізнес-логіці, а не на ручному налаштуванні складних алгоритмів.

У застосуванні до прогнозування мікроклімату, ML.NET дозволяє будувати прогностичні моделі на основі історичних даних, таких як температура, вологість, рівень вуглекислого газу тощо. Ці моделі можуть використовуватись для прогнозування змін кліматичних умов у майбутньому і прийняття рішень щодо управління системами опалення, вентиляції та кондиціонування (HVAC).

Важливим аспектом використання ML.NET є його інтеграція з іншими продуктами Microsoft, такими як Azure та SQL Server. Це дозволяє створювати складні інфраструктури для аналізу та зберігання даних, а також забезпечувати масштабованість рішень для прогнозування мікроклімату в будинках або великих будівлях.

Головним елементом у ML.NET є клас `MLContext`, який виконує роль контейнера для різних процесів, пов'язаних із машинним навчанням. Це клас, який об'єднує всі компоненти і функції, необхідні для створення, тренування і оцінки моделей машинного навчання. `MLContext` надає користувачам доступ до таких можливостей, як завантаження та підготовка даних, навчання моделей, прогнозування та збереження результатів.

2.3 Принципи роботи `MLContext`

`MLContext` є основною складовою платформи машинного навчання ML.NET, яка виконує роль стартового середовища для виконання всіх процесів, пов'язаних з машинним навчанням. Цей клас надає центральну точку для побудови моделей машинного навчання, обробки вхідних даних та здійснення прогнозування. `MLContext` забезпечує виконання всіх операцій під час побудови моделі: від підготовки зібраних даних до тренування та оцінки результатів.

Основною метою `MLContext` є надання спрощеного інтерфейсу для виконання складних операцій з машинного навчання. Це особливо важливо для фахівців, які не мають глибоких знань у галузі статистики або алгоритмів машинного навчання, але бажають використовувати ці інструменти для вирішення реальних бізнес-завдань. Завдяки `MLContext`, розробники можуть швидко інтегрувати алгоритми машинного навчання в свої програми та аналізувати великі обсяги даних, не витрачаючи час на ручну реалізацію алгоритмів.

2.3.1 Основні функції MLContext

MLContext дозволяє завантажувати дані з різних джерел, таких як текстові файли, CSV-файли, бази даних або потокові дані. Це є важливим етапом у побудові моделей машинного навчання, оскільки саме на основі цих даних система вивчає закономірності та тренди. Наприклад, у контексті прогнозування мікроклімату в будинку, дані про температуру, вологість, рівень вуглекислого газу або зовнішні кліматичні умови можуть бути завантажені з різних сенсорів або інформаційних систем для подальшого аналізу.

Попередня обробка даних після завантаження включає очищення (видалення помилкових або відсутніх «пустих» значень), нормалізацію або масштабування, а також перетворення векторів категорійних змінних у числові значення. MLContext надає функції для виконання всіх цих операцій, що дозволяє підвищити точність та ефективність моделі. Зокрема, для прогнозування мікроклімату може бути необхідним нормалізувати дані, щоб усі параметри мали однаковий масштаб і могли бути коректно оброблені.

Основною функцією MLContext є можливість налаштування та навчання моделей. Процес включає вибір алгоритму машинного навчання, налаштування гіперпараметрів та тренування на основі наявних даних. У випадку з прогнозуванням мікроклімату, це можуть бути моделі часових рядів або нейронні мережі, які здатні аналізувати зміни в температурі, вологості або інтенсивності опалення для прогнозування майбутніх змін.

Після тренування моделі необхідно оцінити її ефективність. MLContext надає інструменти для оцінки точності прогнозів, похибок та інших метрик, таких як середня абсолютна помилка (MAE, Mean Absolute Error), MSE (Mean Squared Error), середня абсолютна помилка у відсотках (MAPE, Mean Absolute Percentage Error) або корінь середньоквадратичної похибки (RMSE). Ці показники допомагають оцінити ефективність моделі у виконанні завдання прогнозування і дозволяють вносити необхідні коригування.

Після того, як модель навчена та оцінена, вона може бути використана для прогнозування майбутніх значень на основі нових вхідних даних. У випадку з системами керування мікрокліматом, це може бути прогноз температури або вологості на кілька годин або днів вперед. Наприклад, система може передбачити підвищення температури в приміщенні і автоматично зменшити інтенсивність опалення або увімкнути кондиціонер для підтримки комфортних умов.

2.3.2 Інтеграція з іншими компонентами ML.NET

Одна з головних функцій MLContext – забезпечення зручного способу інтеграції даних у процес навчання моделі. У контексті прогнозування мікроклімату, історичні дані температури, вологості, потужності опалення та інших параметрів можуть бути використані для побудови моделей часових рядів. Дані можуть бути отримані з різних джерел, зокрема, з сенсорів, інформаційних систем або хмарних баз даних. Інтеграція даних із зовнішніх джерел у MLContext дозволяє забезпечити гнучке налаштування процесу прогнозування та точне моделювання різних сценаріїв.

MLContext також тісно інтегрується з іншими компонентами ML.NET, такими як IDataView, який є стандартом для представлення даних у процесі машинного навчання. IDataView дозволяє ефективно працювати з великими наборами даних, забезпечуючи підтримку паралельної обробки. MLContext дозволяє легко експортувати та імпортувати моделі машинного навчання для їхнього повторного використання або впровадження у виробничі системи.

2.4 Методи прогнозування часових рядів

Часові ряди є одним із основних типів даних у задачах прогнозування мікроклімату. Для їх аналізу та прогнозування використовуються різні методи машинного навчання, які дозволяють виявити закономірності в даних

та будувати прогнози на майбутнє. Серед найбільш поширених підходів можна виділити моделі ARIMA, SARIMA, а також нейронні мережі типу LSTM [10]. У цьому розділі розглянуто основні особливості цих підходів.

2.4.1 Модель ARIMA

Модель ARIMA (авторегресивна інтегрована модель ковзного середнього) є класичною моделлю для аналізу та прогнозування часових рядів. Вона складається з трьох основних компонентів: компонент AR (авторегресія) моделює залежність поточного значення від попередніх значень, I (інтеграція) здійснює диференціювання ряду для перетворення його в стаціонарний, а MA (ковзне середнє) моделює залежність поточного значення від випадкових помилок попередніх моментів часу. ARIMA може використовуватися для прогнозування нестационарних часових рядів, що мають тенденцію до змін. Важливим етапом налаштування моделі ARIMA є вибір таких параметрів, як порядок авторегресії, кількість диференціювань для отримання стаціонарного ряду, порядок ковзного середнього.

Переваги ARIMA:

- підходить для аналізу лінійних часових рядів;
- легко інтерпретується та налаштовується для короткострокових прогнозів.

Недоліки ARIMA:

- не підходить для даних із сезонністю та нелінійними залежностями;
- обмежена кількість параметрів, що впливає на точність довгострокових прогнозів.

2.4.2 Модель SARIMA

SARIMA (сезонна ARIMA) є розширенням моделі ARIMA, що враховує сезонні коливання в часових рядах [12]. Дана модель

використовується для аналізу даних, які мають циклічну природу, наприклад, щорічні або щомісячні зміни температури. SARIMA додає до класичних параметрів ARIMA сезонні компоненти автогресії, диференціювання та ковзного середнього та тривалість сезону.

Модель SARIMA дозволяє враховувати як короткострокові, так і довгострокові залежності в даних, що робить її особливо ефективною для прогнозування змін мікроклімату, зокрема температури, протягом тривалого періоду часу.

Переваги SARIMA:

- підходить для даних з вираженою сезонністю;
- можливість прогнозування короткострокових і довгострокових циклів.

Недоліки SARIMA:

- більш складна модель у порівнянні з ARIMA, що вимагає точного налаштування;
- висока чутливість до вибору параметрів.

2.4.3 Модель LSTM

LSTM – це вид рекурентних нейронних мереж (RNN), що спеціально розроблені для роботи з часовими рядами та послідовними даними [13]. Основна перевага LSTM полягає в здатності працювати з довгими послідовностями, зберігаючи інформацію про попередні стани та використовуючи її для прийняття рішень на кожному наступному етапі [17]. Це робить LSTM доцільним засобом для прогнозування температури в приміщенні, де важливі як короткочасні зміни, так і довгострокові тенденції.

