

<https://doi.org/10.31891/2307-5732-2026-361-9>
УДК 004.032.26:519.246:517.9

КОСОВАН ГРИГОРІЙ

Чернівецький Національний Університет ім. Юрія Федьковича
<https://orcid.org/0000-0002-3351-3852>
e-mail: g.kosovan@chnu.edu.ua

МЕЛЬНИК ВЛАДИСЛАВ

Чернівецький Національний Університет ім. Юрія Федьковича
<https://orcid.org/0009-0004-9847-2635>
e-mail: melnyk.vladyslav@chnu.edu.ua

ОЦІНКА ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ МОДЕЛЕЙ GRU ТА LSTM ДЛЯ ПРОГНОЗУВАННЯ ПОВЕДІНКИ СИСТЕМИ ЛОРЕНЦА ПІД ЧАС ГЕНЕРУВАННЯ ПОСЛІДОВНОСТЕЙ ЧИСЕЛЬНИХ ЗНАЧЕНЬ

У статті представлено порівняльний аналіз можливості застосування нейронних мереж, а саме рекурентних моделей Gated Recurrent Units (GRU) та Long Short-Term Memory (LSTM) з точки зору їхньої здатності передбачати поведінку хаотичних систем при генеруванні числових значень. Основною метою є використання можливостей розпізнавання образів, притаманних нейронним мережам, для здійснення точних прогнозів поведінки хаотичної системи Лоренца при побудові на її основі телекомунікаційних систем зв'язку. У сфері телекомунікаційних систем передбачення вихідних значень хаотичної системи допомагає в оптимізації мережі, запобіганні збоїв, підвищенні пропускної здатності, а також при побудові систем зв'язку із розширенням спектру. GRU та LSTM - дві популярні архітектури рекурентних нейронних мереж для послідовної обробки даних. GRU має простішу структуру і навчається набагато швидше, але LSTM має складнішу структуру, що забезпечує їй здатність запам'ятовувати інформацію на довший період. В даній роботі моделі GRU та LSTM були реалізовані за допомогою мови програмування Python та навчені прогнозувати можливі вихідні значення багатовимірної хаотичної системи Лоренца та оцінка можливості застосування нейронних мереж для покращення відновлення інформаційних потоків на приймальній стороні системи зв'язку із розширенням спектру. Оцінювання цих моделей проводилося за допомогою метрик середньої квадратичної похибки (MSE), середньої абсолютної похибки (MAE) та коефіцієнта детермінації, а також представлено результати тестування на відповідних графіках. Результати цього дослідження демонструють високу ефективність моделей у прогнозуванні поведінки хаотичної системи.

Ключові слова: нейронна мережа, GRU, LSTM, хаотична система.

KOSOVAN HRYHORIY, MELNYK VLADYSLAV

Chernivtsi National University named after Yuriy Fedkovych

EVALUATION OF THE EFFECTIVENESS OF USING GRU AND LSTM MODELS FOR PREDICTING THE BEHAVIOR OF THE LORENTZ SYSTEM WHEN GENERATING SEQUENCES OF NUMERICAL VALUES

This article presents a comparative analysis of the applicability of neural networks, namely recurrent models Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM), in terms of their ability to predict the behavior of chaotic systems when generating numerical values. The main goal is to use the pattern recognition capabilities inherent in neural networks to make accurate predictions of the behavior of the chaotic Lorenz system when building telecommunications systems based on it. In the field of telecommunications systems, predicting the output values of a chaotic system helps to optimize the network, prevent failures, increase bandwidth, and build communication systems with extended spectrum. GRU and LSTM are two popular architectures of recurrent neural networks for sequential data processing. GRU has a simpler structure and learns much faster, but LSTM has a more complex structure, which gives it the ability to remember information for a longer period. In this work, the GRU and LSTM models were implemented using the Python programming language and trained to predict the possible output values of the multidimensional chaotic Lorenz system and to evaluate the possibility of using neural networks to improve the recovery of information flows on the receiving side of the communication system with spectrum expansion. These models were evaluated using the mean square error (MSE), mean absolute error (MAE), and coefficient of determination metrics, and the test results are presented in the corresponding graphs. The results of this study demonstrate the high efficiency of the models in predicting the behavior of a chaotic system.

Keywords: neural network, GRU, LSTM, chaotic system.

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/)

© Косован Григорій, Мельник Владислав

Постановка проблеми

Теорія хаосу [1] є перспективним напрямком досліджень у прикладній математиці, який займається вивченням хаотичних систем. Динамічні системи визначаються як системи, стани яких змінюються з часом відповідно до певних правил або законів. Сучасна теорія динамічних систем [2] бере свій початок у 19 столітті, коли були сформульовані фундаментальні питання, що стосуються стабільності та еволюції Сонячної системи. Динамічні системи [3], які демонструють хаотичну поведінку, повсюдно зустрічаються як у природному, так і в науковому контекстах. Ці системи характеризуються високою чутливістю до початкових умов і складною, непередбачуваною поведінкою. Спектр областей, в яких вони зустрічаються, різноманітний. У телекомунікаційних системах вони використовуються для оптимізації мережевих ресурсів, забезпечення

безпеки мережі, побудова мереж зв'язку з прямим розширенням спектру, прогнозування навантаження, управління перевантаженнями, підвищення якості обслуговування і моніторингу в режимі реального часу. Широкий спектр потенційних застосувань хаотичних систем зумовив давній науковий інтерес до цієї галузі досліджень. Дуже важливо мати можливість розуміти і передбачати поведінку цих систем.

