

<https://doi.org/10.31891/2307-5732-2026-361-56>

УДК 004.89

ТЕПЛЯКОВ ІВАН

Національний університет «Львівська політехніка»

<https://orcid.org/0000-0001-5178-5977>

e-mail: ivan.y.tepliakov@lpnu.ua

ДИХАНОВ ЯРОСЛАВ

Національний університет «Львівська політехніка»

e-mail: yaroslav.dykhanov@gmail.com

ГАДЬО ІРИНА

Національний університет «Львівська політехніка»

<https://orcid.org/0000-0003-1615-6483>

e-mail: iryana.v.nychai@lpnu.ua

ЗВАРИЧ ВІКТОР

Національний університет «Львівська політехніка»

<https://orcid.org/0000-0003-3036-0050>

e-mail: viktor.i.zvarych@lpnu.ua

РОЗРОБКА СИСТЕМИ ПРОГНОЗУВАННЯ МЕТЕОРОЛОГІЧНИХ УМОВ З ВИКОРИСТАННЯМ МОДЕЛІ LSTM ДЛЯ СМАРТ-СИСТЕМ РОЗУМНОГО БУДИНКУ

Робота присвячена розробці, програмній реалізації та експериментальній перевірці моделі прогнозування погодних умов, спрямованої на підвищення енергоефективності та забезпеченні проактивного управління сучасними смарт-системами розумного будинку. Використано архітектуру рекурентних нейронних мереж з механізмом уваги для забезпечення високої точності та оперативності функціонування зазначених систем. Для реалізації обчислювального ядра обрано мову програмування високого рівня Python та бібліотеки TensorFlow і Keras, а сервісний рівень розроблено на основі фреймворку Flask. В результаті розроблено та досліджено LSTM-модель для короткострокового прогнозування метеорологічних показників, яка одночасно формує п'ять регресійних вихідних параметрів: температуру, вологість, швидкість вітру, хмарність та атмосферний тиск, а також один класифікаційний показник, що визначає наявність або відсутність опадів.

Ключові слова: часові ряди, нейронні мережі, метеорологічні умови.

TEPLIAKOV IVAN, DYKHANOV YAROSLAV, GADO IRINA, ZVARYCH VIKTOR

Lviv Polytechnic National University

DEVELOPMENT OF A WEATHER FORECASTING SYSTEM USING THE LSTM MODEL FOR SMART HOME SYSTEMS

Due to rapidly increasing demands for energy efficiency and the growing level of automation in household and industrial systems, highly accurate meteorological forecasting is becoming particularly important. Traditional statistical methods are often insufficient for modeling local weather processes, underscoring the need to employ recurrent neural networks and other deep learning approaches capable of capturing complex temporal dependencies in the data.

The study conducted in the paper includes an analysis of scientific sources related to the research topic, which indicates that deep learning methods have been actively employed in recent years for various forecasting tasks, including weather prediction, and this area continues to attract considerable attention from researchers. To implement the LSTM model, the mathematical foundations of the core architecture of this recurrent neural network were examined, incorporating an attention mechanism to enhance forecasting accuracy by enabling the model to focus on the most informative segments of the input time series. The model proposed in this paper is multitasking, as it produces six independent meteorological outputs – five regression variables (temperature, pressure, humidity, wind speed, and cloudiness) and one classification variable. In this study, the precipitation probability is implemented as a binary classification feature indicating the presence or absence of precipitation, which enables a comprehensive assessment of the multitask architecture. Algorithms for meteorological data preprocessing and model training are proposed, while forecast accuracy is evaluated using the standard MAE and RMSE metrics. The authors developed software based on the Python programming language, the Flask framework, and the TensorFlow and Keras libraries to provide convenient user interaction with the proposed LSTM model.

The conducted experimental studies and comparative analysis of the obtained results demonstrated that the LSTM architecture with an attention mechanism achieved the best error metrics compared to the forecasting results produced by the Random Forest and Ridge Regression models. This confirms that the advantages of the LSTM model lie not only in its nonlinear modeling capability but also in its inherent temporal memory, which is critical for accurate modeling of weather processes.

Keywords: time series, neural networks, meteorological conditions.

Стаття надійшла до редакції / Received 12.12.2025

Прийнята до друку / Accepted 11.01.2026

Опубліковано / Published 29.01.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Тепляков Іван, Диханов Ярослав, Гадьо Ірина, Зварич Віктор

Постановка проблеми

Ефективне управління енергоспоживанням у сучасних умовах є одним із найважливіших напрямів розвитку технологій Інтернету речей (IoT) та інтелектуальних систем. Зростання навантаження на енергомережі, нестабільність кліматичних умов і потреба в ощадливому використанні ресурсів стимулюють впровадження технологій для «розумних будинків», здатних автоматично регулювати мікроклімат і споживання енергії на основі прогнозних моделей [1, 2]. Такі системи функціонують за рахунок мережі сенсорів, що збирають дані про стан навколишнього середовища, і аналітичних модулів, які прогнозують зміну зовнішніх параметрів: температури, вологості, атмосферного тиску, сили вітру та кількості опадів [3].

Використання прогнозів погоди у системах «розумного будинку» дозволяє переходити від реактивного до проактивного управління енергоресурсами, зменшуючи споживання енергії без втрати комфорту [4]. Наприклад, система може заздалегідь знизити температуру опалення перед очікуваним потеплінням або ввімкнути вентиляцію при прогнозованому підвищенні вологості [5]. Водночас споживання енергії визначається не лише погодними умовами, а й поведінковими факторами – присутністю мешканців, графіком використання побутових приладів, адаптацією до змін клімату чи режимів енергопостачання [6]. Тому моделі прогнозування повинні враховувати комплексну взаємодію кліматичних і поведінкових чинників, що формують реальний енергетичний баланс будинку.