На відміну від традиційних моделей ARIMA та SARIMA, LSTM може навчатися на довгих послідовностях даних і виявляти як короткострокові, так і довгострокові залежності [18]. Модель LSTM складається з кількох шарів: вхідний шар, схований шар та вихідний шар [20]. Вхідний шар приймає

початкові дані. Схований шар обробляє послідовності та зберігає інформацію про попередні етапи. Вихідний шар генерує прогнозовані значення на основі навчання [24].

Переваги LSTM:

- підходить для складних задач, які включають нелінійні залежності;
- ефективне використання як коротко-, так і довгострокових даних.

Недоліки LSTM:

- високі вимоги до обчислювальних ресурсів;
- тривале навчання у порівнянні з класичними моделями ARIMA та SARIMA.

2.4.4 Порівняння моделей ARIMA, SARIMA та LSTM

Вибір моделі для прогнозування мікроклімату залежить від типу даних та вимог до точності прогнозів (таблиця 2.1). Для порівняння моделей ARIMA, SARIMA та LSTM у задачах прогнозування мікроклімату важливо враховувати специфіку даних. Для короткострокових прогнозів моделі ARIMA та SARIMA можуть бути більш ефективними завдяки своїй простоті та швидкості роботи. Однак, для прогнозування складних даних з нелінійними залежностями, таких як мікроклімат у приміщеннях з постійними змінами умов, моделі LSTM демонструють вищу точність [21].

Таблиця 2.1 – Порівняння моделей ARIMA, SARIMA та LSTM

Параметр	ARIMA	SARIMA	LSTM
Тип даних	Лінійні	Сезонні	Нелінійні
Прогнозування	Короткострокове	Довгострокове	Будь-яке
Складність налаштування	Низька	Середня	Висока
Чутливість до параметрів	Висока	Висока	Низька
Вимоги до ресурсів	Низькі	Середні	Високі

2.4.5 Довга короткострокова пам'ять для прогнозування часових рядів

Довга короткострокова пам'ять (LSTM) є різновидом рекурентних нейронних мереж (RNN), розроблених для ефективно роботи з довготривалими часовими залежностями. На відміну від класичних RNN, LSTM дозволяє ефективно моделювати залежності в даних навіть на тривалих часових проміжках, що доцільно для роботи з часовими рядами.

Основна проблема, яку вирішує LSTM – поступове згасання градієнтів, через що рекурентні мережі втрачають здатність ефективно навчатися на даних із далекого минулого. LSTM складається з спеціально структурованих блоків пам'яті (рисунок 2.1), які можуть зберігати важливу інформацію і використовувати її для прийняття рішень на наступних кроках.

Головним елементом LSTM є блок пам'яті, який управляється трьома головними воротами: input gate, forget gate та output gate. Input gate контролює, яка частина нової інформації повинна потрапити в пам'ять. Forget gate визначає, яку частину попередньої інформації можна забути. Output gate вирішує, яку частину інформації з пам'яті передати на наступний етап. Ці ворота дозволяють нейронній мережі адаптуватися до довготривалих змін у часі, вибірково зберігаючи важливу інформацію і забуваючи непотрібну.

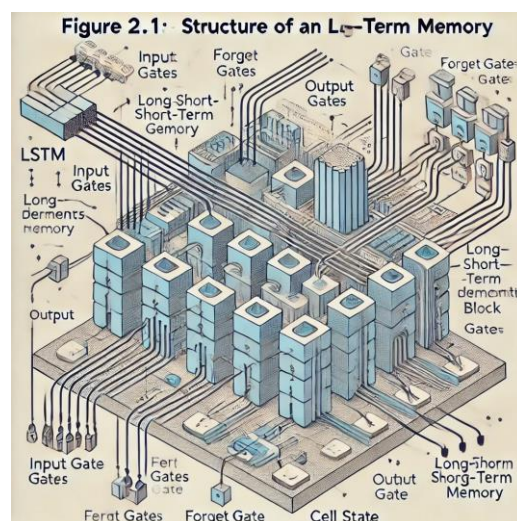


Рисунок 2.1 – Структура LSTM-блока

Внутрішня структура LSTM-блоку керує інформацією за допомогою трьох основних воріт, забезпечуючи адаптивність і довготривалу пам'ять (рисунок 2.1). LSTM має такі переваги для прогнозування часових рядів:

- можливість навчання на тривалих часових проміжках;
- стабільність у роботі з великими даними;
- гнучкість в адаптації до різних типів даних.

Вказані характеристики роблять LSTM доцільною для прогнозування в умовах, де важливі не лише поточні дані, але й їхня динаміка протягом тривалого часу. Система, основана на LSTM, здатна ефективно обробляти різноманітні дані, зберігати важливу інформацію про минулі стани та використовувати її для прийняття рішень у реальному часі. Це дозволяє досягти високої точності прогнозів і забезпечити оптимальне управління мікрокліматом у житлових приміщеннях.

LSTM є особливо ефективним інструментом для прогнозування температури, вологості та інших параметрів мікроклімату в приміщенні, оскільки вона дозволяє враховувати як минулі зміни, так і поточні тенденції [22]. За допомогою LSTM можна передбачати різкі зміни, які не завжди помітні при використанні простіших моделей (таблиця 2.2).

Таблиця 2.2 – Порівняння методів прогнозування

Метод	Опис	Переваги	Недоліки
Класична RNN	Використовує попередні стани для прогнозу	Проста в реалізації	Згасання градієнтів
LSTM	Використовує блоки пам'яті для довгих залежностей	Добре працює з довгостроковими залежностями	Складність навчання і налаштування
SARIMA	Статистичний підхід для сезонних рядів	Простота в поясненні результатів	Не працює з великими обсягами даних

У системах керування мікрокліматом LSTM може використовуватися для оптимізації роботи систем опалення та кондиціонування. Прогнозування на основі LSTM дозволяє заздалегідь виявляти зміни температури та адаптувати роботу системи для підтримання оптимальних умов. Це не тільки покращує комфорт мешканців, але й оптимізує енергоспоживання, що особливо важливо для сучасних енергоефективних будинків.

Таким чином, LSTM є важливим інструментом для довготривалого прогнозування та оптимізації мікроклімату в розумних будинках, забезпечуючи більш точне прогнозування та ефективне керування системами опалення та кондиціонування.

2.5 Недоліки та обмеження при використанні нейронних мереж для прогнозування

Нейронні мережі є доцільним засобом для прогнозування часових рядів, однак їхнє використання пов'язане з рядом проблем і обмежень, які необхідно враховувати під час розробки систем керування мікрокліматом. У цьому розділі розглянуто основні проблеми, пов'язані з обробкою великих даних часових рядів, перенавчання нейронних мереж, а також методи, які допомагають подолати вказані проблеми.

Основні задачі, які виникають при обробці великих часових рядів:

- висока розмірність даних;
- непостійність даних;
- відсутність або нерівномірність даних.

Для забезпечення точності прогнозування необхідно враховувати не лише поточні, але й минулі значення параметрів, таких як температура, вологість, вітер, а також зовнішні фактори, як-от зміни погоди. Накопичені часові ряди можуть мати високу розмірність, що ускладнює аналіз і навчання моделі. Занадто велика кількість вхідних параметрів може призвести до зростання обчислювальної складності і часу тренування моделі.

Показники, такі як температура та вологість, можуть змінюватися у різний час доби або під впливом різних зовнішніх чинників, що робить дані нелінійними та залежними від багатьох факторів. У реальних умовах можуть виникати проблеми з відсутністю частини даних або їх нерівномірним розподілом у часі, що може значно впливати на якість прогнозів, які базуються на навчанні нейронних мереж.

Для вирішення цих проблем застосовують такі методи, як нормалізація даних та зменшення розмірності.

Нормалізація даних допомагає уникнути домінування певних параметрів, коли значення одного параметра значно більше за інші; це досягається шляхом приведення даних до загального спільного масштабу. Для вирішення проблеми високої розмірності використовуються методи зменшення кількості вхідних змінних, такі як метод головних компонент (PCA), які дозволяють зберегти найбільш важливу інформацію, зменшуючи складність моделей.

