

## КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

УДК 519.7

### ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗАДАЧ ПРОГНОЗУВАННЯ

**О. В. Бубнов**

здобувач вищої освіти першого (бакалаврського) рівня, група ПМ-41,  
навчально-науковий інститут автоматичної, кібернетичної та обчислювальної техніки  
Науковий керівник – к.ф.-м.н., доцент О.В. Прищеп

*Національний університет водного господарства та природокористування,  
м. Рівне, Україна*

У статті розглядається один із підходів прогнозування даних для вирішення актуальних задач сьогодення. За допомогою алгоритмів прогнозування ми можемо передбачити погоду, стан ґрунтів, ціни на бензин, активи криптовалют та інше, використовуючи статистичні дані. Проте алгоритми прогнозування потребують удосконалення та застосування нових підходів, зокрема, з використанням нейронних мереж. Це дозволить підвищувати точність прогнозування та зберігати складні закономірності у даних.

**Ключові слова:** нейронні мережі, прогнозування, статистичні методи, архітектура нейронних мереж.

**The article deals with one of the data forecasting approaches for solving today's relevant problems. With the help of forecasting algorithms, we can predict the weather, soil conditions, gasoline prices, cryptocurrency assets, and more using statistical data. However, forecasting algorithms need improvement and the application of new approaches, particularly using neural networks. This will improve forecasting accuracy and preserve complex patterns in the data.**

**Keywords:** neural networks, forecasting, statistical methods, architecture of neural networks.

Задачам на прогнозування часових рядів приділяється висока увага протягом кількох десятиліть. Але дослідження в даній сфері проводилися більшою мірою з застосуванням класичних статистичних методів. Найважливішим серед цих інструментів є багатовимірною лінійною регресією. Більшість досліджень при прогнозуванні часових рядів присвячені регресійним моделям (лінійна, множинна, нелінійна), які зараз майже не застосовуються на відміну від авторегресійних моделей.

Наступним етапом досліджень в прогнозуванні криптовалютних ринків стає розробка підходів на основі штучного інтелекту та штучних нейронних мереж ANN (Artificial Neural Network)[1]. Перші роботи в даному напрямку поєднали AR моделі та їх модифікації з ANN, що дало в результаті модель NNAR (Neural Network Autoregression). Аналіз результатів застосування NNAR показав, що дана модель з переоцінкою на кожному етапі може бути застосована у погодинних і похвилинних прогнозах. Методика з використанням логістичної регресії та методу опорних векторів SVM (Support Vector Machine) для прогнозування ціни біткоїна. Модель на основі BNN (Bayesian Neural Network) досліджується в роботі [2], а в пропонується застосування ансамбля нейронних мереж. Також в дослідили прогнозування біткоїна за допомогою методики на основі штучної нейронної мережі ANN (Artificial Neural Network) в поєднанні з використанням ринкових технічних показників – технічного аналізу.

Проте отримані результати виявилися неприйнятними через низьку продуктивність та час навчання.

При застосуванні ANN виникає питання її структури та підготовки вхідних даних. В роботі досліджується дві нейронні мережі RNN-LSTM та GMDH-подібна. Велика кількість досліджень з застосування даних інструментів не дають сталої думки щодо ефективності їх як моделей для розв'язання задачі прогнозування криптовалютних часових рядів. Тому в даному дослідженні пропонується на вхід мережі подавати не історичні дані часового ряду, а його емпіричну модову декомпозицію – внутрішні модові функції.

**Метою даної статті є** виявлення найтиповіших проблем та способи їх вирішення при роботі з алгоритмами прогнозування даних. Не всі методів прогнозування добре працюють із даними з великими коливаннями. Припустимо, що ціна біткойна зростає (рис.1).