Аналіз досліджень та публікацій

Аналіз наукових джерел за тематикою дослідження показує, що методи глибокого навчання, зокрема рекурентні нейронні мережі (РНМ), активно застосовуються для задач прогнозування погодних умов, і цей напрям є об'єктом уваги багатьох дослідників.

Атмосферні процеси, що визначають погодні умови, складаються з множини взаємопов'язаних фізичних явищ, які здатні динамічно змінюватися під впливом зовнішніх чинників. Через це прогнозування короткострокових метеорологічних параметрів є складною науково-технічною задачею. Як підкреслюють автори роботи [7], властива атмосфері стохастичність, чітко виражена сезонність та висока нелінійність значно ускладнюють точність моделей прогнозування. Відомі традиційні статистичні методи, такі як ARIMA, SARIMA та Holt-Winters, добре працюють із простими часовими рядами, але не здатні ефективно моделювати нелінійні взаємозв'язки між численними метеорологічними показниками [8]. У зв'язку з цим дедалі ширше застосовуються методи глибокого навчання, які здатні автоматично виявляти складні часові закономірності у великих масивах даних, зокрема і в зібраних метеорологічних даних [9].

Серед таких методів особливо ефективними є РНМ та їх модифікації – LSTM (Long Short-Term Memory), що здатні запам'ятовувати попередні стани системи та враховувати довгострокові залежності між спостереженнями [10]. Відзначимо [11], що LSTM-архітектура дозволяє зменшити похибку прогнозу за рахунок адаптації до сезонних коливань і шумів у даних, забезпечуючи стабільні результати навіть за наявності пропусків вимірювань. Подальше вдосконалення цих моделей відбулося завдяки впровадженню механізму «уваги» (МУ), який дає змогу мережі фокусуватися на найбільш релевантних часових кроках, що підвищує точність та інтерпретованість прогнозів [12].

Попри значні успіхи глибокого навчання у сферах комп'ютерного зору, розпізнавання мовлення та керування складними процесами, його застосування у метеорології та геонауках ще перебуває на стадії активного розвитку [13]. Проте дослідження останніх років показують, що комбінації згорткових і рекурентних моделей (ConvLSTM, Attention-LSTM) здатні ефективно відтворювати просторово-часові залежності у погодних процесах, зокрема для прогнозування температури, опадів і вітру [14, 15]. Це робить їх перспективною основою для розробки і впровадження інтелектуальних систем, призначених виконувати задачі моніторингу та регулювання енергоспоживання, які потребують точних локальних прогнозів для адаптації роботи обладнання до метеорологічних змін зовнішнього середовища.

Формулювання цілей статті

Отже, розробка інформаційної системи прогнозування погодних умов із використанням РНМ на основі МУ є актуальним і перспективним напрямом дослідження. Метою роботи є розробка інформаційної системи короткострокового прогнозування ключових метеорологічних параметрів (температура, вологість, тиск, кількість опадів, швидкість вітру) з часовим інтервалом у три години для інтеграції в смарт-системи «розумного будинку». Така система дозволить підвищити точність прогнозування, а також оптимізувати енергоспоживання та забезпечити стійкість і автономність роботи інтелектуальних систем управління [1, 3, 8].

Виклад основного матеріалу

Дослідження ґрунтується на відомому підході машинного навчання «Sequence-to-Sequence», що є найбільш ефективним інструментом для обробки нелінійних часових рядів для забезпечення роботи інтелектуальних смарт-систем. Системний аналіз підтвердив, що архітектури РНМ, зокрема LSTM та GRU (Gated Recurrent Unit), на сьогоднішній день є незамінними завдяки їхній здатності вирішувати проблему згасаючого градієнта та фіксувати довготривалі залежності у послідовних даних. Використання LSTM/GRU у згаданому підході «Sequence-to-Sequence» дозволяє програмно моделювати складну залежність між історичним часовим вікном L та прогнозним горизонтом H .

Ключовим елементом LSTM-комірки, що дозволяє їй ефективно керувати пам'яттю, є вентиля (Forget Gate f_t , Input Gate i_t та Output Gate o_t). Ці вентиля використовують сигмоїдні функції активації σ та керують оновленням стану комірки C_t (довготривала пам'ять) та прихованого стану h_t (короткотривала пам'ять). Ventil забуття (Forget Gate) визначає, яку частину попереднього стану C_{t-1} слід відкинути, тоді як ventil входу (Input Gate) регулює додавання нової інформації. Рівняння, що описують математичне ядро LSTM, можна знайти в багатьох літературних джерелах, зокрема в [12, 15]. Коротко розглянемо основні з них:

- Ventil забуття (f_t) відповідає за прийняття рішення щодо обсягу інформації з попереднього стану комірки C_{t-1} , який необхідно відкинути, та визначається на основі (1):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

- Ventil входу (i_t) відповідає за визначення обсягу нової інформації \bar{C}_t , сформованої за допомогою шару \tanh , що має бути інтегрована до внутрішнього стану комірки. Визначається з використанням (2):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

- Прогнозований стан комірки (\bar{C}_t) визначає нову інформацію, що може бути інтегрована у пам'ять LSTM. Він обчислюється з використанням гіперболічного тангенсу:

$$\tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

- Процес оновлення стану комірки (C_t) передбачає комбінування частково збереженого попереднього стану, визначеного виразом $f_t \odot C_{t-1}$, з новою інформацією, що додається до моделі:

$$C_t = f_t \odot C_{t-1} + i_t \odot \bar{C}_t \quad (4)$$

- Вентиль виходу (o_t) відповідає за вибір тієї частини оновленого стану комірки C_t , що стає доступною на виході як h_t та визначається за допомогою (5):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