У сучасну епоху спостерігається значний прогрес у галузі машинного навчання [4], завдяки якому можна розробляти складні моделі для обробки величезних обсягів даних і робити точні прогнози. Нейронні мережі добре підходять для прогнозування числових рядів з підвищеною точністю [5, 6]. Одним з найбільш перспективних напрямків застосування машинного навчання є використання нейронних мереж (НМ) [7] для моделювання хаотичних систем. Нейронна мережа - це математична модель, яка працює за принципом людського мозку. Вона навчається моделювати процеси на основі застосованих вхідних і вихідних значень, а продуктивність покращується, коли для навчання моделей використовується більше даних. Звичайні нейронні мережі не мають можливості аналізувати події, що відбулися в минулому, але рекурентні нейронні мережі намагаються вирішити цю проблему.

Рекурентні нейронні мережі [8] (RNN) - це категорія штучних нейронних мереж, призначених для обробки послідовних даних. Особливістю цього типу нейронних мереж є наявність зворотного зв'язку, що дозволяє їм зберігати інформацію. Рекурентні нейронні мережі можна використовувати для аналізу та прогнозування часових рядів. На відміну від прямих мереж, які обробляють інформацію за один прохід, рекурентні мережі обробляють дані протягом декількох часових кроків, що робить їх особливо придатними для моделювання та обробки тексту, мови та часових рядів.

Використання нейронних мереж для прогнозування поведінки хаотичних систем є актуальною задачею. За допомогою нейронних мереж можна сформувати банк даних на основі різних комбінацій початкових умов та параметрів систем. Отримані бази знань можуть бути використані для відновлення початкових умов та параметрів хаотичних систем на основі порівняння з отриманим хаотичним сигналом в телекомунікаційній системі, що в подальшому дозволить покращити ситуацію з розпізнаванням інформаційних повідомлень в телекомунікаційних системах із розширенням спектру.

Теоретичне підґрунтя та пов'язані роботи

Наразі є багато публікацій щодо застосування нейронних мереж в різних сферах. В [5] автори застосовували моделі нейронних мереж LSTM та (GRU) в аналізі даних часових рядів. Моделі LSTM та GRU навчалася з використанням даних, змінюючи випадкові початкові або статичні початкові параметри, щоб знайти оптимальні умови, що призводить до кращої точності моделі. Крім того, був продемонстрований вплив налаштування гіперпараметрів на продуктивність моделей та порівнювався час навчання, необхідний для кожної архітектури.

В [6] автори продемонстрували можливість застосування нейронних мереж для прогнозування фінансових показників. У дослідженні було вперше сформульовано проблему прогнозування часових рядів та її математичні основи. Після цього надається опис найпопулярніших архітектур глибокого навчання, що використовуються на сьогоднішній день з успіхом у прогнозуванні часових рядів, з акцентом як на їхні переваги, так і на недоліки. Особливу увагу приділяють мережам прямого зв'язку, рекурентним нейронним мережам (таким як мережі Елмана), довготривалій та короткочасній пам'яті (LSTM) та ґратованим рекурентним одиницям (GRU). Крім того, переваги методу дерева бустування XGBoost.

Авторами в [9] продемонстровано можливість застосування моделі глибокого навчання для визначення оптимальної моделі, що може передбачати споживання енергії в довгостроковій перспективі. У роботі автори розробляли ефективну прогностичну модель з меншою кількістю помилок у тестах прогнозування. Для цього порівнювались модель із стробованим рекурентним блоком (GRU) та довгострокову пам'ять (LSTM) та визначалась оптимальна модель за кількома показниками, таких як R-квадрат, мінімальна середньоквадратична помилка (MSE), середня абсолютна помилка (MAE), середньоквадратична помилка (RMSE).

В телекомунікаціях хаотичні системи часто використовуються для побудови систем із прямим розширенням спектру [10]. Їх застосування дозволяє підвищити надійність систем зв'язку. В той же час існує проблема з детектуванням сигналів в системах із розширенням спектру. Одним із напрямків вирішення цієї проблеми це застосовувати нейронні мережі для прогнозування поведінки хаотичної системи, і як наслідок можливості розпізнавання інформаційного повідомлення на приймальній стороні [11].

Нейронна мережа - це сукупність нейронних елементів та зв'язків між ними. Первинним компонентом нейронної мережі є штучний нейрон, який виконує нелінійне перетворення суми добутків вхідних сигналів на вагові коефіцієнти. Під час роботи нейрон одночасно отримує багато вхідних сигналів, де вхідна інформація проходить через ваги (параметри), значення яких визначають поведінку нейрона, а потім результат надсилається на єдиний вихід. Найпростішою формою штучної нейронної мережі є одношаровий перцептрон (рис.1).

На рис. 1: 1 - нейрони, вихідні сигнали яких надходять на вхід цього перцептрона (ω_i - ваги вхідних сигналів), 2 - суматор вхідних сигналів, 3 - обчислювач передатної функції, 4 - нейрони, на входи яких подається сигнал з виходу цього перцептрона.

Одношаровий перцептрон може бути розширений до багатшарового перцептрона. У такій конфігурації присутні кілька шарів, кожен з яких складається з певної кількості нейронів.

Звичайні нейронні мережі не здатні аналізувати події, що відбулися в минулому. Рекурентні нейронні мережі (RNN) були розроблені для подолання цього обмеження.

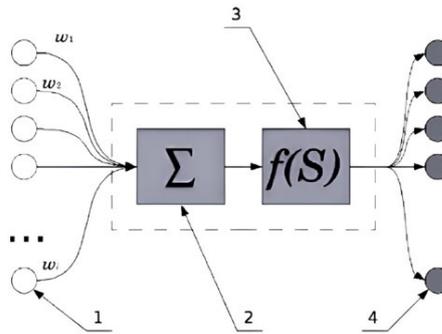


Рис. 1. Схема одношарового перцептрона

RNN призначені для обробки даних, які надходять послідовно, включаючи часові ряди, текст, аудіо та відео. Рекурентну нейронну мережу можна уявити як послідовність одношарових елементів, які передають отриманий сигнал наступній копії (рис. 2).

