

УДК 004.8

DOI 10.30837/bi.2025.2(103).02

А. Д. Костиюченко¹, В. В. Герасимов²¹ДНУ, м. Дніпро, Україна, kostiuchenko_a@365.dnu.edu.ua,
ORCID iD: 0000-0002-3940-8797²ДНУ, м. Дніпро, Україна, herasymov_v@365.dnu.edu.ua,
ORCID iD: 0000-0002-1366-715X

КОМП'ЮТЕРНІ МОДЕЛІ ПРОГНОЗУВАННЯ ЗНАЧЕНЬ ЧАСОВИХ РЯДІВ

У роботі розглянуто підхід до комп'ютерного моделювання часових рядів на основі сучасних архітектур глибокого навчання, зокрема LSTM, GRU та їхніх гібридних комбінацій, а також ансамблевих моделей. Запропоновано порівняльний аналіз глибоких нейромережових структур різної параметричної складності за метриками MAE, MSE, SMAPE та MAPE на тренувальній, валідаційній та тестовій вибірках. Показано, що гібридні архітектури LSTM+GRU забезпечують кращу якість прогнозування порівняно з окремими моделями, однак подальше нарощування глибини та кількості параметрів призводить лише до незначного приросту точності. Найкращі результати отримано для ансамблевого алгоритму, сформованого на основі кількох різномірних моделей, який демонструє найнижчі значення похибок та підвищену стійкість до шумів і аномалій даних. Результати експериментів підтверджують, що поєднання глибоких рекурентних архітектур із ансамблевими підходами є ефективним інструментом підвищення точності та стабільності прогнозування довгострокових метеорологічних часових рядів і може бути використане як основа для побудови прикладних систем підтримки прийняття рішень в енергетиці, транспорті та інших динамічних галузях.

МОДЕЛЬ ГЛИБОКОГО НАВЧАННЯ, LSTM, GRU, АНСАМБЛЕВА МОДЕЛЬ, ЧАСОВИЙ РЯД

A.D. Kostiuchenko, V.V. Herasymov. Computer models for time series forecasting. The paper presents an approach to computer modelling of time series based on modern deep learning architectures, specifically LSTM, GRU, and their hybrid combinations, as well as ensemble models. A comparative analysis of deep neural network structures with varying parametric complexity is proposed, utilising MAE, MSE, SMAPE, and MAPE metrics on training, validation, and test samples. It is demonstrated that hybrid LSTM+GRU architectures yield better prediction quality compared to individual models; however, further increasing the depth and number of parameters results in only a slight increase in accuracy. The best results were obtained for an ensemble algorithm based on several heterogeneous models, which demonstrates the lowest error values and increased resistance to noise and data anomalies. The results of the experiments confirm that the combination of deep recurrent architectures with ensemble approaches is an effective tool for improving the accuracy and stability of long-term meteorological time series forecasting and can serve as a basis for building applied decision support systems in energy, transportation, and other dynamic industries.

DEEP LEARNING MODEL, LSTM, GRU, ENSEMBLE MODEL, TIME SERIES

Вступ

Упродовж останніх років моделі глибокого навчання зарекомендували себе як важливий інструмент розв'язання цілої низки задач штучного інтелекту. Згорткові нейронні мережі та архітектури, побудовані на основі операції згортки, дозволяють класифікувати, визначати межі та сегментувати об'єкти на зображеннях із майже абсолютною точністю. Останні експерименти свідчать, що сучасні моделі типу ResNet50, VGG19 та InceptionV3 класифікують об'єкти у випадках кількох можливих правильних відповідей навіть краще, ніж людське око. Задачі обробки природної мови також отримали нові підходи до вирішення: архітектури типу GPT, BERT, T5 вже здатні не лише анутовати, перекладати чи переказувати тексти, але й генерувати власні, унікальні відповіді у форматі чат-ботів, що й було покладено в основу всесвітньо відомих ChatGPT, Gemini, Grok та інших. Таке широке впровадження моделей штучного інтелекту неминуче вплинуло й на домен обробки послідовностей, оскільки навіть базові задачі, такі як автоматичний переклад природної мови, за своєю природою є задачами моделювання

залежностей слів у реченнях. У цих сценаріях кожен наступний елемент послідовності залежить від попередніх, що потребує здатності моделі утримувати й опрацьовувати контекст різної довжини.