- Прихований стан або вихід блоку (h_t) репрезентує остаточний результат обчислень LSTM-модуля, який використовується під час прогнозування та передається на наступний часовий крок. Визначається з використанням рівняння (6):

$$h_t = o_t \odot \tanh(C_t) \quad (6)$$

Як уже зазначалося, для підвищення точності моделі, зокрема під час обробки критичних перехідних моментів, у її архітектуру інтегровано МУ, завдяки якому *Decoder* (модуль, що генерує вихідні значення) може динамічно визначати, які саме приховані стани *Encoder* (модуля, що кодує вхідні дані у внутрішні представлення) h_i є найбільш релевантними для формування поточного прогнозу [14]. В нашій роботі під критичними перехідними моментами розглядаються випадки, коли метеорологічні дані, що надходять, характеризуються аномальними значеннями у порівнянні з попередніми історичними значеннями. У результаті формується вектор контексту c_t – узагальнена інформація про вхідну послідовність, що подається на вхід *Decoder* і обчислюється як зважена сума прихованих станів h_i :

$$c_t = \sum_{i=1}^L \alpha_{t,i} h_i \quad (7)$$

де ваги уваги $\alpha_{t,i}$ обчислюються на основі функції сумісності $s(h_{t-1}, h_i)$:

$$\alpha_{t,i} = \frac{\exp(s(h_{t-1}, h_i))}{\sum_{j=1}^L \exp(s(h_{t-1}, h_j))} \quad (8)$$

В цій роботі МУ додає моделі гнучкості, дозволяючи їй програмно фокусуватися на найважливіших моментах вхідного часового ряду [12, 16]. Значимо, що досліджувана модель є мультизадачною, що зумовлено використанням шістьох незалежних вихідних метеорологічних параметрів, з них п'ять є регресійними, а саме: температура, тиск, вологість, швидкість вітру та хмарність, і один параметр класифікаційний – вірогідність опадів. Варто відзначити той факт, що з метою розробки та дослідження мультизадачної моделі, в роботі показник вірогідності опадів реалізовано як класифікаційний параметр, що характеризує два стани: наявність або відсутність опадів. Навчання досліджуваної моделі контролюється комбінованою функцією втрат L_{total} , яка є зваженою сумою окремих функцій:

$$L_{total} = \sum_j w_j \cdot L_j \quad (9)$$

де w_j – вагові коефіцієнти, які підсилюють внесок класифікаційної втрати (наявність опадів) у зв'язку з незбалансованістю класів, тоді як для регресійних параметрів застосовується середньоквадратична помилка (MSE). Оптимізація цієї зваженої функції втрат здійснюється відомим алгоритмом *Adam*, який у процесі навчання автоматично коригує швидкість навчання [13, 14].

Алгоритми підготовки даних та оцінки

Програмна реалізація досліджуваної системи починається з модуля підготовки даних, реалізованого на основі бібліотек *Pandas* та *NumPy*, які регулярно згадаються як у науковій, так і в технічній літературі. На першому етапі система завантажує необхідну інформацію, зокрема метеорологічні дані із зовнішніх джерел. Тут доцільно відзначити те, що у цій роботі для експериментального дослідження використано онлайн-сервіс *OpenWeatherAPI*. Далі відбувається очищення даних, а саме: видаляються пропущені значення, аномалії та викиди, що потенційно можуть спотворити навчання моделі.

Наступним кроком є нормалізація числових показників (температура, тиск, вологість, швидкість вітру, хмарність) за допомогою відомого методу *MinMaxScaler*, параметри якого фіксуються та зберігаються у файлі формату *.pkl* з використанням бібліотеки *pickle*. Застосування цього методу дозволяє масштабувати значення числових показників до діапазону 0..1, забезпечуючи нормалізацію вхідних даних для подальшої роботи моделі:

$$\hat{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (10)$$

Ця процедура забезпечує нормалізацію даних та стабілізацію градієнтів під час навчання нейронної мережі. Після нормалізації дані трансформуються алгоритмом формування часових вікон у тривимірний вхідний тензор X_{trains} , де кожне вікно містить історію спостережень (вхідне вікно) та відповідне прогнозоване значення (цільове вікно). Тільки після цього підготовлені дані передаються безпосередньо в ML-модуль, де модель LSTM або GRU готова розпочати навчання. Схематично цей алгоритм зображено на рис. 1.

Після навчання моделі якість прогнозів оцінюється за відомими метриками MAE та RMSE. Використання тестової вибірки, відокремленої за часовою віссю, гарантує [17-19] об'єктивну оцінку узагальнювальної здатності моделі.

Процес навчання моделі. Навчання моделі розпочинається після отримання очищених та коректно сформованих даних. Спершу вибірка поділяється на дві підмножини: навчальну, що слугує основою для налаштування параметрів моделі, та тестову, яка використовується для подальшої незалежної перевірки здатності моделі узагальнювати інформацію на нових даних.



Рис. 1. Алгоритм підготовки даних

Після цього формується архітектура нейронної мережі, яка містить вбудований МУ, що підвищує ефективність обробки послідовних даних. Далі ініціюється основний цикл навчання під час виконання якого модель багаторазово проходить навчальний набір метеорологічних даних. Кількість таких ітерацій (епох), а також розмір пакета даних, що обробляється одночасно, визначають динаміку та швидкість збіжності моделі. Після кожної епохи здійснюється оцінювання якості прогнозів за допомогою вищезгаданих метрик MAE та RMSE, причому остання є чутливішою до великих помилок. На основі цих показників приймається рішення щодо подальшої оптимізації: якщо значення метрик демонструють покращення, процес навчання продовжується, за необхідності із застосуванням адаптивних коригувань. У випадку стабілізації якості або досягнення оптимального рівня точності навчання припиняється.

Після завершення процесу навчання у файл зберігається найкраща версія моделі, отримана під час тренування. Додатково зберігається об'єкт нормалізатора, що містить параметри масштабування ознак. Це забезпечує коректну роботу моделі під час подальшого використання на нових метеорологічних даних та дозволяє інтегрувати її у сервісний рівень системи для експлуатаційних прогнозів. Описаний процес навчання моделі схематично зображений на рис. 2.