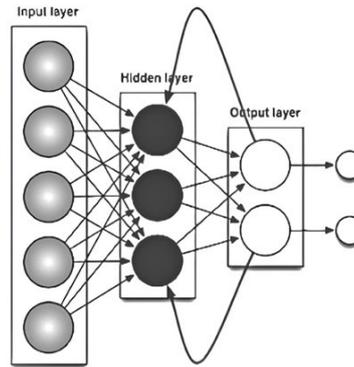


Рис. 2. Схема логіки рекурентної нейронної мережі

На відміну від традиційних нейронних мереж, де всі входи і виходи є незалежними один від одного, цю проблему вирішили рекурентні нейронні мережі, які використовують прихований шар. Основним і найважливішим атрибутом RNN є його прихований стан, який зберігає інформацію, пов'язану з послідовністю даних. Цей стан також називають станом пам'яті, оскільки він накопичує дані про попередні входи мережі. Завдяки цьому RNN здатна враховувати контекст у послідовності.

Особливістю RNN є використання однакових параметрів для кожного входу, оскільки мережа виконує однакові операції на всіх етапах послідовності. Це значно зменшує кількість параметрів, на відміну від багатьох інших типів нейронних мереж, що сприяє простоті та узагальненості моделі. Наприклад, завдання передбачення наступного слова в тексті вимагає інформації про попередні слова, яку мережа зберігає в прихованому стані, забезпечуючи врахування контексту.

Основна проблема з рекурентними нейронними мережами (RNN) може бути пов'язана з явищем, відомим як проблема зникаючого градієнта. Це явище стосується швидкої втрати інформації в мережі з плином часу. Це явище впливає лише на ваги, а не на стан нейронів, проте інформація в них накопичується.

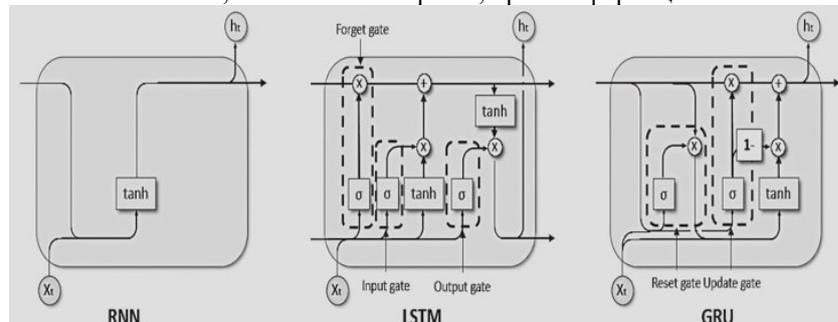


Рис. 3. Схема клітин RNN, LSTM та GRU

Рекурентні нейронні мережі (RNN) існують у багатьох різних формах, кожна з яких підходить для певного типу завдань. До них відносяться стандартні або «ванільні» RNN, двонаправлені RNN, архітектури, покращені механізмами уваги, і більш просунуті моделі, такі як мережі з довгостроковою і короткостроковою пам'яттю (LSTM) і закритими рекурентними одиницями (GRU).

Однак у цьому дослідженні ми зосередимося саме на LSTM та GRU. Ці моделі чудово справляються з довготривалими залежностями даних, що робить їх ефективними інструментами для подолання основних обмежень стандартних RNN.

LSTM (long short-term memory - довга короткочасна пам'ять) - це тип рекурентної нейронної мережі, яка здатна вивчати довгострокові залежності. Вперше LSTM була представлена Хохрайтером і Шмідхубером у 1997 році.

Мережа довготривалої пам'яті була розроблена для зберігання інформації протягом тривалих періодів часу та отримання її з оптимальною ефективністю, коли це необхідно. Фундаментальна концепція полягала в інтеграції набору керуючих елементів, які називаються «ворітьми», в структуру мережі, що дозволяє регулювати потік інформації і обходити проблему градієнтного згасання.

Комірка LSTM складається з наступних компонентів:

- Вхідні ворота: визначають ступінь, до якого нові вхідні дані повинні бути включені в стан комірки.
- Вентиль забування визначає, яка інформація буде відкинута зі стану комірки.
- Вихідний вентиль визначає, яка частина стану комірки буде виведена назовні.

Мережа довготривалої короткочасної пам'яті (LSTM) спочатку була розгорнута в ряді областей, де потрібно було враховувати довготривалі залежності в наборах даних. Одним з перших помітних застосувань LSTM було розпізнавання рукописного тексту.

З часом LSTM зазнали різних модифікацій і вдосконалень. Наприклад, було запропоновано кілька варіантів, що оптимізують оригінальну модель, таких як GRU, яка має менше параметрів, але зберігає багато основних характеристик LSTM.

GRU (Gated Recurrent Unit) - це спрощений тип рекурентної нейронної мережі, запропонований Kyunghyun Cho у 2014 році. GRU є покращенням звичайних RNN і мають багато спільного з LSTM, але мають простішу архітектуру та меншу кількість параметрів. Як результат, ГРУ забезпечує підвищену обчислювальну ефективність, зберігаючи при цьому здатність обробляти довготривалі залежності.

Ключовою особливістю GRU є відсутність окремої комірки пам'яті, як у LSTM. Замість цього, прихований стан оновлюється безпосередньо. Шлюз оновлення відповідає за те, скільки попереднього прихованого стану слід зберігати. Він об'єднує інформацію з попереднього стану і нового входу, дозволяючи мережі зберігати важливу інформацію з минулого.

Шлюз скидання визначає, яку частину попереднього стану слід враховувати для поточного входу. У поєднанні зі шлюзом оновлення це дозволяє GRU динамічно регулювати потік інформації через мережу в залежності від контексту.