Прогнозування майбутніх значень процесу на основі його історичних спостережень є фундаментальним інструментом у фінансовому аналізі, енергетиці, метеорології, екологічному моніторингу, демографії, логістиці, медичній діагностиці та багатьох інших сферах, що було розглянуто детально у дослідженнях [1;2]. Моделі прогнозування знаходять широке застосування у задачах передбачення динаміки метеорологічних показників, зокрема температури повітря та атмосферного тиску, що є важливим для оптимізації роботи інфраструктури, чутливої до погодних умов. У сфері енергетики такі моделі використовуються для оцінювання майбутніх обсягів споживання електроенергії, що дає змогу підвищувати ефективність функціонування енергосистем і забезпечувати їхню стабільність. У фінансовому секторі прогнозування параметрів ринку, включно з динамікою цін, рівнем ліквідності та показниками волатильності фінансових активів, є необхідною умовою

для оцінювання ризиків і формування стратегій управління капіталом. У технічних системах прогнозування ймовірності виникнення відмов дозволяє своєчасно виявляти критичні тенденції та запобігати негативним наслідкам, що сприяє зменшенню операційних витрат і підвищує загальну надійність обладнання. У контексті обробки даних, отриманих з датчиків транспортних засобів, такі системи дозволяють відстежувати рівень палива в системі та аномальну роботу двигуна в різних станах, повідомляти про аварійні ситуації, що пов'язані з паливними баками.

Актуальність задач обробки часових рядів також зумовлена тим, що реальні процеси майже ніколи не бувають представлені лінійними або одновимірними залежностями. У багатьох галузях доводиться працювати з мультिवаріантними часовими рядами, де кілька змінних корелюють значною мірою між собою і визначають поведінку системи. У такому домені, як метеорологія, одночасний аналіз температури, вологості, атмосферного тиску та швидкості вітру забезпечує набагато точніше моделювання погодних умов, ніж окремі прогнози кожної змінної. У фінансових ринках взаємозв'язки між індексами, активами та макроекономічними факторами формують складну систему, в якій локальні зміни матимуть значні довгострокові наслідки. Такі системи потребують моделей, здатних уловлювати не лише часові, а й структурні взаємозв'язки між ознаками. Питання впровадження комбінованих архітектур, методів машинного та глибокого навчання, а також статистичного аналізу значною мірою досліджувались у роботах [3-5].

Метою даного дослідження є обґрунтування та емпірична перевірка доцільності використання глибоких нейромережових архітектур типу LSTM, GRU та їхніх гібридів, а також ансамблевих алгоритмів на їх основі для прогнозування значень часових рядів метеорологічних показників. Передбачається встановити, чи забезпечують гібридні архітектури LSTM+GRU та їхній ансамбль статистично значуще покращення якості прогнозування порівняно з окремими моделями, а також визначити компроміс між параметричною складністю та точністю прогнозу.

Виклад основного матеріалу

Відповідно до визначення, поданого у роботі [6], часовий ряд – це послідовність упорядкованих у хронологічному порядку сукупності значень певного статистичного показника, які відображають динаміку певного явища у часі. Елементи послідовності $y_1, y_2, y_3, \dots, y_n$ називають рівнями часового ряду. Індекс $t = 1, 2, 3, 4, \dots, n$ позначає порядковий номер моменту або інтервалу часу, до якого належить відповідний рівень, а величина n визначає довжину часового ряду.

Залежно від характеру часового виміру часові ряди класифікують на моментні та інтервальні. У

моментних рядах кожен рівень фіксує стан явища на конкретний момент часу; типовими прикладами є обсяг золотовалютних резервів на початок року або чисельність зареєстрованих безробітних на початок кварталу. На відміну від цього, рівні інтервальних рядів відображають результат процесу, накопичений протягом певного часового проміжку. Прикладами таких рядів можуть слугувати обсяги виробленої електроенергії за квартал, біоелектричні сигнали організму людини або показники сонячної активності упродовж року.