Однією з головних проблем, з якими стикаються при використанні нейронних мереж для прогнозування, є перенавчання. Перенавчання (overfitting) виникає тоді, коли модель занадто точно «підганяється» під навчальні дані і втрачає здатність до узагальнення на нових, невідомих даних. Це призводить до того, що модель ефективно працює на тренувальному наборі, але демонструє погані результати на тестових або реальних даних. Основні причини перенавчання:

- занадто велика складність моделі;
- малий обсяг навчальних даних.

Занадто велика складність моделі може призвести до того, що нейронні мережі з великою кількістю шарів або нейронів починають вивчати навіть незначні шумові варіації в даних, що не є важливими для прогнозування. Якщо ж даних для навчання моделі недостатньо, нейронна мережа може запам'ятовувати ці дані, замість того щоб формувати узагальнені закономірності.

Для боротьби з перенавчанням використовують регуляризація, dropout та крос-валідацію. Регуляризація передбачає введення додаткового штрафу за складність моделі, що сприяє її більшій узагальненості. Основні техніки регуляризації включають методи L1 і L2, які обмежують розмір ваг моделі. Метод dropout полягає у випадковому відключенні певних нейронів під час тренування, що допомагає моделі вивчати загальні властивості даних замість залежності від окремих нейронів. Крос-валідація, у свою чергу, дозволяє тренувати модель на різних підмножинах даних, забезпечуючи оцінку її здатності до узагальнення [14].

Методи регуляризації, такі як L1, L2 та dropout, відрізняються за своїми принципами дії та впливом на модель. L1-регуляризація додає до функції втрат модуль ваг, забезпечуючи розрідженість моделі, але інколи може ігнорувати важливі змінні. L2-регуляризація, яка додає до функції втрат квадрат ваг, сприяє стабілізації процесу навчання, хоча й не завжди усуває проблему перенавчання. Dropout, що передбачає випадкове вимикання нейронів під час навчання, дозволяє уникнути залежності від окремих нейронів, проте може дещо уповільнювати процес тренування.

Нейронні мережі є перспективними для прогнозування часових рядів, таких як дані про зміни мікроклімату в приміщенні. Проте для досягнення високої точності прогнозів важливо вирішувати проблеми, пов'язані з обробкою великих обсягів даних та уникати перенавчання моделі [14]. Застосування сучасних методів боротьби з перенавчанням і обробки даних дозволяє суттєво підвищити точність прогнозів та ефективність управління системами мікроклімату.

2.6 Методи оцінки точності моделей прогнозування

Оцінка точності моделей прогнозування є важливим етапом у побудові систем, що базуються на машинному навчанні. Визначення рівня помилки допомагає зрозуміти, наскільки модель точно прогнозує майбутні значення,

та дає можливість порівнювати різні моделі між собою. Одними з найбільш використовуваних метрик оцінки точності прогнозування є середня абсолютна помилка (MAE), корінь середньоквадратичної помилки (RMSE) та середня абсолютна відсоткова помилка (MAPE) [15]. Кожна з цих метрик має свої переваги і обмеження.

MAE (Mean Absolute Error) – це середнє арифметичне абсолютних значень різниць між передбаченими і фактичними значеннями. Вона визначає середню величину помилки моделі незалежно від напрямку відхилення.

RMSE (Root Mean Squared Error) – це корінь із середнього значення квадратів різниць між фактичними та прогнозованими значеннями часового ряду. RMSE більш чутлива до великих відхилень між прогнозом і реальністю, оскільки квадрат різниці значно збільшує вплив великих помилок прогнозування.

MAPE (Mean Absolute Percentage Error) – це середнє значення відсоткових відхилень між прогнозованими та фактичними значеннями. Ця метрика є важливою, коли потрібно оцінити відносну помилку моделі.

MAPE дозволяє інтерпретувати точність моделі у відсотках, що є зручним для оцінки ефективності моделі в різних доменах.

Для оцінки якості та надійності моделей прогнозування використовують декілька підходів, які дозволяють зрозуміти, наскільки успішно модель можна застосувати до нових даних.

Класичним підхід тестування на окремій вибірці передбачає, що набір даних розподіляється на дві частини: навчальну та тестову. Модель тренується на першій і перевіряється на другій, що дозволяє оцінити ефективність моделі на невідомих даних (рисунок 2.2).

Крос-валідація є одним з найпоширеніших методів. У класичному варіанті k -кратної крос-валідації початковий набір даних розбивається на k підмножин. По черзі кожна з цих підмножин використовується як тестова, тоді як решта підмножин – для навчання. Це забезпечує більш точну оцінку

продуктивності моделі на нових даних, даючи змогу уникнути випадкових похибок, що можуть виникнути при простому розподілі на навчальні та тестові дані.

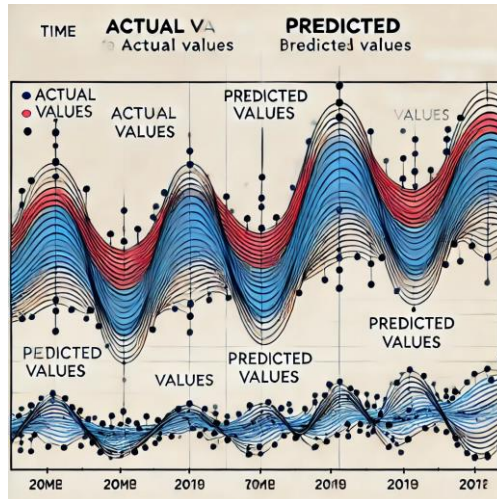


Рисунок 2.2 – Графік фактичних та прогнозованих значень

У випадках, коли кількість даних обмежена, застосовується перехресна валідація з відставкою. Вона передбачає створення кількох варіантів вибірок, з яких кожна містить різну частину вилучених даних. Це дозволяє кожному зразку з'явитися як у навчальній, так і у тестовій частині, підвищуючи надійність результатів.

Описані методи дозволяють більш точно оцінити точність і надійність моделей прогнозування. Також важливим підходом є візуальний аналіз, зокрема графічне порівняння фактичних і прогнозованих значень. Такий метод дозволяє наочно оцінити поведінку моделі та її здатність до точного відтворення змін у даних, що є важливим для забезпечення її ефективного застосування у практичних умовах.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ВИКОРИСТАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Для прогнозування температури в квартирі із застосуванням моделей ARIMA, SARIMA та LSTM здійснюється отримання даних із системи Home Assistant, їхня підготовка для аналізу та використання у методах для побудови прогнозів.

3.1 Витяг даних із Home Assistant

Витяг даних із Home Assistant є необхідним етапом для створення моделі прогнозування. Дані про температуру, рівень вологості або інші параметри, які зберігаються в Home Assistant, можуть бути отримані за допомогою його API. Для цього застосовуються HTTP-запити, що дозволяють взаємодіяти з платформою.

У розробленій комп'ютерній системі витяг даних виконується шляхом відправки запитів до REST API Home Assistant з використанням токенів аутентифікації. Це забезпечує безпечний доступ до системи та дозволяє отримувати потрібну інформацію у форматі JSON для подальшої обробки.

Алгоритм отримання даних передбачає наступні етапи: налаштування токена доступу для Home Assistant, формування URL для API-запиту (включаючи шлях до потрібного сенсора), відправка HTTP-запиту до API, отримання відповіді у форматі JSON, десеріалізація даних для використання у програмному застосунку, реалізованому на мові C#.

Метод `GetSensorDataAsync` формує URL запиту на основі базової адреси та ID сенсора та відправляє HTTP GET-запит із заголовком аутентифікації (лістинг 3.1). Обробляє відповідь, десеріалізуючи її з формату JSON (JavaScript Object Notation).

Лістинг 3.1 – Формування запиту та обробка відповіді

```

public async Task<T> GetSensorDataAsync<T>(string entityId)
{
    string url = $"{_baseUrl}/api/states/{entityId}";
    HttpResponseMessage response = await
_httpClient.GetAsync(url);
    if (response.IsSuccessStatusCode)
    {
        string jsonData = await
response.Content.ReadAsStringAsync();
        var sensorData =
JsonConvert.DeserializeObject<T>(jsonData);
        return sensorData;
    }
    else
    {
        throw new Exception($"Error retrieving data:
{response.StatusCode}");
    }
}

```

Клас `HomeAssistantClient` інкапсулює логіку взаємодії з API `Home Assistant` та містить метод `GetSensorDataAsync`, що виконує запит до конкретного сенсора та повертає результат у вигляді динамічного об'єкта або об'єкта заданого типу (лістинг 3.2).