Для роботи GRU виконує кілька математичних операцій, зокрема:

- Сигмоїдна функція для управління шлюзом (оновлення та скидання).
- Гіперболічний тангенс для створення нових прихованих кандидатів на стан.

Ця гнучка та спрощена архітектура дозволяє GRU ефективно навчатися та застосовувати його для широкого спектру задач, таких як аналіз послідовностей, прогнозування часових рядів та розпізнавання мови.

Нами були вибрані саме GRU та LSTM моделі через ряд їх переваг в порівнянні з іншими. Класичні (прості) RNN мають суттєвий недолік — проблему зникаючого градієнта (vanishing gradients), що виникає під час зворотного поширення помилки (backpropagation). Це унеможливило запам'ятовування інформації на довгі проміжки часу.

Подолати проблему зникаючого градієнта можна завдяки використанню спеціальних "вентилів" (gates) — механізмів управління потоком інформації. Моделі GRU та LSTM з цим успішно справляються, а також вони можуть вибірково зберігати або відкидати інформацію. Це дозволяє їм "пам'ятати" важливі дані протягом сотень або навіть тисяч часових кроків, а також виконувати ефективне захоплення довгострокових залежностей. Ці особливості є критичними для довготривалого прогнозування.

Експериментальна методологія

Для проведення експериментального дослідження нейронних мереж нами було обрано хаотична система Лоренца, що часто застосовується в якості генератора псевдовипадкових значень для систем зв'язку на основі хаосу.

Едвард Нортон Лоренц – американський математик і метеоролог, який створив теоретичну основу прогнозування погоди і клімату, а також основу для комп'ютеризованої фізики атмосфери і метеорології. У 1963 році Лоренц вивів систему з трьох нелінійних диференціальних рівнянь для моделювання атмосферної конвекції.

Система Лоренца визначається часовою еволюцією змінних x , y та z як:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z \end{cases} \quad (1)$$

У контексті системи Лоренца параметри мають конкретні фізичні інтерпретації:

- σ , відоме як число Прандтля, пов'язане з в'язкістю рідини.
- ρ , відоме як число Релея, пов'язане з різницею температур, що зумовлює конвекцію.
- β - параметр, пов'язаний з геометрією конвекційних комірок

Типові значення, що використовуються в системі Лоренца: $\sigma=10$, $\beta= 8/3$, і $\rho=28$.

Параметри системи можуть бути змінені, що впливатиме на форму кінцевої траєкторії стану системи. Початкові умови можуть бути обрані досить довільно, але рекомендується в межах (0; 10]. В процесі дослідження нами були вибрані типові значення параметрів керування та початкові умови зі значенням 1.0.

Система є спрощеним представленням рівнянь, що описують рух рідини внаслідок різниці температур, сформульованим таким чином, щоб можна було продемонструвати ряд динамічних характеристик.

Модель проста, але потужна, демонструє складну і непередбачувану поведінку, що робить її ідеальним кандидатом для вивчення хаосу і передбачуваності. Система Лоренца, нарижний камінь у вивченні хаотичних систем, може бути ефективно візуалізована та проаналізована за допомогою реалізації на Python.

Щоб спрогнозувати систему Лоренца за допомогою рекурентних нейронних мереж (RNN), потрібно виконати п'ять кроків:

1. Представити рівняння Лоренца мовою Python.
2. Генерування даних з системи Лоренца.
3. Підготовка даних для RNN.
4. Визначення моделі RNN.
5. Навчання моделі.
6. Використання навченої моделі для прогнозування.

У Python є різноманітні бібліотеки, які можуть допомогти з прогнозуванням. У цьому випадку будуть використані: NumPy - для обчислень; SciPy - для розв'язання диференціальних рівнянь; Matplotlib - для побудови графіків даних; TensorFlow - бібліотеки глибокого навчання для побудови та навчання RNN моделей.

Написання коду починається з визначення диференціальних рівнянь у функції з назвою `lorenz_system`. Ця функція обчислює швидкість зміни змінних системи на основі поточного стану та заданих параметрів. Наступним етапом є генерація даних, необхідних для моделі навчання та прогнозування. Для розв'язання рівнянь Лоренца та генерації часових рядів використовується функція `odeint`. Результуючий набір даних часових рядів складається з трьох змінних (x, y, z).

Наступним етапом є підготовка даних для використання в моделі прогнозування. Оскільки RNN обробляють послідовності, важливо готувати дані послідовно, причому кожна послідовність складається з вікна часових кроків. Далі йде нормалізація даних за допомогою `MinMaxScaler`, який вибирає і нормалізує значення для масштабування. Це забезпечує узгодженість моделей машинного навчання та підвищує ефективність навчання.

Наступним кроком є визначення моделі RNN. Ця дія полегшить використання TensorFlow, бібліотеки програмного забезпечення з відкритим вихідним кодом для машинного навчання та штучного інтелекту. Вона пропонує всеосяжну і гнучку екосистему інструментів і бібліотек, які можуть полегшити розробку додатків на основі ML. У цьому прикладі будуть використані моделі GRU та Dense. Рівень GRU складається з 50 блоків з функцією активації «tanh», яка обробляє послідовності з 50 часових кроків з однією характеристикою на кожному кроці. GRU є ефективними для виявлення залежностей у послідовних даних. Шар Dense приймає вихідні дані від GRU і прогнозує наступне значення в серії, створюючи єдиний числовий прогноз на основі вивчених закономірностей. Потім модель на основі GRU тренується на навчальному наборі для 30 епох з розміром партії 32, тоді як тестовий набір використовується для перевірки. На наступному етапі обчислюються похибки, які будуть використані для оцінки продуктивності моделі. Ми використовуємо середню абсолютну похибку (MAE) та точність у межах допуску. Це дасть нам уявлення про те, наскільки добре модель здатна передбачати дані Лоренца