Структура часових рядів визначається сукупністю характеристик, які описують спосіб організації та внутрішні властивості даних, впорядкованих у часовому вимірі. Часовий ряд $\{x_t\}_{t=1}^T$ розглядається як реалізація стохастичного процесу, де кожне спостереження є результатом дії як детермінованих, так і випадкових компонент. Загальноприйнятою є адитивна або мультиплікативна декомпозиція часових рядів, у межах якої рівень у момент часу t подається як сума або добуток структурних складових. У базовому адитивному варіанті модель представлення рядка записується у вигляді:

$$x_t = m_t + s_t + c_t + \varepsilon_t \quad (1)$$

де m_t – трендова компоненту; s_t – сезонна складова з фіксованим періодом повторення; c_t – циклічні зміни; ε_t – міра впливу стохастичного процесу.

Для оцінки якості прогнозування значень моделей глибокого навчання було використано метрики MSE (англ. – Mean Squared Error) та SMAPE (англ. – Symmetric Mean Absolute Percentage Error), розраховані за формулами (2) і (3).

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (2)$$

де y_i – значення цільової функції; \hat{y}_i – прогнозоване значення; N – кількість елементів у вибірці.

$$SMAPE = \frac{100}{N} \sum_{i=1}^N \frac{2|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}, \quad (3)$$

де y_i – значення цільової функції; \hat{y}_i – прогнозоване значення;

Питання обробки послідовностей динамічно розвивається з кінця минулого сторіччя. Саме тоді було запропоновано архітектуру LSTM (англ. – Long-Short Term Memory), що стала альтернативою використанню стандартних та домінуючих у галузі III стандартних рекурентних нейронних мереж. У статтях [7] та [8] детально описано кроки реалізації.

Архітектура LSTM належить до класу рекурентних нейронних мереж і була розроблена з метою подолання фундаментальних недоліків класичних RNN, пов'язаних зі зниканням і вибухом градієнтів під час поширення похибки у часовому напрямку. Основним конструктивним елементом моделі є комірка пам'яті,

яка забезпечує здатність зберігати інформацію протягом тривалих часових інтервалів завдяки окремому стану пам'яті c_t , структурно відокремленому від прихованого стану h_t . На відміну від традиційних RNN, де інформаційний потік повністю контролювався нелінійною динамікою одного прихованого шару, LSTM впроваджує систему гейтів, що регулюють надходження та видалення інформації, надаючи моделі гнучкість та стійкість при роботі з довгими послідовностями.

Обчислення всередині LSTM-комірки визначається поєднанням сигмоїдальної та гіперболічного тангенсу активаційних функцій, які формують керувані інформаційні канали. Гейт забуття обчислюється через сигмоїдальну функцію активації та визначає частину попереднього стану пам'яті c_{t-1} , що підлягає збереженню. Завдяки множенню цього гейта на c_{t-1} модель здатна контролювати довжину ефективного контексту й адаптуватися до структури часових залежностей. Наступним компонентом є гейт входу, що модулює вплив нової інформації на стан пам'яті.

Паралельно з цим обчислюється нове значення із використанням гіперболічного тангенсу, яке відображає потенційну зміну внутрішнього стану. Комбінація гейта входу та даного нового значення визначає ступінь оновлення c_t , що забезпечує поступове оновлення пам'яті з урахуванням релевантних ознак вхідного сигналу. Після цього формується вихідний гейт, який визначає, яку частину оновленого стану пам'яті буде відображено у прихованому стані h_t . Використання $\tanh(c_t)$ у поєднанні з сигмоїдальним гейтом виходу формує прихований стан, що відображає внутрішню інформацію, необхідну для подальших прогнозів, класифікації або інших завдань.

GRU (англ. – Gated Recurrent Unit) є рекурентною архітектурою, розробленою як компактна та обчислювально ефективніша альтернатива LSTM, з метою зменшення кількості параметрів і спрощення механізмів керування пам'яттю без втрати здатності моделювати довготривалі залежності у часових послідовностях. На відміну від LSTM, яка використовує окрему комірку стану, GRU об'єднує механізми коротко- та довготривалої пам'яті у єдиний прихований стан h_t , а замість трьох гейтів LSTM (забуття, вхідний та вихідний) застосовує лише два, а саме update gate та reset gate. Оновлений підхід робить архітектуру структурно простішою, порівняно простішою у процесі навчання та більш придатною для задач, що передбачають високу швидкість навчання, зменшення складності за пам'яттю чи застосування моделей на обмежених обчислювальних ресурсах.