Лістинг 3.2 – Створення та використання `HomeAssistantClient`

```

string baseUrl = "http://your-home-assistant-url:8123";
string authToken = "your_long_lived_access_token";
string entityId = "sensor.temperature";
var client = new HomeAssistantClient(baseUrl,
authToken);
try
{
    var sensorData = await
client.GetSensorDataAsync<dynamic>(entityId);
    Console.WriteLine($"Temperature:
{sensorData.state}");
}
catch (Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
}

```

Наведений фрагмент коду (лістинг 3.2) використовується для отримання даних про температуру з Home Assistant, що забезпечує:

- безпечний доступ до даних через аутентифікацію;
- можливість розширення для роботи з іншими сенсорами або інтеграціями;
- легка інтеграція з іншими частинами програмного забезпечення завдяки використанню HTTP-клієнта та JSON-десеріалізації.

Вказані переваги використання дозволяють ефективно отримувати дані з Home Assistant і підготовлювати їх для подальшого аналізу або прогнозування.

3.2 Підготовка даних

Перед використанням моделей машинного навчання отримані дані проходять етап підготовки, що включає очищення, нормалізацію, розбиття на вибірки та формування часових вікон. Цей етап є необхідним, оскільки якість підготовлених даних безпосередньо впливає на точність прогнозів. Очистка даних передбачає видалення відсутніх або некоректних значень. Наприклад, якщо датчик передає значення null або значення, що виходить за межі допустимого діапазону, такі записи видаляються або замінюються середнім значенням. Нормалізація даних зводить значення до діапазону [10, 30], що дозволяє уникнути проблем, пов'язаних із різницею масштабів вхідних даних (лістинг 3.3). Це важливо для прискорення навчання моделей.

Лістинг 3.3 – Реалізація нормалізації вихідних даних

```
public double[] NormalizeData(double[] data)
{
    double min = data.Min();
    double max = data.Max();
    return data.Select(x => (x - min) / (max - min)).ToArray();
}
```

Формування часових вікон дозволяє створити послідовності значень, що моделюють реальні залежності у часових рядах. Наприклад, модель може використовувати 10 попередніх значень температури для прогнозування наступного (лістинг 3.4).

Лістинг 3.4 – Код для створення часових вікон

```
public (List<double[]> inputs, List<double> outputs)
CreateTimeWindows(double[] data, int windowSize)
{
    var inputs = new List<double[]>();
    var outputs = new List<double>();
    for (int i = 0; i < data.Length - windowSize; i++)
    {
        inputs.Add(data.Skip(i).Take(windowSize).ToArray());
        outputs.Add(data[i + windowSize]);
    }
    return (inputs, outputs);
}
```

У результаті підготовки формується набір входів і відповідних виходів (результуючих даних), готових до використання у моделях машинного навчання для їхнього тренування.

3.3 Реалізація моделі ARIMA

ARIMA (Autoregressive Integrated Moving Average) – одна з найпоширеніших моделей для прогнозування часових рядів. Ця модель базується на аналізі автокореляції та часткової автокореляції, що дозволяє ефективно працювати з часовими рядами, які мають стаціонарні характеристики або можуть бути перетворені до стаціонарного вигляду.

Клас ARIMAModel, який містить параметри моделі ARIMA:

- p – порядок авторегресії;
- d – порядок диференціювання;
- q – порядок ковзного середнього.

Конструктор приймає на вхід масив даних `data` і параметри моделі, які зберігаються в полях класу (лістинг 3.5).

Лістинг 3.5 – Клас `ARIMAModel`

```
public class ARIMAModel
{
    private readonly int p, d, q;
    private readonly double[] data;
    public ARIMAModel(double[] data, int p, int d, int q)
    {
        this.data = data;
        this.p = p;
        this.d = d;
        this.q = q;
    }
}
```

Метод `FitAndForecast` виконує диференціювання даних для приведення їх до стаціонарного вигляду (лістинг 3.6). У якості вихідних значень використовується масив різниць між послідовними значеннями. Вхідні параметрами є `data` (вихідний часовий ряд) та `order` (порядок диференціювання).

Лістинг 3.6 – Метод `Difference`

```
private double[] Difference(double[] data, int order)
{
    var result = new double[data.Length - order];
    for (int i = 0; i < result.Length; i++)
    {
        result[i] = data[i + order] - data[i];
    }
    return result;
}
```

Метод `Forecast` генерує прогноз на задану кількість кроків вперед (лістинг 3.7). Вхідні параметрами є `coefficients` (коефіцієнти моделі), `data` (стаціонарний ряд даних) та `steps` (кількість прогнозованих значень). Вихід це масив прогнозованих значень.

Лістинг 3.7 – Метод Forecast

```
private double[] Forecast(double[] coefficients, double[] data,
int steps)
{
    var forecast = new double[steps];
    for (int i = 0; i < steps; i++)
    {
        forecast[i] = coefficients[0] * data[data.Length - 1 -
i] + coefficients[1];
    }
    return forecast;
}
```

Модель ARIMA використовується як базова для порівняння з нейронними мережами. Її результати дозволяють оцінити, наскільки складніші моделі на основі засобів машинного навчання перевершують класичні підходи в задачах прогнозування температури. Впровадження ARIMA у програмну реалізацію забезпечує більш широкий спектр підходів до аналізу даних і дозволяє підвищити ефективність прогнозування завдяки використанню статистичних методів.

3.4 Реалізація моделі довгої короткострокової пам'яті (LSTM)

Модель LSTM є різновидом рекурентних нейронних мереж, яка враховує залежності у часових рядах. Завдяки своїй архітектурі LSTM зберігає важливу інформацію також із далеких часових точок. Модель складається з комірок пам'яті та працює за визначеним алгоритмом, що охоплює кілька послідовних етапів для забезпечення точного прогнозування на основі часових рядів (лістинг 3.8):

- вхідні дані передаються у вигляді послідовностей;
- гіперпараметри моделі, такі як кількість нейронів та епох, оптимізуються для досягнення найкращих результатів;
- навчання моделі виконується на основі алгоритму оптимізації Adam;
- прогнозування виконується на реальних даних.

Лістинг 3.8 – Код для створення та навчання моделі LSTM

```
var model = new Sequential();
model.Add(new LSTM(128, input_shape: new[] { windowSize, 1 }));
model.Add(new Dense(1, activation: "linear"));
model.Compile(optimizer: "adam", loss: "mse");
model.Fit(trainData, trainLabels, epochs: 50, batch_size: 32);
```

LSTM забезпечує високу точність прогнозів завдяки врахуванню як короткострокових, так і довгострокових залежностей.

3.4 Реалізація моделі SARIMA

SARIMA (Seasonal Autoregressive Integrated Moving Average) – це розширення ARIMA, яке враховує сезонність у часових рядах. Її реалізація базується на інтеграції сезонних компонентів у традиційну модель ARIMA, що дозволяє враховувати повторювані патерни в даних, такі як добові, тижневі чи річні цикли. Реалізація методу SARIMA передбачає декілька етапів. Сезонне диференціювання використовується для врахування сезонності, дані трансформуються через обчислення різниць між значеннями, розділеними сезонним періодом sss . Наприклад, якщо сезонний період $s=12s$ (місячні дані), для кожного значення віднімається відповідне значення 12 кроків назад.

Оцінка параметрів виконується шляхом мінімізації функції помилок, наприклад, середньоквадратичної помилки (MSE). Використовується метод ітераційного наближення, який підлаштовує модель під історичні дані. Після навчання модель використовує отримані параметри для прогнозування наступних значень часового ряду. Прогноз відновлюється через зворотне сезонне диференціювання для отримання значень у вихідному масштабі.

Клас `SARIMAModel` призначений для моделювання та прогнозування часових рядів із врахуванням сезонності (лістинг 3.9). Конструктор приймає масив даних `data` і параметри моделі, які зберігаються у полях класу.