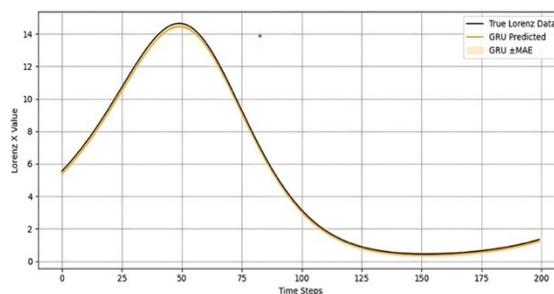


Рис. 4. Порівняння передбачених GRU та істинних значень Лоренца по змінній x

На рис. 4 наведено графік, що відображає передбачення моделі GRU у порівнянні з істинними даними системи Лоренца. Чорна лінія показує фактичні значення системи Лоренца, тоді як помаранчева лінія показує передбачені значення моделі GRU. Заштрихована область представляє межі середньої абсолютної похибки (MAE), що ілюструє невизначеність прогнозу.

Результати експерименту

Для оцінки ефективності роботи нейронних мереж необхідно порівняти результати їх роботи.

Як згадувалося раніше, ці моделі мережі GRU та LSTM є вдосконаленими типами рекурентних нейронних мереж (RNN), призначених для вирішення проблеми зникнення градієнта, поширеної в традиційних RNN. Хоча вони мають певну схожість. Найбільша відмінність між ними полягає в їхній архітектурі та операційних механізмах, що призводить до різних характеристик продуктивності в різних програмах. На рис. 3 показані різні за структурою механізми. Для того, щоб почати порівняльний аналіз продуктивності цих

моделей, спочатку необхідно впровадити певні модифікації коду. По-перше, важливо гарантувати відтворюваність результатів. Це надзвичайно важливо для машинного навчання та аналізу даних, оскільки випадкові процеси можуть впливати на ініціалізацію вагових коефіцієнтів, розподіл даних на навчальні та тестові набори та інші аспекти алгоритму.

Щоб почати порівнювати продуктивність цих моделей, потрібно додати деякі зміни в код. По-перше, необхідно переконатися, що результати є відтворюваними. Тому потрібно зробити фіксацію для генераторів випадкових чисел. У цьому випадку це довільне число 42. Наступним кроком є налаштування параметрів Python і TensorFlow. `PYTHONHASHSEED` - допомагає уникнути змін у поведінці обробки даних, які можуть бути опосередковано пов'язані з порядком хешування. `TF_DETERMINISTIC_OPS` - встановлює параметр, який змушує TensorFlow виконувати лише детерміновані (стабільні) операції, навіть якщо вони трохи повільніші, гарантуючи передбачувані результати.

Наступним кроком є додавання самої моделі LSTM, тому давайте оновимо код функції `create_model`. Тепер одна функція підтримує створення моделей на основі LSTM і GRU замість створення окремих функцій для кожного типу. Це робить функцію універсальною, зменшує дублювання коду та полегшує порівняння різних архітектур. Щоб об'єктивно порівняти моделі LSTM і GRU, ми розробили автоматизований процес навчання та оцінки, який передбачає кілька циклів навчання та аналіз результатів. Спочатку для кожної моделі заздалегідь були визначені основні параметри навчання, а саме:

- Кількість епох: забезпечує достатню кількість ітерацій для ефективною оптимізації моделі.
- Розмір партії (batch_size): впливає на швидкість навчання та стабільність процесу градієнтного спуску.
- Кількість повторів (num_trials): дозволяє усереднити результати та мінімізувати вплив випадкових факторів.

Серед цих параметрів змінювалася лише кількість епох (25, 50 і 100), що дозволило отримати оптимальні умови для коректного порівняння моделей.

Далі, щоб реалізувати навчання, ми створили функцію `train_evaluate_model`, яка відповідає за: створення моделі, вимірювання навчального часу та оцінки точності за ключовими метриками. Спочатку формується модель певного типу (LSTM або GRU) із заданою кількістю шарів і необхідним рівнем регуляризації (dropout). Наступним кроком є запис тривалості процесу навчання, що дозволяє оцінити обчислювальну ефективність обраної архітектури. Важливим етапом порівняння є оцінка точності. Для оцінки було обрано три параметри. Середня квадратична помилка (MSE) [12,13] є одним із найпоширеніших показників, які використовуються для оцінки якості регресійної моделі. MSE відображає середнє значення квадратів різниць між прогнозованими та фактичними значеннями. Розраховується за формулою (2)

$$E = \frac{p}{s_0} \cdot \frac{h_0}{h_1}, \quad (2)$$

Де:

- N – кількість точок даних.
- y_i – фактичне значення для i-го спостереження.
- \hat{y}_i – прогнозоване значення для i-го спостереження.

Інтерпретація самого MSE. Невелике значення вказує на те, що модель добре відповідає даним і робить точні прогнози. Велике значення вказує на те, що модель не змогла добре описати залежності в даних і створює значні помилки.

Наступним показником є середня абсолютна похибка (MAE) [14,15], яка відображає середню похибку між прогнозованими та фактичними значеннями. Він показує, наскільки в середньому прогноз моделі відхиляється від фактичних даних, незалежно від напрямку похибки. Розраховується за формулою (3)

$$E = \frac{p}{s_0} \cdot \frac{h_0}{h_1}, \quad (3)$$

Де:

- N – кількість точок даних.
- y_i – фактичне значення для i-го спостереження.
- \hat{y}_i – прогнозоване значення для i-го спостереження.
- $|y_i - \hat{y}_i|$ є модулем різниці між прогнозним і фактичним значенням.

Інтерпретація результатів така ж, як і в MSE: чим менше значення, тим краще модель справляється з прогнозом.