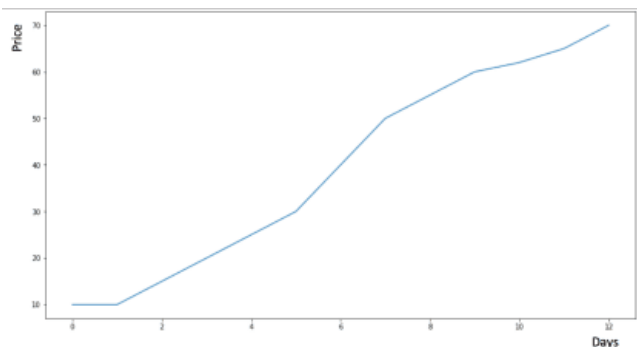


Рис. 1. Ціна біткойна

Тренд – це загальна картина цін, яку ми спостерігаємо протягом певного періоду часу. У цьому випадку бачимо, що існує тенденція до збільшення. Нам потрібний метод, який може точно відображати тренд без будь-яких припущень. Такий метод, який враховує тренд набору даних, називається методом лінійного тренду Холта. Кожен набір даних часового ряду може бути розкладений на складові: тренд, сезонність та залишок. Будь-який набір даних, який слідує за трендом, може

використовувати метод лінійного тренду Холта для прогнозування (рис. 2).

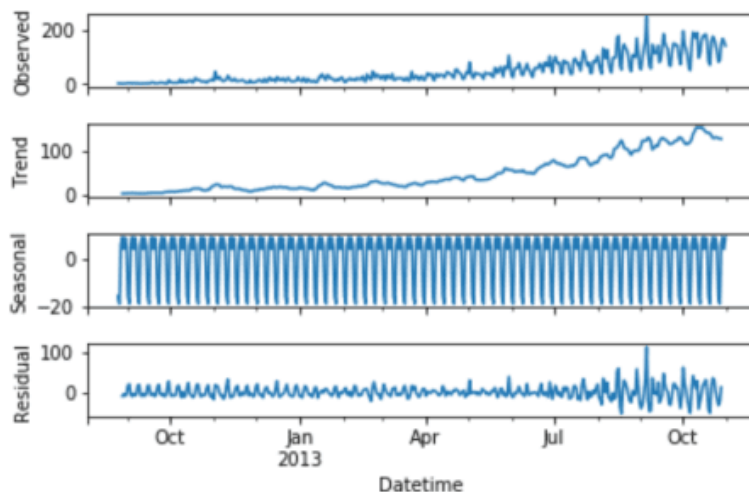


Рис. 2. Складові часового ряду

З отриманих графіків видно, що цьому набору даних відповідає тренд, що зростає. Отже, ми можемо використати лінійний тренд Холта для прогнозування майбутніх цін.

Холт розширив просте експонентне згладжування, щоб дозволити прогнозувати дані з трендом [2]. Це не більше, ніж експоненційне згладжування, яке застосовується як до рівня (середнє значення в ряду), так і до тренду. Щоб подати це формально, нам тепер потрібно три рівняння: одне – для рівня, інше – для тренду та третє – для об'єднання рівня та тренду, щоб отримати очікуваний прогноз.

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1},$$

$$\hat{y}_{t+h|t} = l_t + hb_t$$

У наведених вище трьох рівняннях ви можна помітити, що додано рівень та тренд для створення рівняння прогнозу. Ці рівняння необхідні, щоб створити рівняння прогнозу. Ми також можемо згенерувати мультиплікативне рівняння прогнозу, помноживши тренд та рівень замість складання. Коли тренд росте чи падає лінійно, використовується адитивне рівняння, коли тренд росте чи падає експоненційно, використовується мультиплікативне рівняння. Практика показує, що множення є стійкішим предиктором, проте адитивний метод простіше зрозуміти (рис. 3).

