

## СТВОРЕННЯ ТА РЕАЛІЗАЦІЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ ЦІНИ ВІРТУАЛЬНИХ АКТИВІВ

А. Цемко, З. Любунь

*Кафедра радіофізики та комп'ютерних технологій,  
Львівський національний університет імені Івана Франка,  
вул. Ген. Тарнавського, 107, 79017 Львів, Україна*

[zinoviy.lyubun@lnu.edu.ua](mailto:zinoviy.lyubun@lnu.edu.ua)

Рекурентні нейронні мережі використовують для задач прогнозування. Оскільки маючи набір даних за певний проміжок часу, ми можемо навчити нейронну мережу виявити певні закономірності між даними і спрогнозувати майбутні значення часового ряду. Починаючи від початку 2021 року віртуальні активи почали ставати популярними, і задача прогнозування майбутніх значень ціни волатильних активів набула популярності. Використання рекурентних нейронних мереж, на відміну від методів сімейства ARIMA, може дозволити враховувати за вхідні параметри не тільки поточне значення ціни, а і інші фактори, що можуть впливати на ціну.

*Ключові слова:* рекурентні нейромережі, прогнозування, віртуальні активи.

Метою роботи є використання рекурентних нейронних мереж для прогнозування майбутніх значень ціни віртуальних активів базуючись на динаміці їх зміни за попередні інтервали часу. Потрібно визначити, яка структура рекурентних мереж може надати найбільш вірогідні дані після навчання базуючись виключно на даних ціни минулих років.

Для створення та навчання нейронної мережі, було обрано бібліотеку TensorFlow [1] мови програмування Python від компанії Google, з відкритим кодом. Бібліотека TensorFlow містить в собі API Keras, який дозволяє створювати структури нейронних мереж пошарово, налаштовуючи відповідні параметри кожного з шарів. Так, для створення рекурентних нейронних мереж API Keras надає можливість створення таких шарів: RNN, LSTM та GRU [2, 3].

Класичні рекурентні нейромережі (RNN) складаються з нейронів, які ітеративно опрацьовують вхідні дані. Таким чином, один нейрон отримує і обробляє декілька вхідних значень. Якщо звичайний нейрон містить вхід, вихід і значення коефіцієнта ваг, то рекурентний нейрон крім того з'єднує свій вихід зі входом, тим самим створюючи рекурсивний зв'язок (рис. 1). Нейронні мережі, які складаються з рекурентних нейронів займають менший об'єм пам'яті, але процес обробки вхідного вектора є повільнішим, ніж у звичайній нейронній мережі.

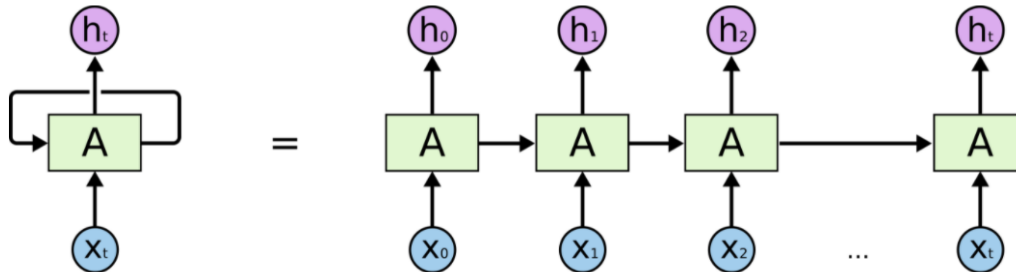


Рис. 1. Демонстрація роботи рекурентного нейрона

Оскільки навчання нейронної мережі проходить ітеративно згідно кожної навчаючої пари, а кожен нейрон обраховує вхідне значення декілька разів, то це створює проблему зникнення градієнта [4]. З кожною ітерацією обрахунку значення градієнту перемножується, а це призводить до ефекту зникнення градієнту. Дану проблему частково розв'язують так звані LSTM (Long-Short temporary memory) нейронні мережі [5].