Коефіцієнт детермінації (R^2) [16, 17]: демонструє, наскільки добре модель враховує залежності в даних і пояснює варіацію цільової змінної. Розраховується за формулою (4)

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (4)$$

Де:

- y_i – фактичне значення для i-го спостереження.
- \hat{y}_i – прогнозоване значення для i-го спостереження.
- \bar{A} – є середнім значенням реальних спостережень.
- $\sum (y_i - \bar{y})^2$ – це залишкова сума квадратів (RSS), яка представляє помилки моделі.

- $\sum (y_i - \bar{y})^2$ – це загальна сума квадратів (TSS), що показує загальну варіацію цільової змінної.

Він ідеально підходить для оцінки загальної ефективності регресійних моделей і порівняння їх продуктивності.

Після завершення всіх циклів були розраховані середні значення та стандартні відхилення для кожної моделі. Це дозволило нам зробити висновки щодо:

- Точність прогнозів моделі оцінювали за допомогою середньої квадратичної помилки (MSE), середньої абсолютної помилки (MAE) та коефіцієнта детермінації (R^2).
- Ефективність процесу навчання оцінювалася за часом, необхідним для його завершення.
- Стабільність результатів демонструє середньоквадратичне відхилення метрик.

Такий підхід дозволив отримати комплексний набір даних для порівняльного аналізу моделей LSTM та GRU, що включає як точність прогнозування, так і швидкість навчання. Це дозволило об'єктивно оцінити відносну придатність двох моделей для прогнозування хаотичних систем.

На рис. 5–7 представлено детальний порівняльний аналіз моделей LSTM та GRU, які навчалися впродовж 25, 50 та 100 епох. Основна увага зосереджена на ключових показниках ефективності: часі навчання, середній квадратичній похибці (MSE), середній абсолютній похибці (MAE) та коефіцієнті детермінації (R^2). Отримані результати дозволяють оцінити переваги та обмеження кожної моделі з огляду на точність прогнозів і швидкість навчання.

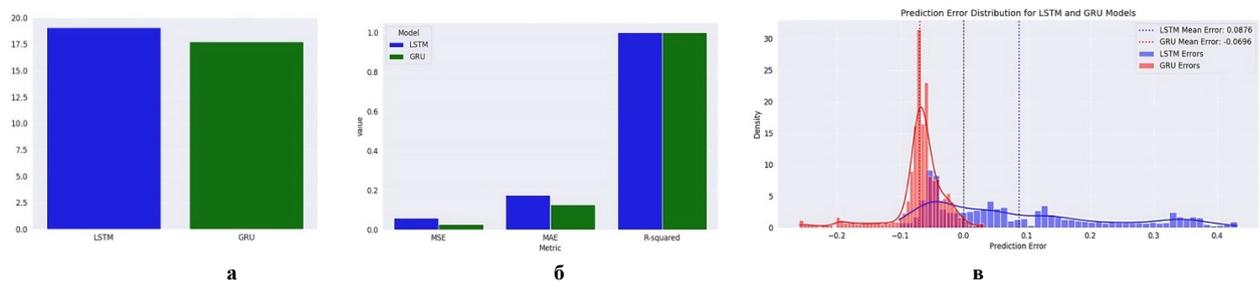


Рис. 5. а) - Діаграма - порівняння часу навчання моделей LSTM і GRU протягом 25 епох (оцінка в секундах) б) - Діаграма - порівняння продуктивності моделей LSTM і GRU протягом 25 епох в) - Діаграма - розподіл помилок прогнозування для моделей LSTM і GRU за 25 епохами

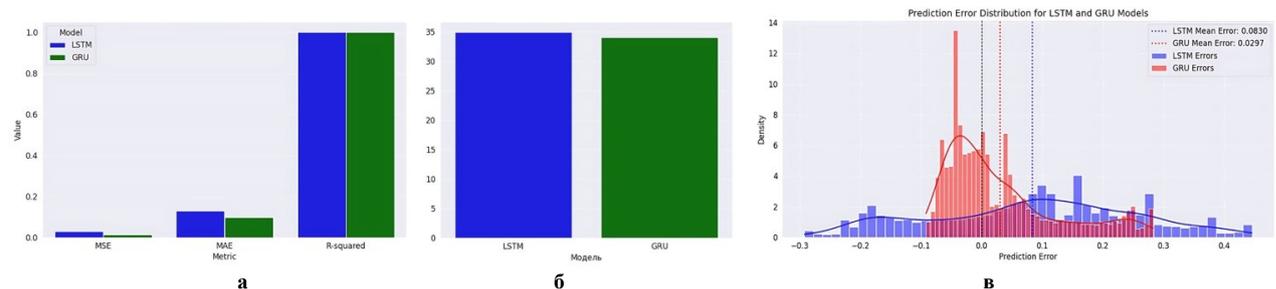


Рис. 6. а) - Діаграма — порівняння часу навчання моделей LSTM і GRU протягом 50 епох (оцінка в секундах) б) - Діаграма - порівняння продуктивності моделей LSTM і GRU протягом 50 епох в) - Діаграма - розподіл помилок прогнозування для моделей LSTM і GRU за 50 епох

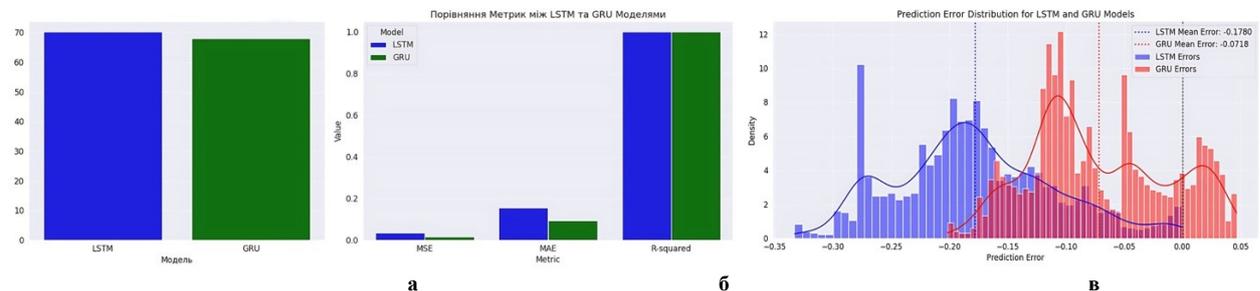


Рис. 7. а) - Діаграма - порівняння часу навчання моделей LSTM і GRU протягом 100 епох (оцінка в секундах) б) - Порівняння діаграм продуктивності моделей LSTM і GRU за 100 епох в) - Діаграма - розподіл помилок прогнозування для моделей LSTM і GRU за 100 епох=