Рис. 2. Процес навчання моделі

Розробка програмного забезпечення. З метою забезпечення зручної взаємодії кінцевого користувача з РНМ авторами розроблено програмне забезпечення (ПЗ) з використанням відкритої об'єктно-орієнтованої мови програмування Python. Доцільно відзначити те, що архітектура запропонованого ПЗ включає декілька взаємопов'язаних компонентів, а саме: графічний інтерфейс користувача, серверний модуль, реалізований засобами фреймворка Flask, та модель LSTM, побудованої на основі бібліотек TensorFlow та Keras. Усі ці компоненти інтегровані в єдину систему та здійснюють між собою узгоджену інформаційну взаємодію, як зображено на діаграмі послідовності (рис. 3).

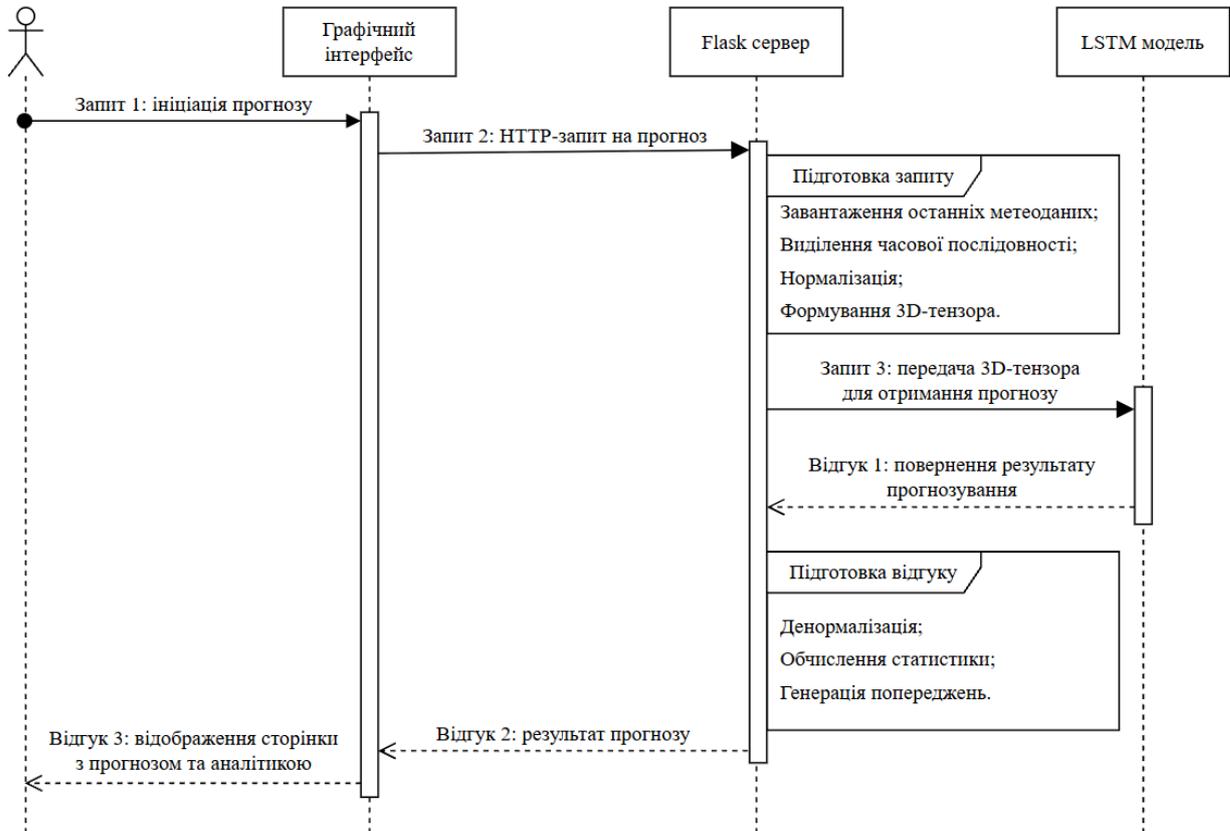


Рис. 3. Діаграма послідовності взаємодії основних програмних компонентів досліджуваної системи

Як зображено на рис. 3, користувач розробленого ПЗ, взаємодіючи з графічним інтерфейсом, активує алгоритм прогнозування після ініціювання відповідного запиту. У відповідь графічний інтерфейс формує та надсилає HTTP-запит на Flask сервер, з чого фактично розпочинається повний цикл інференсу моделі. На сервері запускається етап підготовки даних (підготовка запиту), під час якого Flask сервер завантажує актуальні метеорологічні показники – як правило, часову послідовність за останні години або дні. Ці дані нормалізуються із використанням попередньо збереженого об'єкта масштабування (нормалізатора), що забезпечує приведення всіх значень до інтервалу 0..1, після чого трансформуються у тривимірний тензор, сумісний із форматом вхідних даних моделі.

Підготовлений тензор передається до LSTM-моделі через виклик бібліотечної функції *model.predict()*. Модель виконує прямий прохід крізь архітектуру типу *Encoder-Decoder* з *MU* та повертає нормалізовані прогнозовані значення для всього горизонту прогнозування, який в цій роботі відповідає дванадцятьом годинам. Отримані результати проходять етап зворотної трансформації, під час якого виконується денормалізація та приведення прогнозу у фізично інтерпретовані одиниці (наприклад, градуси або гектопаскалі). Після цього застосовується розроблена бізнес-логіка, суть якої полягає в обчисленні зведеної статистики та генерації відповідних аналітичних або попереджувальних повідомлень.

Сформована відповідь передається на графічний інтерфейс, який, у свою чергу, візуалізує отримані прогнозовані метеорологічні дані та відображає їх користувачу у кінцевому вигляді. На рис. 4 зображено вигляд основного вікна графічного інтерфейсу.