Оскільки рекурентні нейронні мережі, за своєю архітектурою, на кожній ітерації перемножують значення вхідного вектора, то кожна ітерація перемноження зменшує вплив оброблених значень на минулій ітерації вдвічі. Дану проблему проілюстровано на Рис. 2. Нехай ми маємо вхідний вектор з 7 значень числового ряду. Оскільки рекурентний нейрон буде по черзі обраховувати кожне вхідне значення, то з кожним наступним кроком, вплив минулого значення зменшується вдвічі. Чим більше ітерацій обрахунку, тим менше перше вхідне значення впливає на результуючий вихід. Рекурентні нейронні мережі LSTM якраз дозволяють уникнути цієї проблеми [5]. LSTM нейронні мережі містять в собі додаткові комірки пам'яті, що обирають які дані варто запам'ятати, а які забути.

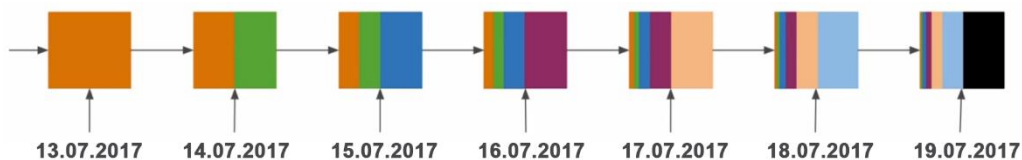


Рис. 2. Ілюстрація впливу значень від перших до останніх

Розглянемо задачу прогнозування значення ціни криптовалюти Bitcoin відносно долара. Було обрано набір даних – щоденне співвідношення вартості обох валют починаючи від 22 вересня 2014 року до 28 листопада 2021 року [6]. Набір даних містить масив 383 значень максимальної ціни протягом дня (рис. 3).

Дані для навчання розділяються на так звані вікна із заданою кількістю відліків  $m$ :  $t_1, t_2, \dots, t_m$ . Цей набір даних використовуватиметься як вхідні дані для навчання. За бажаний вихід беремо значення в момент часу  $t_{m+1}$  (рис. 4). Загальна кількість навчаючих пар порядку 300, при вікні спостереження розміром 25 відліків. Вікном спостереження  $n$ -го розміру в даному випадку називають набір значень часових відліків від  $k$ -го до  $k+n$  значення.



Рис. 3. Вартість Bitcoin'у по днях



Рис. 4. Принцип підготовки навчальних пар для одно крокового прогнозування на основі вікна спостереження розміром в 25 точок

Для отримання прогнозованих значень вибрана LSTM мережа, перший шар якої містить в собі 128, а другий шар – 64 навчальних параметрів. Оскільки нейронна мережа має прогнозувати одне наступне значення, то кінцевий шар містить в собі лише один лінійний нейрон.

Перед навчанням нейронної мережі датасет було розділено на дві частини: навчаючі пари і пари для тестування. Навчаючі пари було використані для навчання нейронної мережі, а пари для тестування були використанні для оцінки точності прогнозування вже навченої мережі. На рис. 5 зображено розділення даних:

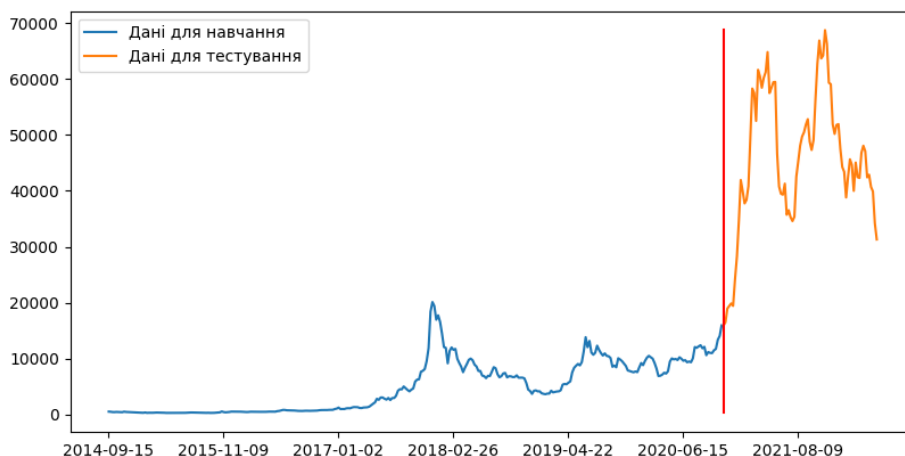


Рис. 5 Проміжки даних для навчання та тестування

Процес навчання моделі з вікном спостереження в 25 точок для прогнозування однієї наступної точки (одно крокове прогнозування) зображено на рис. 6, де горизонтальна вісь відповідає за номер епохи, а вертикальна вісь – середньоквадратичну похибку між прогнозованим і очікуваним значенням. Навчання тривало 2500 епох.