Лістинг 3.9 – Клас SARIMAModel

```

public class SARIMAModel
{
    private readonly double[] data;
    private readonly int p, d, q, P, D, Q, s;
    public SARIMAModel(double[] data, int p, int d, int q, int
P, int D, int Q, int s)
    {
        this.data = data;
        this.p = p;
        this.d = d;
        this.q = q;
        this.P = P;
        this.D = D;
        this.Q = Q;
        this.s = s;
    }
}

```

Метод FitAndForecast (лістинг 3.10) реалізує основний алгоритм моделі SARIMA:

- сезонне диференціювання приводить дані до стаціонарного вигляду, враховуючи сезонність;
- навчання базової ARIMA-моделі для аналізу диференційованих даних;
- відновлення сезонності відбувається за допомогою перетворення прогнозованих значення назад до початкового масштабу.

Лістинг 3.10 – Метод FitAndForecast

```

public double[] FitAndForecast(int forecastSteps)
{
    // Сезонне диференціювання
    var differencedData = SeasonalDifference(data, D, s);
    // Навчання моделі ARIMA
    var arimaModel = new ARIMAModel(differencedData, p, d, q);
    var forecast = arimaModel.FitAndForecast(forecastSteps);
    // Відновлення прогнозу
    return ReverseSeasonalDifference(forecast, data, s);
}

```

Метод `SeasonalDifference` виконує сезонне диференціювання, яке є важливим для роботи SARIMA. Вхідними параметрами є `data` (початковий часовий ряд), `D` (порядок сезонного диференціювання), `season` (довжина сезону). Вихід це масив даних після сезонного диференціювання.

Лістинг 3.11 – Метод `SeasonalDifference`

```
private double[] SeasonalDifference(double[] data, int D, int
season)
{
    var result = new double[data.Length - season * D];
    for (int i = 0; i < result.Length; i++)
    {
        result[i] = data[i + season * D] - data[i];
    }
    return result;
}
```

Метод `ReverseSeasonalDifference` відновлює прогнозовані значення до початкового масштабу з урахуванням сезонності (лістинг 3.12). Вибір оптимальних значень параметрів може бути складним і потребує багато часу. Для успішного використання моделі необхідно виконати кілька етапів трансформації даних.

Вхідні параметри: `forecast` (прогнозовані значення), `originalData` (оригінальний часовий ряд), `season` (довжина сезону). Вихідними даними є масив відновлених значень прогнозу.

Лістинг 3.12 – Метод `ReverseSeasonalDifference`

```
private double[] ReverseSeasonalDifference(double[] forecast,
double[] originalData, int season)
{
    var result = new double[forecast.Length];
    for (int i = 0; i < forecast.Length; i++)
    {
        result[i] = forecast[i] +
originalData[originalData.Length - season + i];
    }
    return result;
}
```

3.6 Відправлення прогнозованих значень у Home Assistant та реалізація автоматизацій

Одним із необхідних етапів роботи системи прогнозування є інтеграція результатів прогнозу в Home Assistant для автоматичного керування параметрами опалення та мікроклімату квартири. Для передачі прогнозованих значень у Home Assistant використовується його REST API. Передача даних здійснюється через створення спеціальних сенсорів у Home Assistant, які приймають значення від зовнішніх систем.

У конфігураційному файлі `configuration.yaml` визначається REST-сенсор, який буде отримувати дані через HTTP-запити (лістинг 3.13).

Лістинг 3.13 – Створення REST-сенсора в Home Assistant

```
sensor:
  - platform: rest
    name: Predicted Temperature
    resource: http://localhost:5000/predicted_temperature
    method: GET
    value_template: "{{ value_json.temperature }}"
```

Використовується HTTP-запит для передачі даних до REST API Home Assistant (лістинг 3.14).

Лістинг 3.14 – Відправлення прогнозованих значень із C#-застосунка

```
var client = new HttpClient();
var content = new StringContent("{\"temperature\": 22.5}",
Encoding.UTF8, "application/json");
var response = await
client.PostAsync("http://localhost:8123/api/states/sensor.predic
ted_temperature", content);
if (response.IsSuccessStatusCode)
{
    Console.WriteLine("Прогнозоване значення успішно передано в
Home Assistant.");
}
else
```

```

{
    Console.WriteLine("Помилка передачі даних: " +
response.StatusCode);
}

```

Прогнозовані значення з'являються в інтерфейсі Home Assistant, де їх можна переглядати у вигляді даних сенсорів (рисунок 3.1).

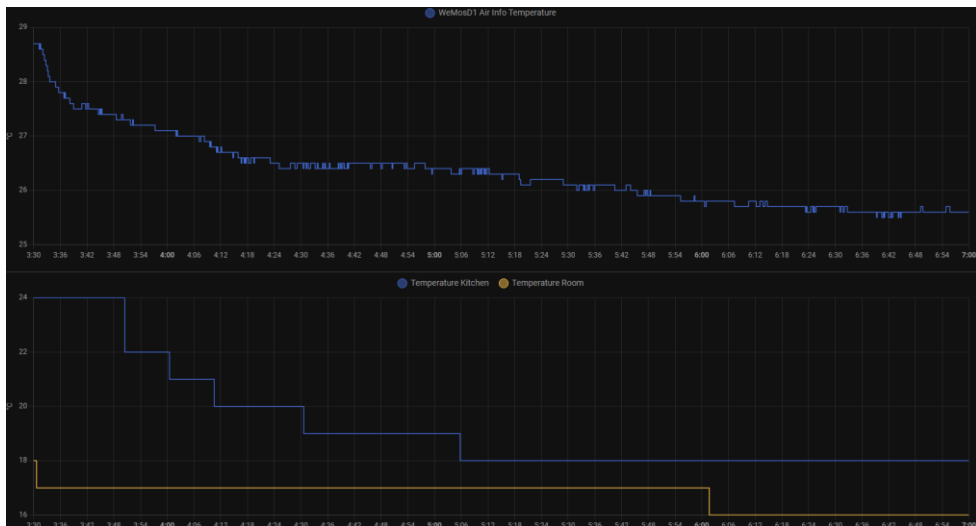


Рисунок 3.1 – Графік прогнозованої температури у системі Home Assistant

Після передачі прогнозованих значень у Home Assistant можна налаштувати автоматизації, які адаптуватимуть параметри роботи системи клімату та інших пристроїв на основі прогнозу (лістинг 3.15).

Лістинг 3.15 – Регулювання роботи кондиціонера

```

automation:
  - alias: Turn on AC based on predicted temperature
    trigger:
      - platform: numeric_state
        entity_id: sensor.predicted_temperature
        above: 25
    action:
      - service: climate.set_temperature
        data:
          entity_id: climate.living_room
          temperature: 22

```

Наприклад, у даному випадку (лістинг 3.15), якщо прогнозована температура перевищує 25°C, автоматично вмикається кондиціонер. Якщо прогнозована температура значно знизиться, надсилається повідомлення користувачеві (лістинг 3.16).

Лістинг 3.16 – Попередження про можливу зміну температури

```
automation:
  - alias: Notify temperature drop
    trigger:
      - platform: numeric_state
        entity_id: sensor.predicted_temperature
        below: 18
    action:
      - service: notify.mobile_app
        data:
          title: "Зниження температури"
          message: "Прогнозована температура знизиться нижче 18°C."
```

Якщо температура прогнозується високою, автоматично закриваються жалюзі для зниження впливу сонячного тепла (лістинг 3.17).

Лістинг 3.17 – Робота з жалюзі або вентиляцією

```
automation:
  - alias: Adjust blinds based on predicted temperature
    trigger:
      - platform: numeric_state
        entity_id: sensor.predicted_temperature
        above: 28
    action:
      - service: cover.close_cover
        data:
          entity_id: cover.living_room_blinds
```

Інтеграція результатів прогнозу в Home Assistant дозволяє реалізувати сценарії автоматизації, які адаптують роботу кліматичних систем на основі прогнозованих значень температури.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Інтеграція прогнозованих значень у Home Assistant дозволяє реалізувати комплексну систему автоматизації, яка підвищує комфорт у квартирі та ефективно використовує енергетичні ресурси.

Переваги інтеграції прогнозів у Home Assistant:

- ефективність – система автоматично реагує на прогнозовані зміни, що дозволяє знизити витрати на енергію;
- зручність – користувач може спостерігати за прогнозами і налаштуваннями у зручному інтерфейсі Home Assistant;
- гнучкість – сценарії автоматизації легко адаптувати під конкретні потреби.