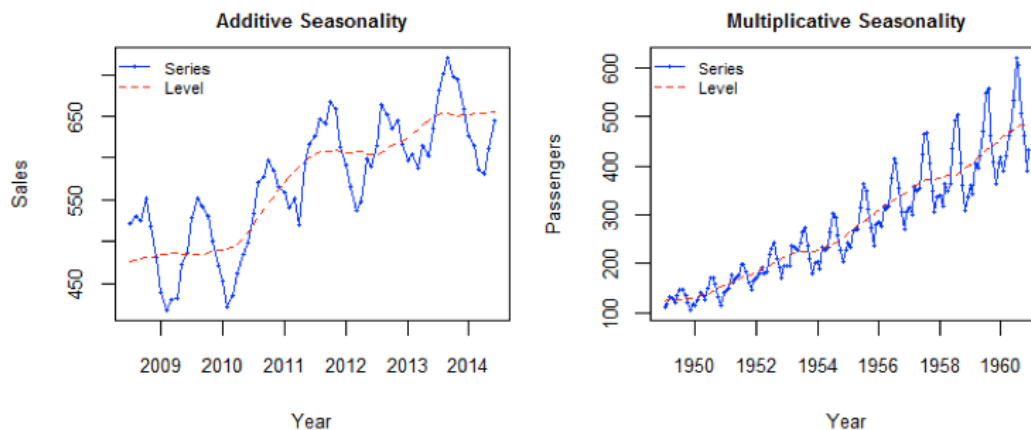


Рис. 3. Приклади прогнозування

При розрахунку RMSE для перевірки точності нашої моделі, ми бачимо, що цей метод точно відображає тренд і, отже, забезпечує найкраще рішення, порівняно з розглянутими раніше моделями. Ми також можемо налаштувати параметри, щоб отримати більш точну модель.

**Обґрунтування вибору LSTM.** Люди не починають думати з чистого аркуша що секунди. Читаючи пост, ви розумієте кожне слово, ґрунтуючись на розумінні попереднього слова. Ми не викидаємо з голови все і не починаємо думати з нуля. Наші думки мають сталість. Традиційні нейронні мережі не мають цієї властивості, і в цьому їхній головний недолік. Уявімо, наприклад, що ми хочемо класифікувати події, що відбуваються у фільмі. Незрозуміло, як традиційна нейронна мережа могла б використати міркування про попередні події фільму, щоб отримати інформацію про наступні.

Вирішити цю проблему допомагають рекурентні нейронні мережі (Recurrent Neural Networks, RNN). Це мережі, що містять зворотні зв'язки та дозволяють зберігати інформацію. На жаль, у міру зростання відстані, RNN втрачають здатність пов'язувати інформацію. Проте LSTM не знає таких проблем. Довга короткострокова пам'ять (Long short-term memory; LSTM) – особливий різновид архітектури рекурентних нейронних мереж, здатна до навчання довгострокових залежностей [3]. Вони були представлені Зеппом Хохрайтером та Юргеном Шмідхубером (Jürgen Schmidhuber) у 1997 році, а потім удосконалені та популярно викладені в роботах багатьох інших дослідників. Вони чудово вирішують цілу низку різноманітних завдань і нині широко використовуються. LSTM розроблено спеціально, щоб уникнути проблеми довготривалої залежності. Запамятовування інформації на довгі періоди часу – це їхня звичайна поведінка, а не щось, чому вони намагаються навчитися.

Розберемо на реальному прикладі роботу LSTM моделі для передбачування стоку річки, маючи дані за останні 75 років (таблиця). Також ознайомимося з проблемами, які

виникли під час прогнозування. Тут ми ознайомимося з основними принципами роботи. Групуємо отримані дані і визначаємо кількість стовпців та рядків.