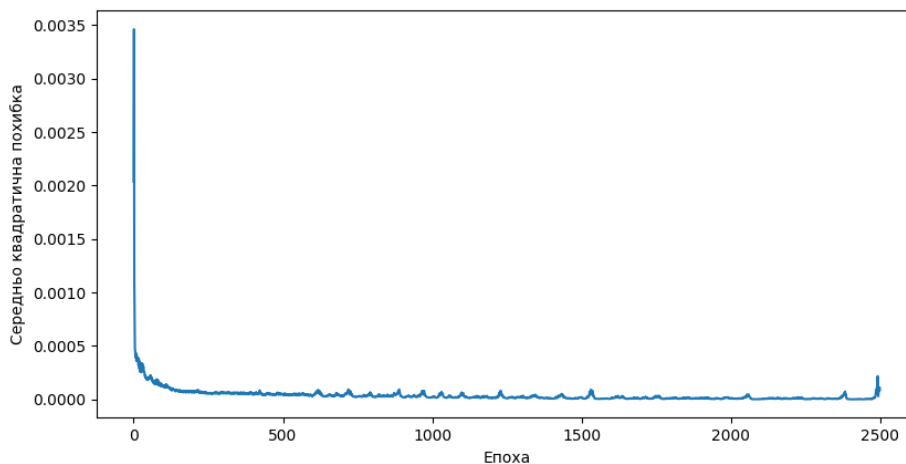


Рис. 6. Процес навчання нейронної мережі

Результати однокрокового прогнозування навченої нейронної мережі відображено на рис. 7. Аналізуючи результати однокрокового прогнозування на проміжку даних, які було взято для навчання, нейронна мережа коректно прогнозує наступні значення.

Однак, на тестових даних нейронна мережа постійно намагається спрогнозувати спад ціни до значень, які лежать у проміжку значень з тестових даних. Тобто, на тестовому проміжку нейронна мережа погано прогнозує наступні значення. Причиною цього може бути те, що дані для навчання не є стаціонарними, і саме через це нейронна мережа «завчила» можливі прогнозовані значення лише у навчених цінових діапазонах. Таким чином, для успішнішого навчання нейронної мережі необхідно забезпечити стаціонарність часової послідовності даних.



Рис. 7. Результати навчання нейронної мережі з прогнозом одного значення

Спроба замінити абсолютні значення вартості на значення щоденних абсолютних значень зміни ціни (рис.8) не дали позитивного результату, бо отриманий часовий ряд також не є стаціонарним і тому не підходить для використання його у процесі навчання нейронної мережі.

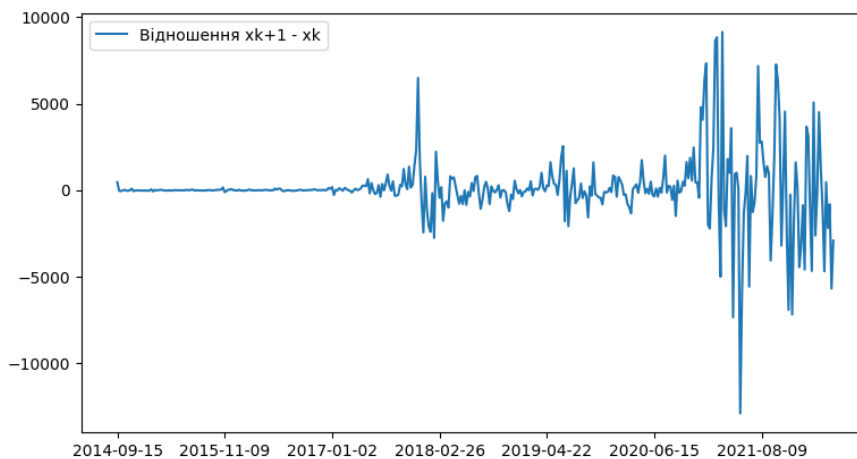


Рис. 8. Представлення даних у форматі приростів

Для отримання стаціонарної часової послідовності  $y(k)$ , вхідну послідовність  $x(k)$  слід замінити на послідовність відношень цін :  $y_k = \frac{x_{k+1}}{x_k}$ , де  $k$  – момент часу.