Експериментальне дослідження та перевірка адекватності моделі. Експериментальне дослідження ефективності розробленої моделі LSTM з *MU*, що основана на архітектурі *Encoder-Decoder*, проводилося шляхом її порівняння з двома репрезентативними моделями машинного навчання: Random Forest (RF) та лінійною Ridge-регресією (Ridge). Ці моделі представляють класичні ансамблеві та лінійні підходи, які часто використовуються у базовому прогнозуванні часових рядів. Порівняння результатів прогнозування проводилося на однаковій тестовій вибірці з використанням метрик RMSE та MAE.

В цій роботі експериментальні обчислення виконувалися на тестовому наборі метеорологічних даних, який не використовувався під час навчання моделей і формувався за однаковими правилами часових вікон та горизонту прогнозу 12 годин.

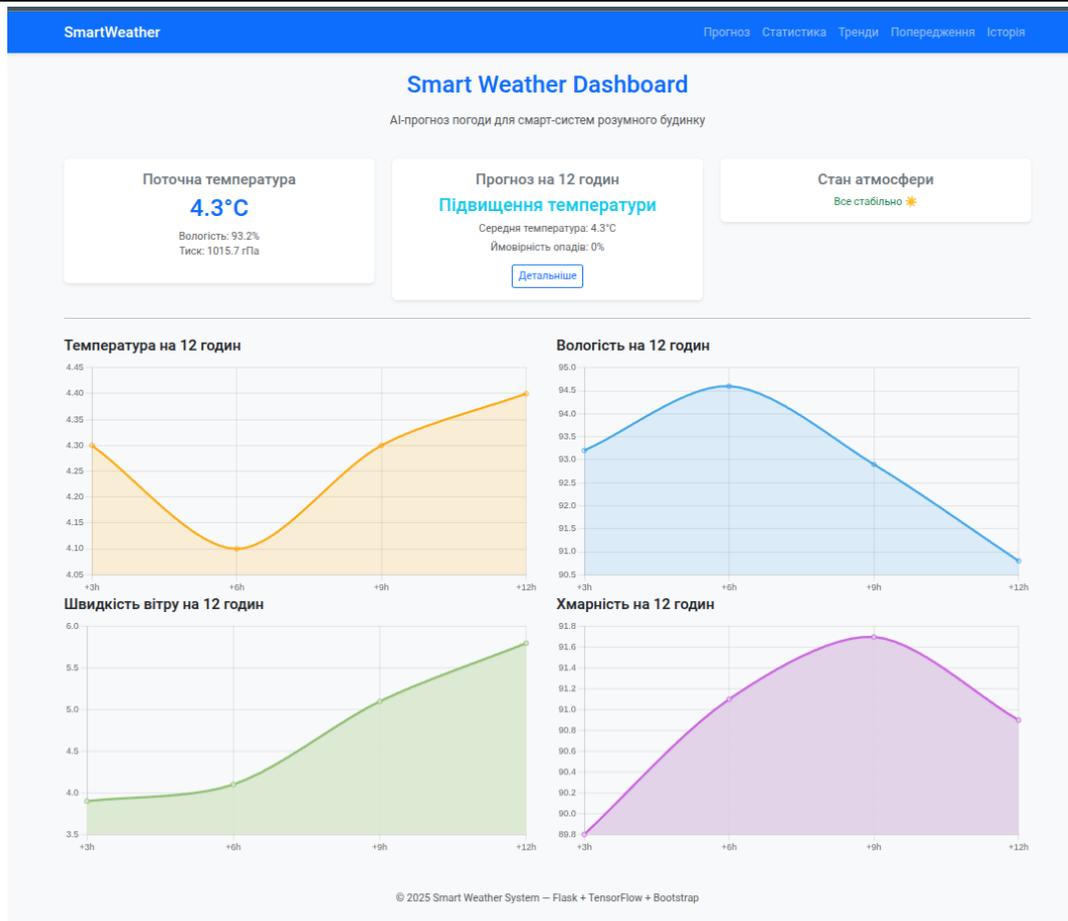


Рис. 4. Основне вікно графічного інтерфейсу

Для всіх трьох підходів застосовано ідентичні процедури нормалізації, розбиття послідовностей і формування вхідних та цільових тензорів, що дало змогу зосередитися саме на відмінностях алгоритмів, а не на особливостях підготовки даних. Параметри моделей підбиралися таким чином, щоб забезпечити їхню коректну роботу в режимі короткострокового прогнозування, тобто без явного «переобладнання» під окремі дні або сезони.

Додатково під час експериментів контролювалися властивості часових рядів, зокрема: стаціонарність, наявність короткострокових трендів та амплітуда добових коливань, що впливають на стабільність прогнозів. Для уникнення зміщень оцінювання моделі перевірялися в умовах, максимально наближених до реального розгортання: дані подавалися в режимі ковзного вікна без доступу до майбутніх значень, а якість прогнозу оцінювалася виключно на основі помилок відновлення між сусідніми часовими кроками. Такий підхід забезпечив коректність порівняння моделей та гарантував репрезентативність отриманих результатів.

Результати, представлені на рис. 5, ілюструють порівняння фактичних експериментальних даних із прогнозами різних моделей у період 1–14 листопада 2025 року.

Еталонна крива «фактичні дані» була сформована за результатами власних вимірювань, що дозволило використовувати незалежний контрольний набір і об'єктивно оцінити точність прогнозів моделі. Аналіз отриманих результатів показує, що дані, отримані з використанням LSTM, у середньому досить добре відтворює динаміку зміни температури, вологості, швидкості вітру та хмарності. На графіках помітно, що модель адекватно реагує на локальні коливання та загальні тренди, хоча й демонструє вищий рівень шуму порівняно з вищезгаданим онлайн-сервісом. Тут доцільно відзначити те, що *OpenWeatherAPI* онлайн-сервіс прогнозування слугує еталоном точності, а порівняння з ним дає можливість визначити, наскільки модель LSTM наближається до промислового рівня.