Механізм роботи GRU ґрунтується на двох гейтах, які керують тим, яку інформацію необхідно зберегти, інтегрувати або відкинути під час переходу від одного часового кроку до іншого. Update gate визначає

ступінь збереження попереднього прихованого стану h_{t-1} та контролює швидкість оновлення внутрішнього представлення. Через використання сигмоїдальної функції активацію update gate набуває значень у діапазоні $[0;1]$, що дозволяє регулювати баланс між новою та попередньою інформацією: значення близьке до 1 зберігає стан пам'яті, тоді як значення, близьке до 0, сприяє її оновленню. Reset gate модулює вплив попереднього стану на формування нового вмісту прихованого шару. Якщо reset gate наближається до 0, модель забуває попередній контекст, що дозволяє сформувати оновлений вектор стану переважно на основі поточного входу x_t . У випадках, коли reset gate наближається до 1, попередній стан включається в обчислення, що дозволяє ефективно моделювати короткі та середньострокові залежності.

Однією з ключових особливостей GRU є те, що вона не має окремого стану комірки, а всі операції здійснюються лише над прихованим станом h_t . Менша кількість параметрів також зменшує ризик перенавчання, що є важливою перевагою у задачах прогнозування часових рядів, де тренувальні вибірки можуть бути обмеженими або мати значну нерівномірність, що було висвітлено у роботах [9] та [10].

Для покращення ефективності прогнозування значень часових рядів було запропоновано реалізувати ансамблевий алгоритм у контексті архітектури LSTM та GRU. Ансамблеві алгоритми посідають центральне місце в сучасному машинному навчанні завдяки можливості поєднувати результати декількох моделей з метою підвищення точності, стійкості та здатності системи до узагальнення. Ансамбль працює ефективніше за окрему модель за умови, що його компоненти є достатньо різномірними та допускають помилки різної природи. Основна ідея ансамблювання полягає у використанні колективного рішення групи моделей, яке мінімізує ймовірність систематичної похибки, властивої кожній окремій моделі, і забезпечує більш стабільну поведінку в ситуаціях шумних, неоднорідних або нестабільних даних.

Процес побудови ансамблевого алгоритму включає кілька послідовних етапів. Першим є вибір множини базових моделей, які повинні бути достатньо різними за архітектурою, функціональною формою або гіперпараметрами. Далі здійснюється незалежне навчання базових моделей на спільному наборі даних або на його різних модифікованих вибірках, що забезпечує різноманітність моделей та мінімізує їхню корельованість. Після цього відбувається побудова механізму об'єднання: для регресійних задач це може бути арифметичне середнє, зважене середнє, медіана або результат метарегресора, для класифікації – голосування чи softmax-агрегація. Важливою складовою є оптимізація ваг у зважених ансамблях, що може здійснюватися методами крос-валідації, стохастичної

оптимізації або байєсівського налаштування гіперпараметрів, що було розкрито у статті [11].

Розглянемо деталізоване створення ансамблю моделей:

1. Підготовка даних. Сформувані навчальну, валідаційну та тестову вибірки, забезпечивши коректний часовий поділ без перемішування. За потреби виконати нормалізацію або стандартизацію часових рядів, а також перетворення у формат ковзних вікон:

$$x_t = (x_{t-w+1}, \dots, x_t), \quad (4)$$

де x_t – значення часового ряду.

2. Вибір множини базових моделей. Задати набір моделей $\{M_1, M_2, \dots, M_K\}$, який включає архітектури різних типів або однакові моделі з різними гіперпараметрами та ініціалізаціями.

3. Налаштування гіперпараметрів базових моделей. Для кожної моделі M_K визначити структуру (кількість шарів та нейронів у кожному, функції активації) та гіперпараметри навчання (оптимізатор, швидкість навчання, розмір батчу, кількість епох, критерії зупинки).

4. Незалежне навчання базових моделей. Для кожної моделі M_K виконати процедуру навчання на одній і тій самій або на модифікованих навчальних даних, фіксуючи її параметри θ_K після досягнення найкращої якості на валідаційній вибірці.