У таблицях 1-3 представлено порівняльний аналіз моделей LSTM та GRU, навчених для 25, 50 та 100 епох. Оцінка зосереджена на ключових показниках ефективності: Час навчання, середня квадратична похибка (MSE), середня абсолютна похибка (MAE) та R^2

Експерименти виконувалися на системі з графічним процесором NVIDIA GeForce RTX 3050 Ti, використовуючи його обчислювальну потужність для прискореного навчання моделей та виведення результатів.

Таблиця 1

Порівняння середньої продуктивності моделей LSTM і GRU для прогнозування (25 епох, 3 повторення)

Metric	LSTM	GRU
Training Time (s)	19.0696 ± 0.3408	17.7432 ± 0.0769
MSE	0.0583 ± 0.0493	0.0267 ± 0.0213
MAE	0.1753 ± 0.0775	0.1267 ± 0.0473
R-squared	0.9990 ± 0.0008	0.9996 ± 0.0004

Таблиця 2

Порівняння середньої продуктивності моделей LSTM і GRU для прогнозування (50 епох, 3 повторення)

Metric	LSTM	GRU
Training Time (s)	34.8870 ± 0.4225	34.0277 ± 0.1292
MSE	0.0286 ± 0.0117	0.0144 ± 0.0075
MAE	0.1303 ± 0.0239	0.0982 ± 0.0288
R-squared	0.9995 ± 0.0002	0.9998 ± 0.0001

Таблиця 3

Порівняння середньої продуктивності моделей LSTM і GRU для прогнозування (50 епох, 3 повторення)

Metric	LSTM	GRU
Training Time (s)	70.1979 ± 0.3274	67.9957 ± 0.6972
MSE	0.0350 ± 0.0117	0.0163 ± 0.0136
MAE	0.1539 ± 0.0224	0.0941 ± 0.0471
R-squared	0.9994 ± 0.0002	0.9997 ± 0.0002

Експерименти, проведені для порівняння LSTM та GRU моделей для задачі прогнозування системи Лоренца, дозволили зробити наступні висновки:

- Час, необхідний для навчання, розподілився наступним чином. GRU демонструє стабільно менший час навчання порівняно з LSTM на всіх етапах процесу. Для ілюстрації, для 25 епох ГПУ потрібно 17,74 секунди, щоб завершити процес навчання, тоді як LSTM - 19,07 секунди. Порівнянна перевага зберігалася для 50 і 100 епох. Це свідчить про те, що архітектура GRU є більш ефективною, що робить її кращим варіантом для задач з обмеженим часом або ресурсами.

- Точність прогнозів оцінювалася за допомогою показників середньої квадратичної похибки (MSE) та середньої абсолютної похибки (MAE). GRU демонструє нижчі значення середньої квадратичної похибки (MSE) і середньої абсолютної похибки (MAE), ніж LSTM, на всіх етапах. Наприклад, на 50-епохальній позначці GRU має середньоквадратичну похибку 0,0144 порівняно з 0,0286 для LSTM, а середню абсолютну похибку 0,0982 порівняно з 0,1303. Це підтверджує твердження, що GRU є більш ефективним у моделюванні залежностей, притаманних даним, тим самим сприяючи більш точному прогнозуванню.

- Коефіцієнт детермінації (R^2). Обидві моделі демонструють високі значення (R^2), що свідчить про здатність ефективно пояснювати варіацію цільової змінної. Тим не менш, GRU продемонструвала кращі результати порівняно з LSTM. Наприклад, на 100 епохах GRU показав R^2 0,9997, тоді як LSTM - 0,9994.

- Стабільність результатів є важливим аспектом, який слід враховувати. GRU продемонстрував зменшені стандартні відхилення для більшості метрик, що свідчить про підвищену стабільність його роботи в повторних експериментах.

Отже, GRU виявилася більш ефективною моделлю для прогнозування проблем, пов'язаних із системою Лоренца, про що свідчить наступне:

- GRU потребує менше часу на навчання і демонструє вищу точність, про що свідчать метрики MSE та MAE.
- Демонструє дещо вищі значення R^2 і більшу стабільність результатів.

Хоча LSTM показав дещо нижчі результати, він залишається надійним варіантом для задач, які потребують моделювання особливо складних довгострокових залежностей. Однак у контексті цієї задачі GRU є більш оптимальним вибором завдяки своїй швидкості та точності.

Отримані результати свідчать про те, що нейронні мережі при правильному навчанні можуть бути використані для прогнозування можливої поведінки хаотичної системи Лоренца. Здійснивши прогнозування роботи системи з різними наборами початкових умов та параметрів можна створити бази знань для нейронних мереж, що в подальшому можуть бути використанні для відновлення цих значень по співпаданню. При детектуванні сигналів в телекомунікаційній системі з прямим розширенням спектру із застосуванням нейронних мереж можливе покращення ефективності розпізнавання хаотичної поведінки системи, і як наслідок зросте ефективність розпізнавання інформаційних повідомлень.