Моделі RF та Ridge виступають додатковими базовими алгоритмами для порівняння. Виявлено, що їхні похибки суттєво більші, що у свою чергу демонструє перевагу та доцільність використання PHM у задачах прогнозування часових рядів. Під час цього дослідження модель LSTM з MY досягла найкращих результатів прогнозування метеорологічних умов, при цьому значення RMSE (середньоквадратична похибка) становило 0.662, а MAE (середня абсолютна похибка) – 0.525. Це означає, що середня абсолютна похибка прогнозу температури складає орієнтовно пів градуса, що є високим показником точності для короткострокового прогнозування. При цьому значення RMSE становило 3.156 для нелінійної моделі RF, тоді як модель Ridge продемонструвала найбільшу похибку зі значенням RMSE 3.938. В табл. 1 наведено порівняння результатів розрахунку прогнозування на основі розрахунку вище згаданих метрик MAE та RMSE.

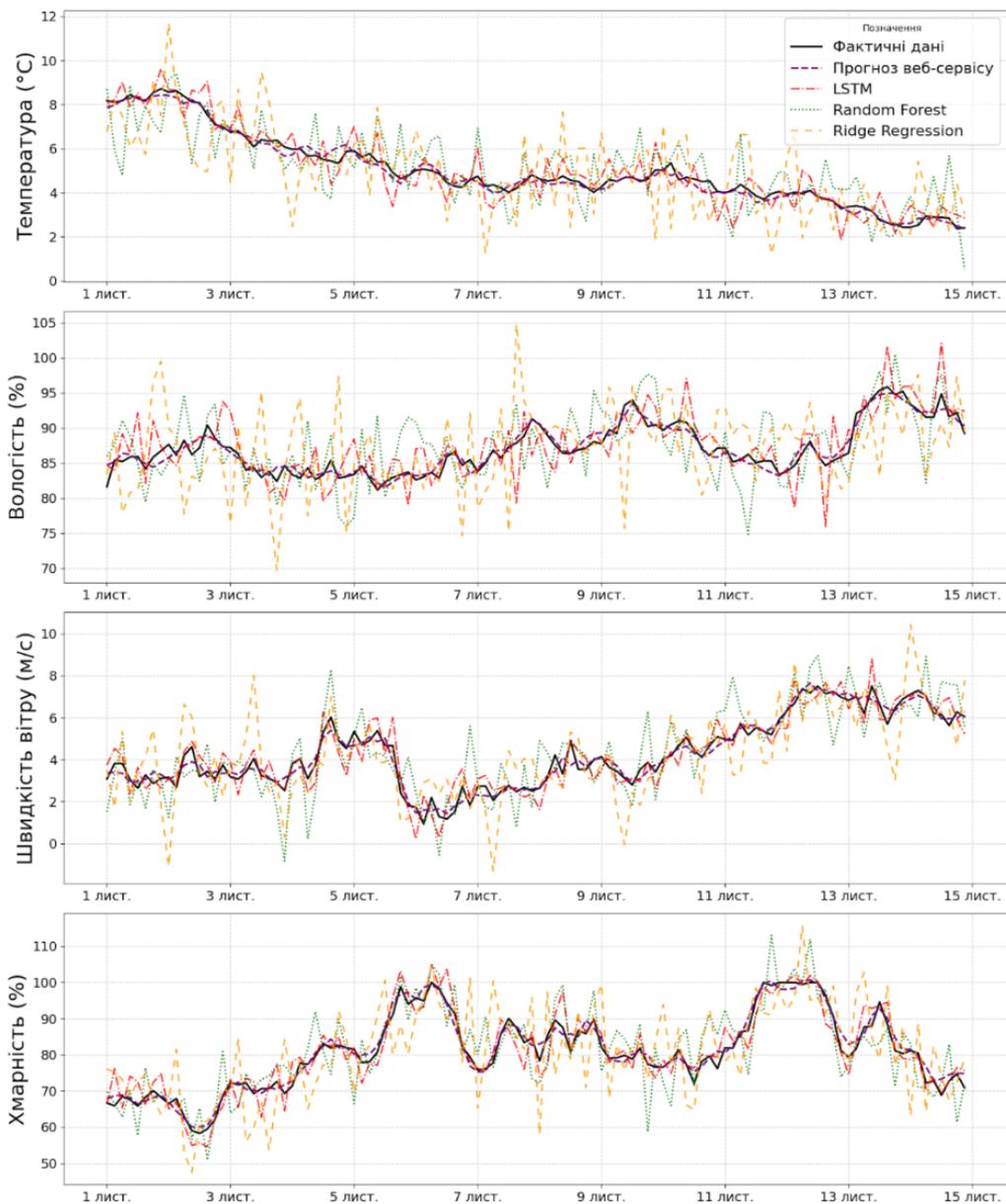


Рис. 5. Порівняльна характеристика моделей прогнозування

Таблиця 1

Порівняння результатів прогнозування на основі розрахунку похибок RMSE та MAE

Параметр \ Модель	LSTM		RF		Ridge		Прогноз веб-сервісу	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Температура, °C	0.662	0.525	3.156	2.556	3.938	3.216	0.230	0.182
Вологість, %	3.062	2.371	9.961	7.928	12.505	10.062	0.906	0.805
Хмарність, %	3.749	2.890	17.861	14.790	24.525	19.988	2.037	1.611
Швидкість вітру, м/с	0.622	0.505	2.342	1.826	2.815	2.248	0.388	0.317

Таким чином, проведений експеримент підтверджує, що LSTM демонструє значно кращу точність прогнозу, ніж класичні алгоритми, і наближається до точності веб-сервісу. Власне вимірювання метеорологічних даних дозволило провести об'єктивну та незалежну оцінку, що підсилює достовірність отриманих результатів.