5. Оцінювання якості базових моделей. Для кожної моделі M_K обчислити значення обраних метрик (MAE, MSE, SMAPE) на навчальному, валідаційному та тестовому наборах даних. За результатами оцінки зафіксувати якість моделей та, за потреби, відкинути явно деградуючі моделі.

6. Формування ансамблевих прогнозів на валідації.

Для кожного спостереження валідаційної вибірки отримати вектор прогнозів:

$$\hat{y}_t^{(k)} = M_k(x_t) \quad (5)$$

де x_t – вектор вхідних ознак.

7. Вибір схеми агрегації. Обрати метод комбінування прогнозів базових моделей. Просте середнє:

$$\hat{y}_t^{ens} = \frac{1}{N} \sum_{k=1}^N \hat{y}_t^{(k)} \quad (6)$$

8. Фінальне оцінювання ансамблю. Застосувати обраний механізм агрегації до прогнозів на тестовій вибірці та обчислити метрики, що використовувалися для базових моделей. Порівняти якість ансамблю із найкращою індивідуальною моделлю, фіксуючи досягнутий приріст точності чи стабільності прогнозування.

Для програмної реалізації експериментів дослідження було обрано мову програмування Python та фреймворк машинного навчання TensorFlow. Такий вибір зумовлений широкими можливостями цих інструментів у побудові та навчанні нейронних мереж, зокрема моделей LSTM і GRU, що вже попередньо реалізовано в бібліотеці keras. TensorFlow підтримує GPU та TPU прискорювачі тензорних обчислень, дозволяє контролювати значення метрик під час навчання. Крім того, Python має розвинену екосистему бібліотек, таких як NumPy, pandas, scikit-learn і Matplotlib, що суттєво спрощують підготовку даних, проведення експериментів і візуалізацію результатів, забезпечуючи відтворюваність та гнучкість дослідження.

На рис. 1 наведено лістинг коду, написано для створення та навчання нейронної мережі на основі архітектури LSTM.

```

model_lstm = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(WINDOW, 1)),
    tf.keras.layers.LSTM(32, return_sequences=False),
    tf.keras.layers.Dense(HORIZON)
])

model_lstm.compile(optimizer=tf.keras.optimizers.Adam(1e-3),
                  loss="mse")

save_best_only = tf.keras.callbacks.ModelCheckpoint(
    '/content/results/lstm_model.keras', monitor='val_loss', save_best_only=True
)

logger = tf.keras.callbacks.CSVLogger('/content/results/lstm_training_logs.csv', separator=',')

history = model_lstm.fit(x_train, y_train, validation_data=(x_val, y_val),
                        epochs=30, batch_size=32, verbose=1, callbacks=[save_best_only, logger])

```

... Epoch 1/30
1153/1153 ————— 7s 5ms/step - loss: 0.0109 - val_loss: 1.2827e-04

Рис. 1. Розробка та навчання нейронної мережі на основі LSTM

Набір даних «Weather Long-Term Time Series Forecasting» представляє собою багатовимірний часовий ряд, що містить регулярні метеорологічні вимірювання за тривалий проміжок часу. Він включає широкий набір атмосферних показників: температуру повітря, вологість, швидкість вітру, інтенсивність опадів, радіацію та інші змінні, які дозволяють комплексно описати стан атмосфери [12].

Кожен запис у наборі відповідає конкретному часовому моменту і формує багатовимірний вектор ознак, що забезпечує представлення як одиничних, так і взаємозалежних метеоумов. Така структура дає змогу моделювати як тимчасові тренди, так і сезонні коливання, а також враховувати кореляції між різними атмосферними змінними.

Для підготовки даних типовими кроками є очищення від пропущених або аномальних значень, нормалізація ознак, а також формування вікон для навчання часових моделей. У наборі зберігається 52560 записів, що було розподілено на тренувальну, валідаційну та тестову вибірку у співвідношенні 70%:15%:15%, відповідно.

У табл. 1 наведено архітектуру моделей глибокого навчання, які використовувались для порівняння ефективності навчання за відповідними метриками із ансамблем більш простих базових моделей.