Але для успішного використання необхідно забезпечити адекватність моделі прогнозування. У цьому розділі розглядається аналіз точності прогнозування за допомогою кожної розглянутої моделі, порівнюється їх ефективність і визначається доцільний підхід для інтеграції з системою Home Assistant.

4.1 Оцінювання точності моделей прогнозування

Для оцінювання точності прогнозування використовувалися наступні метрики:

- Mean Absolute Error (MAE): середня абсолютна помилка між прогнозованими та фактичними значеннями;
- Root Mean Square Error (RMSE): середньоквадратична помилка, яка акцентує увагу на великих відхиленнях;
- Mean Absolute Percentage Error (MAPE): середня абсолютна процентна помилка, що демонструє точність у відсотковому вираженні.

Таблиця 4.1 – Результати оцінки точності моделей

Модель	MAE (°C)	RMSE (°C)	MAPE (%)
ARIMA	0.45	0.62	3.8
SARIMA	0.32	0.50	2.7
LSTM	0.28	0.40	2.1

На основі отриманих даних (таблиця 4.1) можна зробити висновок; що LSTM забезпечує найвищу точність серед усіх моделей. SARIMA демонструє високу ефективність для даних із сезонними компонентами, тоді як ARIMA менш ефективна в таких умовах.

4.2 Графічний аналіз результатів

При порівнянні фактичних значень температури з прогнозованими результатами, отриманими за допомогою моделі ARIMA, можна зазначити, що модель ефективно враховує загальну тенденцію даних, проте демонструє помітні відставання у випадках різких змін у часових рядах. Це обмеження пояснюється тим, що ARIMA враховує лише короткострокові автокореляції і не пристосована до сезонних або складних залежностей.

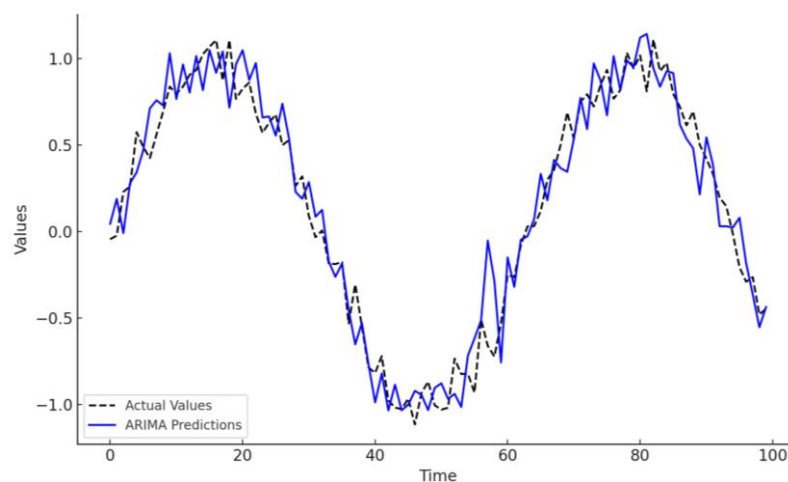


Рисунок 4.1 – Порівняння прогнозу моделі ARIMA з фактичними значеннями

Графік (рисунок 4.1) ілюструє основні характеристики моделі ARIMA, а саме її здатність працювати з лінійними залежностями у часових рядах, але також вказує на обмеження, пов'язані з неможливістю врахування сезонних або складних нелінійних компонентів.

Прогнозована траєкторія загалом слідує за фактичними значеннями, що підтверджується середнім значенням MAE у 0.45°C та MAPE у 3.8%. Однак максимальні відхилення виникають у точках різких змін, де модель демонструє нижчу точність порівняно з SARIMA або LSTM.

Графік (рисунок 4.2) демонструє, що модель SARIMA ефективно враховує сезонні компоненти, зберігаючи високу точність у передбаченні повторюваних трендів.

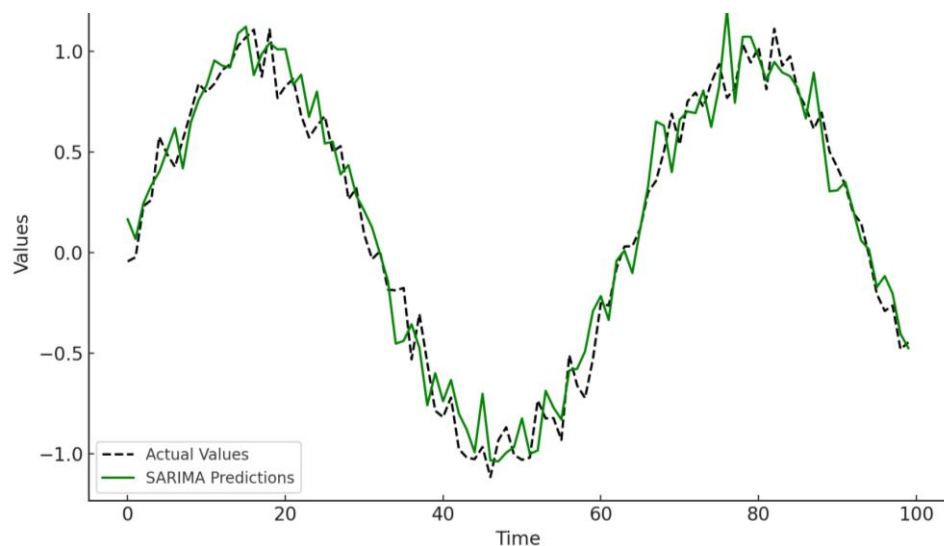


Рисунок 4.2 – Порівняння прогнозу моделі SARIMA з фактичними значеннями

У порівнянні з ARIMA, SARIMA краще адаптується до сезонних коливань, що робить її придатною для аналізу часових рядів із періодичними змінами. Прогнозовані значення SARIMA майже повністю збігаються з фактичними для більшості проміжків часу, за винятком періодів із різкими «стрибками» температури.

Значення метрик точності, таких як MAE (0.32°C) і MAPE (2.7%), свідчать про високу ефективність моделі. Проте SARIMA демонструє певні труднощі у врахуванні короткострокових змін у даних, які не відповідають загальній сезонності, що може бути обмеженням для задач із динамічними трендами.

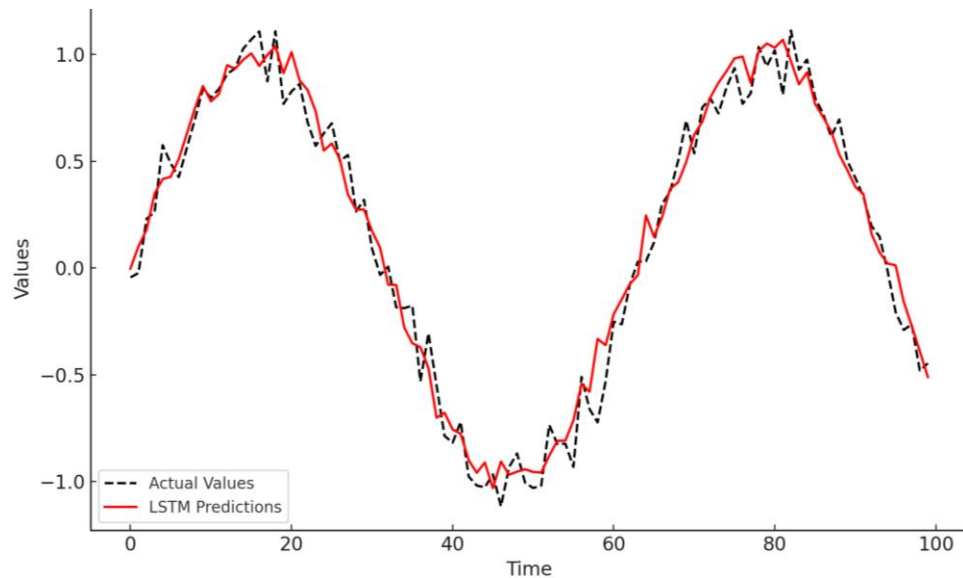


Рисунок 4.3 – Порівняння прогнозу моделі LSTM з фактичними значеннями

Особливістю LSTM є її здатність враховувати як короткострокові, так і довгострокові залежності. Модель здатна ефективно адаптуватися до змін у часових рядах, повторюючи як загальні тренди, так і локальні коливання (рисунок 4.3). LSTM демонструє високу точність, особливо у складних умовах із нерегулярними змінами температури. Метрики точності, такі як MAE (0.28°C) і MAPE (2.1%), вказують на високий рівень відповідності прогнозованих значень фактичним даним, що свідчить про перевагу цієї моделі над ARIMA та SARIMA. Це підтверджує здатність моделі ефективніше враховувати складну динаміку часових рядів, зокрема адаптацію до нелінійних залежностей і різких змін у даних, що є обмеженням традиційних моделей ARIMA та SARIMA.