На рис. 9 показана послідовність відношень ціни обраних для дослідження даних. Отримана послідовність має ознаки стаціонарності, оскільки на будь-якому з проміжків має середнє значення близьке до 1.015 та значення дисперсії близьке до 0.0094.

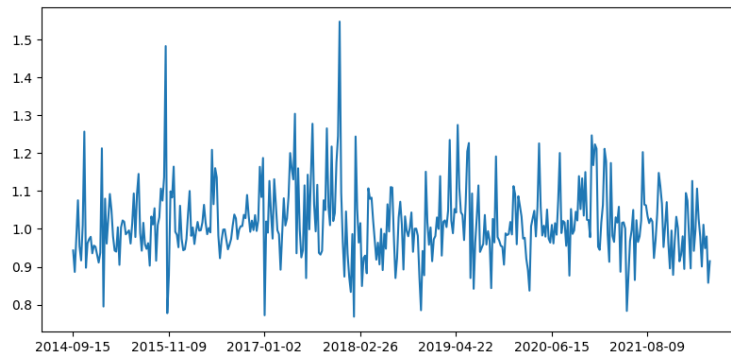


Рис. 9. Відносна часова послідовність типу  $\frac{x_{k+1}}{x_k}$

Отримана часова послідовність була розділена на навчальний та тестовий проміжки (рис. 10). Навчальний проміжок даних був використаний для навчання нейронної мережі з вікном спостереження в 25 точок для прогнозування однієї наступної точки. Результат прогнозування навченої нейронної мережі зображено на рис. 11. Аналізуючи отримані результати можна побачити, що навчена на такій послідовності нейронна мережа непогано прогнозує поведінку часової послідовності, на відміну від мережі навченої на абсолютних цінових значеннях.

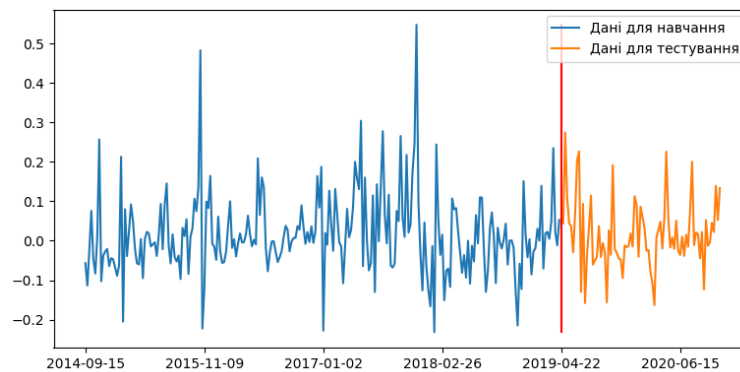


Рис. 10. Проміжки даних для навчання та тестування

відносної часової послідовності типу  $\frac{x_{k+1}}{x_k}$



Рис. 11 Результати навчання нейронної мережі з прогнозом одного значення для відносних часових рядів з вікном спостереження в 25 точок

Для визначення оптимального розміру вікна спостережень як критерій оптимальності було вибрано середньо квадратичну похибку між тестовими даними та отриманими прогнозованими значеннями нейронної мережі. Результати досліджень показані на Рисунку 12. Таким чином, оптимальним розміром вікна спостереження для отримання однокрокового прогнозу є розмір вікна в двадцять відліків.

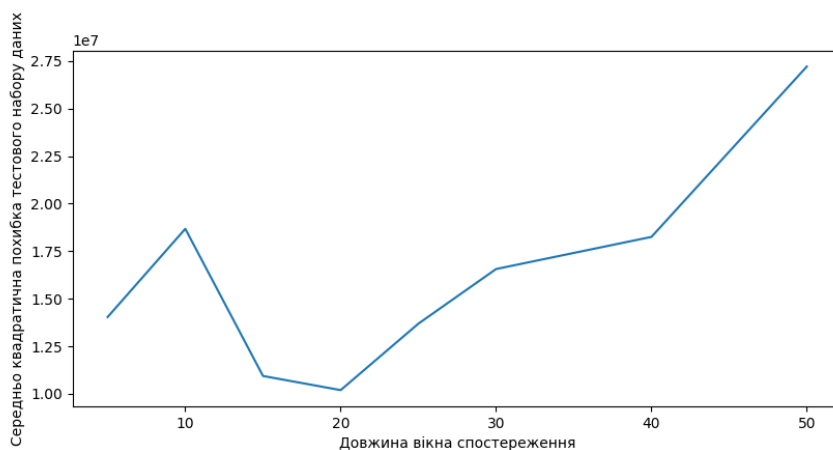


Рис. 12 Залежність середньо квадратичної похибки прогнозування нейронної мережі від розміру вікна спостереження.