Висновки

Спроба передбачити поведінку хаотичної системи, такої як система Лоренца, є складним завданням, головним чином через особливе явище, яке називається ефектом метелика. Цей ефект означає, що невеликі відмінності в початкових умовах можуть призвести до помітно різних результатів.

Використання рекурентних нейронних мереж (RNN), зокрема LSTM або GRU, дозволяє здійснювати прогнозування хаотичних систем, де вони можуть навчатися і ефективно відтворювати закономірності в даних при різних комбінаціях початкових умов та параметрів хаотичної системи. Тим не менш, методологія не позбавлена обмежень.

У хаотичних системах накопичення помилок прогнозування відбувається з надзвичайною швидкістю через екстремальну чутливість до зміни значень початкових умов та параметрів, а також при наявності помилок заокруглень. Хоча RNN здатні вловлювати короткострокову динаміку, їхні прогнози все більше відхиляються від істинних значень на довших часових горизонтах через поширення помилок. Крім того, хаотична система Лоренца, що розглядаються, включають дуже складну, нелінійну динаміку, яку стандартні архітектури RNN можуть не повністю врахувати. Це впливає на здатність моделі узагальнювати дані, з якими вона стикається.

Отримані результати свідчать про те, що обрані моделі нейронних мереж GRU та LSTM практично однаково ефективно справляються з поставленим завданням, хоча перша демонструє дещо кращі результати.

Для телекомунікаційних систем результати дослідження допоможуть в ефективному виборі моделі для прогнозування складних часових рядів, таких як хаотичні мережеві процеси, а також допоможуть в управлінні перевантаженням мережі та виявленні мережевих аномалій. Також перспективним напрямком застосування нейромережевих моделей GRU та LSTM є в прогнозуванні хаотичних коливань для телекомунікаційних систем із прямим розширенням спектра. Завдяки цьому можна покращити детектування та розпізнавання сигналів у високошвидкісних каналах зв'язку, а саме залучення хаотичних послідовностей із широким спектром знизить взаємні завади й підвищить стійкість передачі даних, одночасно забезпечить необхідний рівень безпеки під час кодування та декодування сигналів. Порівняння GRU та LSTM підтверджує, що обидва підходи можуть ефективно обробляти нестабільну та нелінійну динаміку даних. Оптимальний вибір між архітектурами залежить від системних вимог - GRU є менш складною та обчислювально ефективною, але LSTM може ефективно моделювати довгострокові залежності.

Література

1. Oestreicher C.: A history of chaos theory. *Dialogues in Clinical Neuroscience*, 2007, Vol. 9, No. 3, 279–289.
2. Katok A., Hasselblatt B.: *Introduction to the Modern Theory of Dynamical Systems*. Cambridge University Press, Cambridge 1995.
3. Liu Z.: Applications of dynamical systems in physics. *Theoretical and Natural Science*, 2024, Vol. 36, No. 1, 85–89.
4. Sarker I.H.: *Machine Learning: Algorithms, Real-World Applications and Research Directions*. SN Computer Science, 2021, Vol. 2, No. 3, 160.
5. Arokodare Oluwatomisin. Kim Jongyeop, Leveraging Machine Learning for Enhanced Predictive Accuracy in Time Series Forecasting: A Comparative Analysis of LSTM and GRU Models, 2024, 8 p. DOI 10.13140/RG.2.2.30014.80965.
6. Jeaab, Khalid & Saoudi, Youness & Falloul, Moulay El. (2024). A Comparison of LSTM, GRU, and XGBoost for forecasting Morocco's yield curve. *Mathematical Modeling and Computing*, 11, 674-681. 10.23939/mmc2024.03.674.
7. Nielsen M.A.: *Neural Networks and Deep Learning*. Determination Press, 2015.
8. Mienye I.D., Swart T.G., Obaido G.: *Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications*. *Information*, 2024, Vol. 15, No. 9, 517.
9. G. Goui, A. Zrelli and N. Benletaief, "A Comparative Study of LSTM/GRU Models for Energy Long-Term Forecasting in IoT Networks," 2023 IEEE/ACIS 23rd International Conference on Computer and Information Science (ICIS), Wuxi, China, 2023, pp. 60-64, doi: 10.1109/ICIS57766.2023.10210257.
10. N. X. Quyen, Yem Van Vu, Thang Manh Hoang "A Chaos-Based Secure Direct-Sequence/Spread-Spectrum Communication System" *Abstract and Applied Analysis*, Volume 2013, Article ID 764341, 11 pages, <http://dx.doi.org/10.1155/2013/764341>.
11. Serrano-Pérez, J.d.J., Fernández-Anaya, G., Carrillo-Moreno, S. et al. New Results for Prediction of Chaotic Systems Using Deep Recurrent Neural Networks. *Neural Process Lett* 53, 1579–1596 (2021). <https://doi.org/10.1007/s11063-021-10466-1>.
12. Hodson T.O., Over T.M., Foks S.S.: Mean Squared Error, Deconstructed. *Journal of Advances in Modeling Earth Systems*, 2021, Vol. 13, No. 12.
13. Hodson T.O.: Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development*, 2022, Vol. 15, No. 14, 5481–5487.
14. Robeson S.M., Willmott C.J.: Decomposition of the mean absolute error (MAE) into systematic and unsystematic components. *PLoS ONE*, 2023, Vol. 18, No. 2, e0279774.
15. Willmott C.J., Matsuura K.: Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. *Climate Research*, 2005, Vol. 30, No. 1, 79–82.
16. Chicco D., Warrens M.J., Jurman G.: The Coefficient of Determination R-Squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Computer Science*, 2021, Vol. 7, e623
17. Alexander D.L.J., Tropsha A.: Beware of R²: Simple, Unambiguous Assessment of the Prediction Accuracy of QSAR and QSPR Models. *Journal of Chemical Information and Modeling*, 2015, Vol. 55, No. 7, 1316–1322.