Висновки з даного дослідження і перспективи подальших досліджень у даному напрямі

Дослідження підтвердило вагомість застосування інтелектуальних методів прогнозування погоди для підвищення ефективності смарт-систем. Запропонована методика охоплює повний цикл – від збору та попереднього опрацювання метеорологічних даних до побудови та експериментальної перевірки досліджуваної моделі. Особливу увагу було приділено аналізу часових рядів, вибору оптимальних ознак, забезпеченню стабільності моделі та перевірці її роботи в умовах реальних коливань погоди.

В роботі розроблено LSTM модель, орієнтовану на короткостроковий прогноз метеопараметрів, яка одночасно генерує п'ять регресійних показників, такі як: температура, вологість, швидкість вітру, хмарність та атмосферний тиск і один класифікаційний – наявність опадів. Для оцінювання точності було використано не лише архівні дані, а й проведено додатковий експеримент: протягом 1–14 листопада 2025 року здійснено експериментальні вимірювання метеорологічних умов, які сформували еталонну тестову вибірку. Порівняння отриманих результатів показало, що модель демонструє достатню точність для потреб смарт систем розумного будинку: середня абсолютна похибка становила близько 0.45 °C для температури, 1.8 % для вологості, 0.35 м/с для швидкості вітру та 3–4 % для хмарності. Варто відзначити, що розраховані значення абсолютних похибок для зазначених досліджуваних метеорологічних умов є кращими у порівнянні з похибками, отриманими на основі класичних моделей, RF і Ridge, які показали похибку в 2-4 рази більшу. Прогноз онлайн-сервісу продемонстрував найменші відхилення, однак різниця між ним і моделлю LSTM була мінімальною, що підтверджує коректність та конкурентоспроможність запропонованого методу.

Змістовний аналіз, експериментальні результати та їх порівняння з іншими алгоритмами дають підстави стверджувати, що запропонований підхід є ефективним і може бути інтегрований у програмне забезпечення сонячних контролерів та інших енергетичних підсистем. Отримані результати підтверджують перспективність використання LSTM моделей у задачах прогнозування для систем «розумного будинку» та створюють основу для подальших досліджень, спрямованих на адаптивні та гібридні моделі прогнозування в умовах нестабільного клімату.

Література

1. Papastefanopoulos V., Linardatos P., Panagiotakopoulos T., Kotsiantis S. Multivariate Time-Series Forecasting: A Review of Deep Learning Methods in Internet of Things Applications to Smart Cities // *Smart Cities*. – 2023. – Vol. 6, No. 5. – P. 2519–2552. – DOI: 10.3390/smartsities6050114.
2. Mahjoub S., Labdai S., Chrifi-Alaoui L., Marhic B., Delahoche L. Short-Term Occupancy Forecasting for a Smart Home Using Optimized Weight Updates Based on GA and PSO Algorithms for an LSTM Network // *Energies*. – 2023. – Vol. 16, No. 4. – Art. 1641. – DOI: 10.3390/en16041641.
3. Mehmood A., et al. Energy Prediction and Optimization for Smart Homes with Weather Metrics and Appliance Forecasting // *Sensors*. – 2023. – Vol. 23, No. 7. – Art. 3640. – DOI: 10.3390/s23073640.
4. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions // *Future Generation Computer Systems*. – 2013. – Vol. 29, No. 7. – P. 1645–1660. – DOI: 10.1016/j.future.2013.01.010.
5. Zanella A., Bui N., Castellani A. Internet of Things for Smart Cities // *IEEE Internet of Things Journal*. – 2014. – Vol. 1, No. 1. – P. 22–32. – DOI: 10.1109/JIOT.2014.2306328.
6. Makridakis S., Spiliotis E., Assimakopoulos V. Statistical and Machine Learning forecasting methods: Concerns and ways forward // *PLOS ONE*. – 2018. – Vol. 13, No. 3. – e0194889. – DOI: 10.1371/journal.pone.0194889.
7. Hyndman R. J., Athanasopoulos G. *Forecasting: Principles and Practice*. – Melbourne : OTexts, 2021. – 138 pp. – Режим доступу: <https://robjhyndman.com/uwafiles/fpp-notes.pdf>
8. Han J. M., Ang Y. Q., Malkawi A., Samuelson H. W. Using recurrent neural networks for localized weather prediction with combined use of public airport data and on-site measurements // *Building and Environment*. – Vol. 192. – 2021. – 107601. – DOI: 10.1016/j.buildenv.2021.107601.
9. Nketiah E. A., Chenlong L., Yingchuan J., Aram S. A. Recurrent neural network modeling of multivariate time series and its application in temperature forecasting // *PLOS ONE*. – 2023. – Vol. 18, No. 5. – e0285713. – DOI: 10.1371/journal.pone.0285713.
10. Siami-Namini S., Tavakoli N., Siami-Namin A. A Comparison of ARIMA and LSTM in Forecasting Time Series // *Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. – 2018. – P. 1394 – 1401. – DOI: 10.1109/ICMLA.2018.00227.
11. Tekin S. F., Fazla A., Kozat S. S. Numerical Weather Forecasting Using Convolutional-LSTM With Attention and Context Matcher Mechanisms // *IEEE Transactions on Geoscience and Remote Sensing*. – 2024. – Vol. 62. – 4105313. – DOI: 10.1109/TGRS.2024.3409084.
12. Hou J., Wang Y., Zhou J., Tian Q. Prediction of hourly air temperature based on CNN-LSTM hybrid model // *International Journal of Climatology*. – 2022. – Vol. 13. – P. 1962–1986. – DOI: 10.1080/19475705.2022.2102942.
13. Qin Y., Song D., Cheng H., Cheng W. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. – 2017. – arXiv:1704.02971. – DOI: 10.48550/arXiv.1704.02971.
14. Graves A. *Supervised Sequence Labelling with Recurrent Neural Networks*. – Berlin : Springer, 2012. – 326 p. – DOI: 10.1007/978-3-642-24797-2.
15. Wei W. W. S. *Time Series Analysis: Univariate and Multivariate Methods*. – Pearson, 1991. – 245–246 pp. – DOI: 10.2307/2289741.
16. Zenkner G., Navarro-Martinez S. A flexible and lightweight deep learning weather forecasting model // *Applied Intelligence*. – 2023. – Vol. 53, No. 21. – P. 24991–25002. – DOI: 10.1007/s10489-023-04824-w.
17. Shengxiang Lv, Lin Wang, Sirui Wang. A Hybrid Neural Network Model for Short-Term Wind Speed Forecasting // *Energies*. – 2023. – Vol. 16, No. 4. – Art. 1841. – DOI: 10.3390/en16041841.