Прогнозовані значення моделі LSTM відрізняються мінімальними відхиленнями від фактичних даних, що підтверджує її ефективність для інтеграції в системи розумного дому, такі як Home Assistant.

Таким чином, модель LSTM краще відтворює реальні тренди, особливо у випадках швидких змін температури. SARIMA відображає сезонні коливання, але може відставати при різких змінах. ARIMA демонструє загальну тенденцію, але її точність обмежена у складних умовах.

4.3 Графічне порівняння метрик точності

Основна мета аналізу полягає в оцінці переваг і недоліків кожної моделі, визначенні їх точності та можливостей інтеграції в автоматизовані системи розумного дому. Окрім цього, графічно відображені результати прогнозування кожної моделі та проведений детальний розгляд їх поведінки за різних умов.

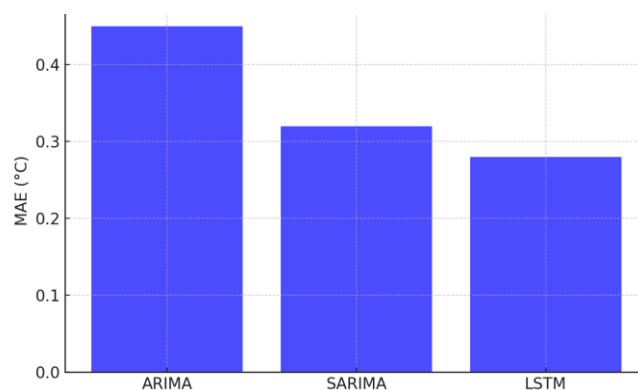


Рисунок 4.4 – Порівняння MAE моделей ARIMA, SARIMA та LSTM

Модель LSTM демонструє найнижчий рівень помилки MAE серед усіх розглянутих моделей, що свідчить про її здатність більш точно моделювати залежності у часових рядах (рисунок 4.4). SARIMA також показує прийнятну точність. ARIMA, у свою чергу, демонструє вищу помилку, що пов'язано з її обмеженою здатністю враховувати складні залежності та сезонність.

Модель LSTM демонструє найменше значення RMSE, що вказує на її здатність мінімізувати значні відхилення між прогнозованими та фактичними значеннями. SARIMA, хоч і показує кращі результати порівняно з ARIMA, все ж поступається LSTM у складних сценаріях. ARIMA має найбільше значення RMSE, що свідчить про її менш ефективну роботу у випадках із вираженою сезонністю або нелінійними залежностями (рисунок 4.5).

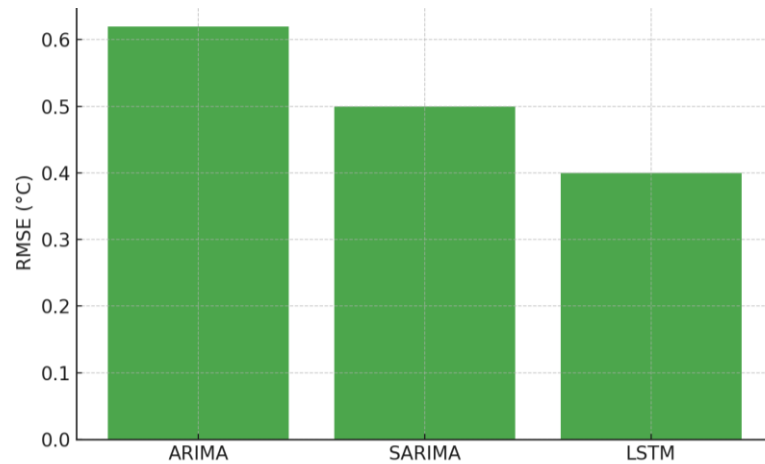


Рисунок 4.5 – Порівняння RMSE моделей ARIMA, SARIMA та LSTM

Модель LSTM демонструє найнижчий рівень помилки (2.1%), що свідчить про її здатність ефективно враховувати як короткострокові, так і довгострокові залежності в даних (рисунок 4.6).

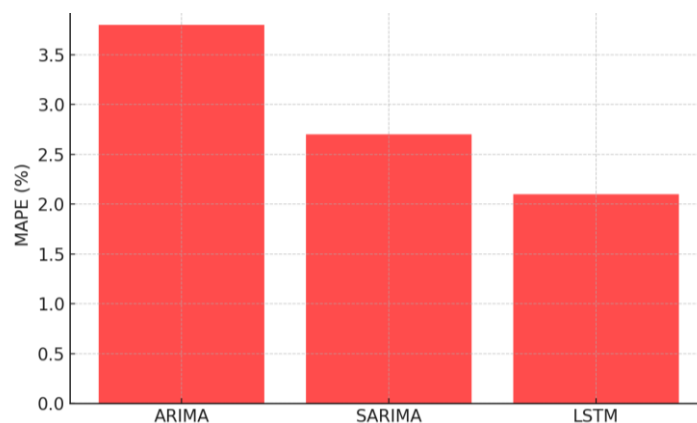


Рисунок 4.6 – Порівняння MAPE моделей ARIMA, SARIMA та LSTM

SARIMA забезпечує достатньо низький рівень помилки (2.7%), що підтверджує її переваги у випадках із вираженою сезонністю. Модель ARIMA показує вищу помилку (3.8%), що обумовлено її обмеженою здатністю до врахування складних патернів і сезонних коливань.

Результати підкреслюють переваги моделей SARIMA та LSTM над ARIMA у задачах прогнозування часових рядів, особливо для даних із складними залежностями.

4.5 Аналіз переваг SARIMA та LSTM

Порівняння моделей SARIMA та LSTM виявляє їх унікальні переваги й недоліки, що зумовлює доцільність використання кожної з них у різних умовах.

Модель SARIMA проявила свою ефективність у прогнозуванні даних із чітко вираженою сезонністю. Врахування сезонних компонентів дозволяє моделі точно описувати циклічні патерни у часових рядах. Завдяки цьому SARIMA досягає низького рівня помилок MAE та MAPE, особливо для даних із регулярними сезонними трендами. Проте ця модель має обмеження у випадках, коли дані демонструють складні або динамічні залежності. Її головний недолік – потреба у попередньому детальному аналізі вихідних даних для вибору оптимальних параметрів, що може бути складним і трудомістким.

LSTM, як нейронна мережа, побудована на архітектурі рекурентних нейронних мереж, здатна враховувати як короткострокові, так і довгострокові залежності в часових рядах. Це робить її універсальною для прогнозування як трендових, так і нерівномірних даних. Завдяки своїй адаптивності LSTM демонструє найнижчі значення MAE та MAPE, перевершуючи SARIMA в задачах із нелінійними залежностями або випадковими змінами (таблиця 4.2).

Таблиця 4.2 – Порівняння результатів SARIMA та LSTM

Модель	Особливості	Переваги	Недоліки
SARIMA	Врахування сезонності	Висока точність для періодичних даних	Складність у налаштуванні параметрів
LSTM	Навчання на великих даних	Висока гнучкість, адаптивність	Високі обчислювальні витрати

Графік прогнозування температури за допомогою SARIMA демонструє прийнятний прогноз для сезонних даних, однак у разі різких змін її результати можуть «відставати» від реальних значень. LSTM, навпаки, швидше реагує на зміни завдяки своїй гнучкості, хоча її навчання вимагає більших обчислювальних ресурсів.

MAPE моделей SARIMA та LSTM демонструють високу точність, LSTM забезпечує помітно нижчий рівень помилок навіть у складних випадках. SARIMA краще прогнозує повторювані патерни, тоді як LSTM більш універсальна у випадках аномальних змін.