Таблиця 1

Глибокі моделі на базі різних нейромережових архітектур

	Модель №1	Модель №2	Модель №3	Модель №4
Шар мережі	LSTM	GRU	LSTM+GRU	LSTM+GRU
1-й	LSTM(128)	GRU(64)	LSTM(64)	LSTM(128)
2-й	LSTM(64)	GRU(32)	GRU(64)	GRU(128)
3-й	LSTM(32)	Dense(1)	LSTM(32)	LSTM(64)
4-й	Dense(1)		GRU(32)	GRU(64)
5-й			Dense(1)	LSTM(32)
6-й				GRU(32)
7-й				Dense(1)
К-сть параметрів навчання	128417	22305	60641	258785

Модель №1 являє собою класичну багатошарову LSTM-архітектуру, що складається з трьох послідовних LSTM-шарів зі зменшенням кількості елементів на кожному шарі, після чого розміщено вихідний Dense-шар. Модель має 128417 параметрів, що робить її помірно складною та потенційно стійкою до недонавчання.

Модель №2 є компактнішою і базується на двошаровій GRU-структурі, де використано GRU(64) та GRU(32), після чого передбачено фінальний Dense-шар. GRU-архітектура характеризується меншою

кількістю параметрів і швидшим навчанням, що відображено у значно меншій загальній кількості параметрів – лише 22305.

Модель №3 представляє гібридний варіант LSTM та GRU, що поєднує LSTM(64), GRU(64), LSTM(32) і GRU(32) перед вихідним Dense-шаром. Комбінація двох типів рекурентних блоків дає можливість одночасно моделювати як довготривалі, так і короткострокові часові залежності.

Модель №4 є найскладнішою, включаючи LSTM та GRU-шари з більшим числом нейронів (128 і 64), що формує багаторівневу гібридну структуру, а також містить 258785 параметрів навчання.

У табл. 2–4 наведено результати навчання цих моделей у контексті оцінки метрик якості.

Таблиця 2

Оцінка ефективності моделей за метрикою MAE

MAE	Модель №1	Модель №2	Модель №3	Модель №4
train	0,18	0,1618	0,1541	0,148
valid	0,223	0,1343	0,13	0,12
test	0,1438	0,121	0,104	0,1

Модель №3, яка поєднує LSTM і GRU, демонструє стабільне покращення якості: її MAE на тестовій вибірці становить 0,104, що вказує на здатність гібридної архітектури краще узагальнювати складні часові залежності. Найнижчу похибку отримано для моделі №4, кількість параметрів якої в 4 рази більша у порівнянні з моделлю №3. З одного боку, це означає, що збільшення кількості шарів і використання більш глибокої комбінації LSTM+GRU позитивно впливає на точність прогнозування. Однак також варто зазначити, що збільшення кількості шарів та нейронів шару сприяло лише незначному покращенню результатів у порівнянні з менш глибокою архітектурою.

Таблиця 3

Оцінка ефективності моделей за метрикою MSE

MSE	Модель №1	Модель №2	Модель №3	Модель №4
train	0,063	0,054	0,052	0,05
valid	0,073	0,036	0,035	0,032
test	0,034	0,027	0,023	0,021

Модель двошарової GRU показує суттєве зниження MSE, зокрема на валідації та тесті, що підтверджує ефективність GRU у компактніших конфігураціях. Модель №3, що поєднує LSTM і GRU, демонструє подальше покращення результатів, досягаючи MSE рівному 0,023 на тестовій вибірці. Поглиблена комбінована модель LSTM+GRU демонструє також незначно покращені результати відносно неглибокої комбінованої моделі.

Таблиця 4
Оцінка ефективності моделей за метрикою SMAPE

SMAPE, %	Модель №1	Модель №2	Модель №3	Модель №4
train	3,8	3,35	2,96	2,82
valid	2,13	2,02	1,892	1,752
test	13,14	10,35	8,15	8,13

На етапі тренування найвищу похибку демонструє модель №1 (3,8%), тоді як моделі №2–№4 характеризуються нижчими значеннями, що свідчить про кращу здатність GRU і гібридних архітектур адаптуватися до даних. На валідаційній вибірці різниця між моделями посилюється: SMAPE моделі №1 становить 2,13%, тоді як моделі №3 та №4 показують суттєво нижчі значення (1,892% та 1,752% відповідно), що вказує на їхню вищу здатність до узагальнення.