18. Wu Y., Tan H. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. – 2016. – arXiv:1612.01022. – DOI: 10.48550/arXiv.1612.01022.
19. Vlahogianni E. I., Karlaftis M. G., Golias J. C. Short-term traffic forecasting: Where we are and where we're going // *Transportation Research Part C: Emerging Technologies*. – 2014. – Vol. 43, Part 1. – P. 3–19. – DOI: 10.1016/j.trc.2014.01.005.

References

1. Papastefanopoulos V., Linardatos P., Panagiotakopoulos T., Kotsiantis S. Multivariate Time-Series Forecasting: A Review of Deep Learning Methods in Internet of Things Applications to Smart Cities // *Smart Cities*. – 2023. – Vol. 6, No. 5. – P. 2519–2552. – DOI: 10.3390/smartsities6050114.
2. Mahjoub S., Labdai S., Chrifi-Alaoui L., Marhic B., Delahoche L. Short-Term Occupancy Forecasting for a Smart Home Using Optimized Weight Updates Based on GA and PSO Algorithms for an LSTM Network // *Energies*. – 2023. – Vol. 16, No. 4. – Art. 1641. – DOI: 10.3390/en16041641.
3. Mehmood A., et al. Energy Prediction and Optimization for Smart Homes with Weather Metrics and Appliance Forecasting // *Sensors*. – 2023. – Vol. 23, No. 7. – Art. 3640. – DOI: 10.3390/s23073640.
4. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions // *Future Generation Computer Systems*. – 2013. – Vol. 29, No. 7. – P. 1645–1660. – DOI: 10.1016/j.future.2013.01.010.
5. Zanella A., Bui N., Castellani A. Internet of Things for Smart Cities // *IEEE Internet of Things Journal*. – 2014. – Vol. 1, No. 1. – P. 22–32. – DOI: 10.1109/JIOT.2014.2306328.
6. Makridakis S., Spiliotis E., Assimakopoulos V. Statistical and Machine Learning forecasting methods: Concerns and ways forward // *PLOS ONE*. – 2018. – Vol. 13, No. 3. – e0194889. – DOI: 10.1371/journal.pone.0194889.
7. Hyndman R. J., Athanasopoulos G. *Forecasting: Principles and Practice*. – Melbourne : OTexts, 2021. – 138 pp. – Режим доступу: <https://robjhyndman.com/uwafiles/fpp-notes.pdf>
8. Han J. M., Ang Y. Q., Malkawi A., Samuelson H. W. Using recurrent neural networks for localized weather prediction with combined use of public airport data and on-site measurements // *Building and Environment*. – Vol. 192. – 2021. – 107601. – DOI: 10.1016/j.buildenv.2021.107601.
9. Nketiah E. A., Chenlong L., Yingchuan J., Aram S. A. Recurrent neural network modeling of multivariate time series and its application in temperature forecasting // *PLOS ONE*. – 2023. – Vol. 18, No. 5. – e0285713. – DOI: 10.1371/journal.pone.0285713.
10. Siami-Namini S., Tavakoli N., Siami-Namin A. A Comparison of ARIMA and LSTM in Forecasting Time Series // *Proceedings of the 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. – 2018. – P. 1394 – 1401. – DOI: 10.1109/ICMLA.2018.00227.
11. Tekin S. F., Fazla A., Kozat S. S. Numerical Weather Forecasting Using Convolutional-LSTM With Attention and Context Matcher Mechanisms // *IEEE Transactions on Geoscience and Remote Sensing*. – 2024. – Vol. 62. – 4105313. – DOI: 10.1109/TGRS.2024.3409084.
12. Hou J., Wang Y., Zhou J., Tian Q. Prediction of hourly air temperature based on CNN–LSTM hybrid model // *International Journal of Climatology*. – 2022. – Vol. 13. – P. 1962 – 1986. – DOI: 10.1080/19475705.2022.2102942.
13. Qin Y., Song D., Cheng H., Cheng W. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. – 2017. – arXiv:1704.02971. – DOI: 10.48550/arXiv.1704.02971.
14. Graves A. *Supervised Sequence Labelling with Recurrent Neural Networks*. – Berlin : Springer, 2012. – 326 p. – DOI: 10.1007/978-3-642-24797-2.
15. Wei W. W. S. *Time Series Analysis: Univariate and Multivariate Methods*. – Pearson, 1991. – 245 – 246 pp. – DOI: 10.2307/2289741.
16. Zenkner G., Navarro-Martinez S. A flexible and lightweight deep learning weather forecasting model // *Applied Intelligence*. – 2023. – Vol. 53, No. 21. – P. 24991 – 25002. – DOI: 10.1007/s10489-023-04824-w.
17. Shengxiang Lv, Lin Wang, Sirui Wang. A Hybrid Neural Network Model for Short-Term Wind Speed Forecasting // *Energies*. – 2023. – Vol. 16, No. 4. – Art. 1841. – DOI: 10.3390/en16041841.
18. Wu Y., Tan H. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. – 2016. – arXiv:1612.01022. – DOI: 10.48550/arXiv.1612.01022.
19. Vlahogianni E. I., Karlaftis M. G., Golias J. C. Short-term traffic forecasting: Where we are and where we're going // *Transportation Research Part C: Emerging Technologies*. – 2014. – Vol. 43, Part 1. – P. 3–19. – DOI: 10.1016/j.trc.2014.01.005.