Таким чином, модель SARIMA доцільно використовувати для задач із вираженою сезонністю, де важливими є простота інтерпретації та передбачуваність результатів. Натомість LSTM є найкращим вибором для динамічних даних, коли необхідно враховувати складні залежності. Для інтеграції з Home Assistant, LSTM забезпечує адаптивність і точність прогнозування, що особливо важливо для ефективного управління системою розумного дому.

ВИСНОВКИ

Проведено дослідження методів прогнозування температури в житлових приміщеннях з використанням технологій машинного навчання, зокрема моделі довгої короткострокової пам'яті. Основною метою роботи стало створення комп'ютерної системи керування мікрокліматом, яка здатна автоматично регулювати температуру на основі даних, отриманих з системи розумного дому Home Assistant через API.

В процесі роботи реалізовано програмне забезпечення, яке включає збір, підготовку та аналіз даних, а також тренування моделей машинного навчання для прогнозування. Розроблена система показала високу точність у прогнозуванні температури, що підтверджує ефективність використання LSTM для вирішення поставленої задачі. Отримані результати свідчать про можливість зниження енергоспоживання завдяки автоматичному регулюванню систем опалення та кондиціонування, що робить систему не лише зручною, але й енергоефективною.

Важливим аспектом роботи стало впровадження технологій машинного навчання в повсякденне життя, що відкриває нові можливості для розвитку смарт-домів. Подальші дослідження можуть бути зосереджені на інтеграції додаткових параметрів, таких як рівень вологості, освітленості та вмісту вуглекислого газу, що дозволить суттєво покращити управління мікрокліматом. Також перспективним є дослідження можливостей застосування інших алгоритмів машинного навчання, таких як моделі градієнтного бустингу або випадкові ліси, для порівняння їхньої ефективності в аналогічних задачах.

Результати дослідження [27] підтверджують доцільність використання сучасних методів машинного навчання для прогнозування мікроклімату, що сприяє підвищенню комфорту проживання та зменшенню витрат на енергоресурси.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Берестегі, Г. Від діагностики до прогнозування забруднення повітря за допомогою нейронних мереж: моніторинг забруднення повітря в Більбао [Текст] / Г. Берестегі, А. Еліас, А. Барона // *Environmental Modelling & Software*. – 2008. – Т. 23. – С. 622–637.
2. Дюрао, Р. Прогнозування рівнів озону у промислових зонах за 24 години наперед з використанням дерев класифікації та моделей MLP [Текст] / Р. Дюрао, М. Мендес, Ж. Перейра // *Atmospheric Pollution Research*. – 2016. – Т. 7. – С. 961–970.
3. Namini, S. A Comparison of ARIMA and LSTM in Forecasting Time Series [Текст] / S. Namini, N. Tavakoli, A. Namin // *IEEE International Conference on Machine Learning and Applications*. – 2018. – Vol. 17. – С. 75–80.
4. Zeng, Z. Self CNN-based time series stream forecasting [Текст] / Z. Zeng, H. Xiao, X. Zhang // *Electronics Letters*. – 2016. – Vol. 52. – P. 1857–1858.
5. Іващенко, Г. С. Прогнозування часових рядів на основі методу висновків за прецедентами з використанням моделей штучних імунних систем [Текст] / Г. С. Іващенко, Н. М. Корабльов // *System technologies*. – 2014. – № 6. – С. 43–51.
6. Ebrahimpour, R. Mixture of MLP-experts for trend forecasting of time series: A case study of the Tehran stock exchange [Текст] / R. Ebrahimpour, H. Nikoo, S. Masoudnia // *International Journal of Forecasting*. – 2011. – Vol. 27. – P. 804–816.
7. Palit, A. *Computational Intelligence in Time Series Forecasting* [Текст] / A. Palit, D. Popovic. – 2006. – 372 p.
8. Chen, S. *Computational Intelligence in Economics and Finance* [Текст] / S. Chen, P. Wang. – 2007. – 53 p. DOI: <https://doi.org/10.1007/978-3-662-06373-6>. ISBN: 978-3-662-06373-6.

9. Mohammed, A. S. Green Hybrid Models Based on Clonal Selection and Case-based Reasoning for Short-term Time Series Forecasting [Текст] / A. S. Mohammed, H. Ivashchenko, T. Filimonchuk // Journal of Green Engineering. – 2020. – Vol. 10. – P. 2139–2154.

10. Piccialli, F. Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion [Текст] / F. Piccialli, F. Giampaolo, P. Edoardo // Information. – 2021. – Vol. 74. – P. 1–16.

11. Kousari, M. Introducing an operational method to forecast long-term regional drought based on the application of artificial intelligence capabilities [Текст] / M. Kousari, M. Hosseini, H. Ahani // Theoretical and Applied Climatology. – 2015. – Vol. 127. – P. 361–380.

12. Mostafa, M. A neuro-computational intelligence analysis of the ecological footprint of nations [Текст] / M. Mostafa, R. Natarajan // Computational Statistics & Data Analysis. – 2009. – Vol. 53. – P. 3516–3531.

13. Afan, H. Past, present and prospect of an Artificial Intelligence (AI) based model for sediment transport prediction [Текст] / H. Afan, A. Shafie, W. Mohtar // Journal of Hydrology. – 2016. – Vol. 541. – P. 902–913.

14. Chatfield, C. Time-Series Forecasting [Текст] / C. Chatfield. – Chapman and Hall/CRC, 2000. – 280 p.

15. Li, S. Deterministic fuzzy time series model for forecasting enrollments [Текст] / S. Li, Y. Cheng // Computers & Mathematics with Applications. – 2007. – Vol. 53. – P. 1904–1920.

16. Wang, Y. Hybrid models for time series forecasting: A review [Текст] / Y. Wang, X. Zhang // Journal of Time Series Analysis. – 2019. – Vol. 40. – P. 367–389.

17. Lim, H. Time series forecasting using recurrent neural networks [Текст] / H. Lim, Y. Zha // Journal of Forecasting. – 2018. – Vol. 37. – P. 222–230.

18. Chen, Y. Long Short-Term Memory Network for Time Series Forecasting: A Review [Текст] / Y. Chen, A. Kechagias, C. J. A. M. – 2019. – Vol. 45. – P. 168–185.

19. Bontempi, G. Machine Learning Strategies for Time Series Forecasting [Текст] / G. Bontempi, Y. Ben Taieb, P. Geurts // 2012 IEEE International Conference on Data Mining Workshops. – 2012. – P. 257–264.
20. Xiong, Y. Forecasting with LSTM neural network [Текст] / Y. Xiong, J. Zeng // IEEE Transactions on Neural Networks and Learning Systems. – 2018. – Vol. 29. – P. 1856–1862.
21. Ahmed, F. Time series forecasting using LSTM and ARIMA models [Текст] / F. Ahmed, M. A. Khan // International Journal of Computer Applications. – 2021. – Vol. 174. – P. 22–28.
22. Fong, H. Deep learning for time series forecasting: A survey [Текст] / H. Fong, C. Wang // Future Generation Computer Systems. – 2020. – Vol. 109. – P. 689–702.
23. Shalev-Shwartz, S. Understanding Machine Learning: From Theory to Algorithms [Текст] / S. Shalev-Shwartz, S. Ben-David. – Cambridge University Press, 2014. – 352 p.
24. Feng, Y. Forecasting short-term power load using LSTM neural network [Текст] / Y. Feng, R. Yu // Energy Reports. – 2021. – Vol. 7. – P. 1014–1022.
25. Konečný, J. Time series prediction using long short-term memory networks [Текст] / J. Konečný, J. Chytil // Journal of Electrical Engineering. – 2020. – Vol. 71. – P. 329–337.
26. Roberts, A., Chen, L., Williams, J. Implementing Machine Learning Models in .NET Using ML.NET: A Practical Guide [Текст] / A. Roberts, L. Chen, J. Williams // Journal of Software Engineering and Applications. – 2023. – Vol. 58. – P. 120–132.
27. Зубенко, Д. Р. Керування мікрокліматом будинку з використанням засобів машинного навчання [Текст] / Зубенко Д. Р., Іващенко Г. С. // Проблеми інформатизації: Тези доповідей дванадцятої міжнародної науково-технічної конференції. Баку – Харків – Бельсько-Бяла. – 21-22 листопада 2024. – Том 2, С. 83.