Таблиця

Стік річки за останні 75 років (км<sup>3</sup>/рік)

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0	52,8	52,8	75,1	88,5	48	37,2	49,2	66,8	65,8	45,5	51,7	35,8
1	21,9	40,7	189	118	83,3	64,3	44,5	50,8	28,7	28	34,6	33,7
2	29	27,8	130	55,3	39,8	26,5	22,6	25,9	17,7	17,7	19,3	10,8
3	11,2	59,5	65,5	130	31,5	21,1	23	9,1	11,3	13,2	20	49
4	30,3	26,1	73,4	443	72,5	24,5	36,6	29,4	16,4	19	21,8	27,5
...	...	...	...	...	...	...	...	...	...	...	...	...
78	37,4	71,3	251	108	33,5	13,4	24,4	17,7	18,2	42,3	41,2	32,7
79	32,7	54,8	50,6	41	24,4	15,4	15,6	38,1	45,7	60,7	44,4	43,4
80	49,7	62,1	110	98	98,9	46,3	31,3	34,6	54,7	61,1	51,8	62,9
81	54,2	43,5	89,7	172	56,9	94,3	44,8	27,6	65,1	67,8	62,4	48
82	47,3	61,1	64,1	38,3	21,1	16,2	11,7	20,4	37,4	51,6	45,7	38,1

[83 rows x 12 columns]

Дуже просто прогнозувати, коли у даних є закономірності. Тобто поступовий спуск чи підйом, який супроводжується і чергується. При роботі зі природними даними такого майже ніколи не відбувається. Завжди є великі стрибки, які в майбутньому можуть дуже сильно впливати на процес навчання. Тому для таких цілей використовується згладжування даних. У наведеному нижче ми згладжуємо дані, які отримуємо з самого початку і вже при подальшому навчанні наша нейронна мережа буде використовувати згладжені дані, що спростить процес навчання та сформує нам в подальшому точні результати прогнозування. Після створення самої LSTM-моделі нейронної мережі, яка в майбутньому буде навчатися і прогнозувати наші дані, отримали (рис. 4).

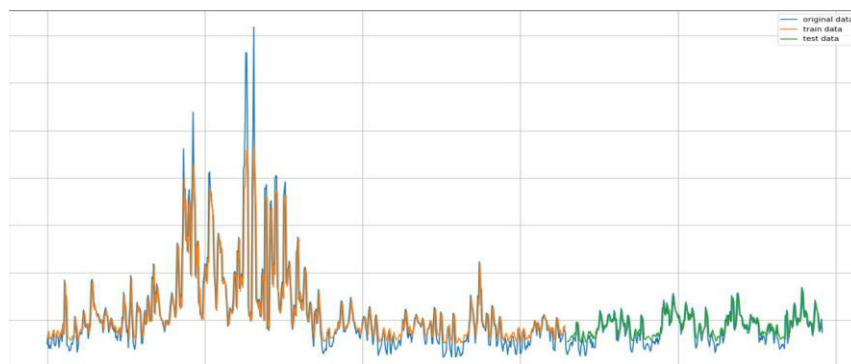


Рис. 4. Згладжування даних

дані. Оранжевим кольором позначено дані, за яким наша LSTM-модель навчалася. Зеленим кольором позначено дані які передбачила наша нейронна мережа.

Отже, ми розглянули основні проблеми при роботі з прогнозуванням даних. Використали LSTM, що є великим кроком у розвитку RNN. При цьому постає природне питання: яким буде

наступний великий крок? На думку дослідників, наступний великий крок – «увага» (attention). Ідея полягає в наступному: кожен крок RNN бере дані з більшого сховища інформації. Наприклад, якщо ми використовуємо RNN для генерації підпису до зображення, то така RNN може розглядати зображення частинами і на підставі кожної частини генерувати окремі слова. Робота Келвіна Ксу (Ху, et al., 2015), присвячена саме такому завданню, може бути гарною відправною точкою для тих, хто хоче вивчити такий механізм, як «увага». Дослідникам вже вдалося досягти вражаючих результатів із використанням цього принципу.

1. Руденко О. Г., Бодяньський Е. В. Штучні нейронні мережі: навч. посіб. Харків: ТОВ «Компанія СМІТ», 2005. 408 с. 2. Сім способів прогнозування часових рядів за допомогою Python. URL: <http://distrland.blogspot.com/2019/09/7-python.html> (дата звернення: 20.10.2022). 3. LSTM. URL: <https://habr.com/ru/company/wunderfund/blog/331310/> (дата звернення: 20.10.2022).