Найбільш показовими є результати на тестовій вибірці, де SMAPE моделі №1 сягає 13,14%, що вказує на значну нестабільність та нижчу точність у прогнозуванні нових даних. Моделі №2 та №3 демонструють суттєве зменшення похибки у межах 10,35% та 8,15% відповідно. Модель №4 показує найнижче значення серед усіх архітектур (8,13%), хоча її перевага над моделлю №3 незначна. Загалом результати свідчать, що гібридні архітектури LSTM+GRU забезпечують найкращу якість прогнозування згідно зі SMAPE, особливо на реальних, невідомих даних.

Таблиця 5
Глибокі моделі на базі різних нейромережевих архітектур

	Модель №1	Модель №2	Модель №3	Модель №4
Шар мережі	LSTM	LSTM	GRU	GRU
1-й	LSTM(32)	LSTM (32)	GRU(32)	GRU(32)
2-й	Dense(1)	Dense(1)	Dense(1)	Dense(1)
К-сть параметрів навчання	13157	13157	10181	10181

У порівнянні з попередніми моделями, зокрема гібридними архітектурами, які містили до 258 тисяч параметрів, такі спрощені моделі мають у 20–25 разів нижчу параметричну потужність. Вони демонструватимуть значно вищу швидкість навчання і менший ризик перенавчання, проте водночас будуть менш здатними моделювати складні, нелінійні та довгострокові залежності в часових рядах. З огляду на це, такі моделі можуть бути придатними для таких експериментів, як побудова ансамблів попереднього аналізу або задач із простою динамікою.

Таблиця 6
Оцінка ефективності ансамблю моделей за метриками якості

	MAE	MSE	MAPE, %
train	0,145	0,04	2,76
test	0,0953	0,021	7,66
validation	0,116	0,031	1,7

Наведена таблиця демонструє, що ансамблевий алгоритм забезпечує найкращі результати серед усіх протестованих підходів, що підтверджується низькими значеннями MAE, MSE та MAPE на всіх етапах оцінювання. Значення MAE = 0,0953 та MSE = 0,021 на тестовій вибірці нижчі, ніж у будь-якої з індивідуальних моделей, включно з гібридними архітектурами. Це свідчить про вищу здатність ансамблю до узагальнення та його стійкість до шумів, локальних аномалій та коливань у часовому ряду. Аналогічна ситуація спостерігається і на тестовій вибірці, де ансамбль демонструє найнижчий рівень MAPE (7,6%), що вказує на високу точність відносних прогнозів.

Перевага ансамблю є закономірною, оскільки він поєднує властивості різних моделей, кожна з яких виявляє різні типи часових залежностей. LSTM більш ефективно моделює довготривалі патерни, тоді як GRU швидше реагує на короткострокові зміни та володіє кращою узагальнювальністю за меншої кількості параметрів. Гібридні моделі поєднують ці переваги, але все одно залишаються обмеженими своєю архітектурою. Ансамбль дозволяє інтегрувати прогнози всіх архітектур, що зменшує дисперсію помилок та компенсує індивідуальні слабкі сторони кожної моделі.

Крім того, ансамбль мінімізує ризик перенавчання, оскільки помилки окремих моделей мають різну природу і з малою ймовірністю збігаються на одних і тих самих прикладах. Завдяки цьому агрегований прогноз стає згладженим та більш стабільним. Наявність значного покращення на тестовій вибірці підтверджує, що ансамблевий підхід краще відтворює реальну поведінку часового ряду та має кращі здатності до прогнозування в умовах, коли окремі моделі можуть допускати систематичні або специфічні для архітектури помилки.

Висновки

Одним із ключових результатів є демонстрація того, що архітектури LSTM та GRU, незважаючи на спільну належність до класу рекурентних нейронних мереж, виявляють різні рівні чутливості до структури часових залежностей. LSTM краще зберігає довгострокові шаблони, тоді як GRU завдяки своїй компактності та меншій кількості параметрів виявляє високу ефективність у моделюванні короткострокової динаміки та швидше пристосовується під час навчання. Гібридні архітектури, що комбінують LSTM та GRU, показали здатність успадковувати переваги обох підходів, а результати саме гібридних моделей продемонстрували істотне зниження MAE, MSE та SMAPE порівняно з окремими моделями.