Таким чином, аналізуючи отримані результати, можна твердити, що використання рекурентних нейронних мереж для прогнозування ціни віртуальних активів дає непогані результати. Навчена мережа здатна відтворити основні закономірності зміни часового ряду. Для навчання таких нейронних мереж замість нормованих абсолютних значень цін

часових рядів потрібно використовувати відносно представлення часових рядів, оскільки воно для випадку досліджуваного віртуального активу Bitcoin, дає близький до стаціонарного часовий ряд. Також для даного набору вихідних даних було визначено оптимальне значення вікна спостережень. Створений застосунок дає змогу використовувати нестационарний часовий ряд цін віртуальних активів для одного крокового прогнозування. Крім цього, застосунок дозволяє визначити оптимальний розмір вікна спостереження для прогнозу ціни віртуальних активів. Створений застосунок цілком може бути використаний для прогнозування реальних значень ціни віртуальних активів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] TensorFlow Guide [Електронний ресурс] – Режим доступу: <https://www.tensorflow.org/guide>
- [2] Time Series forecasting [Електронний ресурс] – Режим доступу: [https://www.tensorflow.org/tutorials/structured\\_data/time\\_series](https://www.tensorflow.org/tutorials/structured_data/time_series)
- [3] Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras [Електронний ресурс] – Режим доступу: <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>
- [4] Проблема зникання градієнту [Електронний ресурс] – Режим доступу: [https://www.wikiwand.com/uk/Проблема\\_зникнення\\_градієнту](https://www.wikiwand.com/uk/Проблема_зникнення_градієнту)
- [5] A Gentle Introduction to Exploding Gradients in Neural Network [Електронний ресурс] – Режим доступу: <https://machinelearningmastery.com/exploding-gradients-in-neural-networks/>
- [6] Bitcoin USD (BTC-USD) [Електронний ресурс] – Режим доступу: <https://finance.yahoo.com/quote/BTC-USD/history/>

**USING A NEURAL NETWORK FOR PRICE PREDICTION OF VIRTUAL ASSETS****A. Tsemko, Z. Liubun**

*Radio Physics and Computer Technologies Department,  
Ivan Franko National University of Lviv,  
107 Tarnavsky St., UA-79017 Lviv, Ukraine  
[zinoviy.lyubun@lnu.edu.ua](mailto:zinoviy.lyubun@lnu.edu.ua)*

Due to the structure of recurrent neural networks, they are used for the prediction tasks, such as price prediction. The price prediction tasks are based on the historical data of price movements during the specified period. This data can be used for training the recurrent neural network for price prediction. Expected, that the neural network will recognize specific patterns in sequential data and will be able to predict the next trends etc. From the 2021 year, virtual assets such as Bitcoin increase their popularity all around the world. Virtual assets such as Bitcoin are classified as highly volatile assets. For this type of asset, the prediction task is so important, due to the ability to make a long and a short position several times per day, week, etc. Using the recurrent neural network, against the ARIMA methods, can help to include the other data except for the



price history. For example, it can you the history of several operations for these assets per day, the price history of the other virtual assets, that can have some relations with, etc. In this article, as a first step, work was focused on properly formatting the price history data for achieving the lowest prediction error. Also, the idea is to create a framework for working with different virtual assets for prediction. The problem with using a neural network for prediction is that absolute values of virtual assets price are not stationary. And the neural network training process stuck between the minimum and maximum values of training data. It creates a problem, where the trained neural network cannot handle the data, that is bigger than the trained and it always tried to predict the value in the trained range. This work, investigated the neural network for single prediction only. Also, was compared the two possible ways to format the data to the stationary data.

*Key words:* recurrent neural networks, prediction, virtual assets

*Стаття надійшла до редакції 14.10.2022.*

*Прийнята до друку 29.11.2022.*