Однак, попри те що найглибші з протестованих моделей у певних випадках досягали кращих результатів, спостерігалось і те, що збільшення параметричності

складності не завжди дає пропорційний приріст точності. Зокрема, найглибша комбінована модель хоча й демонструвала найнижчі значення похибок серед окремих архітектур, однак покращення порівняно з менш глибокою гібридною моделлю було незначним. Такий результат свідчить, що для задач прогнозування часових рядів надмірне збільшення кількості параметрів може не призвести до відповідного зростання якості прогнозу, особливо коли дані містять шум, варіації або нестійкі компоненти.

Найсуттєвіший приріст точності було отримано за рахунок застосування ансамблевого підходу. Ансамбль моделей продемонстрував значно менші значення MAE, MSE та MAPE порівняно з усіма індивідуальними моделями, включно з гібридними архітектурами. Такий результат є цілком очікуваним з огляду на природу ансамблевих алгоритмів, які зменшують дисперсію помилок шляхом агрегування різних прогнозів, тим самим згладжуючи систематичні похибки окремих моделей.

Отримані результати свідчать не лише про переваги ансамблів над окремими моделями, але й про здатність простих архітектур діяти як ефективні складові ансамблю. Незважаючи на те, що моделі з мінімальною глибиною та кількістю параметрів демонструють суттєво гірші результати у порівнянні з глибокими нейромережами в індивідуальному режимі, вони роблять значимий внесок у підсумковий ансамблевий прогноз. Таким чином, ефективність ансамблю залежить не лише від складності базових моделей, але й від їхньої здатності робити незалежні помилки, що знижує корельованість прогнозів.

Отримані результати вписуються у загальну тенденцію сучасних досліджень, відповідно до якої ансамблеві підходи є широковживаним підходом до підвищення точності прогнозування в задачах машинного навчання. Такі методи особливо важливі для моделювання часових рядів, де дані характеризуються складною структурою, нелінійністю та наявністю кількох компонент, серед яких можна відзначити тренд, сезонність, циклічність. Ансамблеві підходи не лише підвищують точність прогнозування, але й забезпечують стійкість до нестабільності даних, що критично важливо для галузей, де помилка прогнозу може мати значні економічні або технічні наслідки.

Загалом проведене дослідження дозволяє стверджувати, що ансамбль нейронних моделей на основі LSTM та GRU є ефективним і доцільним підходом

для прогнозування часових рядів, значно перевершує індивідуальні архітектури та демонструє високу здатність до узагальнення. У майбутніх роботах планується дослідження методів оптимізації вагових коефіцієнтів ансамблю, застосування стохастичного вибору моделей, оцінка доцільності впровадження авторегресивних методів статистичного аналізу, використання моделей на базі архітектури Transformer з метою покращення узагальнюючої здатності комп'ютерних моделей.

Список літератури

- [1] Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194), 20200209.
- [2] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3), e0194889.
- [3] Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159-175.
- [4] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). A comparative analysis of forecasting financial time series using arima, lstm, and bilstm. *arXiv preprint arXiv:1911.09512*.
- [5] Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194), 20200209.
- [6] Єріна, А., & Мазуренко, О. (2022). Статистичний аналіз часових рядів: навчальний посібник. Київ: КНУШ.
- [7] Vennerød, C. B., Kjærø, A., & Bugge, E. S. (2021). Long short-term memory RNN. *arXiv preprint arXiv:2105.06756*.
- [8] Sagheer, A., & Kotb, M. (2019). Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*, 323, 203-213.
- [9] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- [10] Pirani, M., Thakkar, P., Jivrani, P., Bohara, M. H., & Garg, D. (2022, April). A comparative analysis of ARIMA, GRU, LSTM and BiLSTM on financial time series forecasting. In *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)* (pp. 1-6). IEEE.
- [11] Choi, J. Y., & Lee, B. (2018). Combining LSTM network ensemble via adaptive weighting for improved time series forecasting. *Mathematical problems in engineering*, 2018(1), 2470171.
- [12] King, A. (2023). Weather Long-Term Time Series Forecasting [Data set]. Kaggle. <https://www.kaggle.com/datasets/alistairking/weather-long-term-time-series-forecasting>

Надійшла до редколегії 24.09.2025