

УДК 004.942

DOI 10.24144/2616-7700.2022.1(40).109-125

Н. Бойко¹, В. Качмарик²

¹ Національний університет “Львівська політехніка”,
доцент кафедри Системи штучного інтелекту,
кандидат економічних наук
nataliya.i.boyko@lpnu.ua,
ORCID: <https://orcid.org/0000-0002-6962-9363>

² Національний університет “Львівська політехніка”,
студент 4-го курсу кафедри Системи штучного інтелекту,
viktor.kachmaryk.knm.2018@lpnu.ua
ORCID: <https://orcid.org/0000-0003-1074-2721>

**ПОБУДОВА МОДЕЛЕЙ ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ
РЯДІВ ЗАСТОСОВУЮЧИ МЕРЕЖІ ДОВГОСТРОКОВОЇ
ПАМ'ЯТІ**

Дослідження присвячено комплексному вивченню мереж довгої короткочасної пам'яті (ДКЧП). Дана архітектура є видом рекурентних нейронних мереж (РНМ), у яких з'єднання між вузлами утворюють граф, орієнтований у часі. У дослідженні проводиться огляд будови та роботи ДКЧП. Також здійснюється аналіз матеріалів і методів. У статті проводяться експерименти: будуються різні моделі для передбачення часових рядів. Аналізуються отримані результати дослідження. У цьому дослідженні будуть розглянуті цінність ДКЧП з практичної сторони у плані застосування, в залежності від виду вхідних даних, а також з теоретичної сторони, яка базуватиметься на розумінні будови блоку ДКЧП.

Ключові слова: довга короткочасна пам'ять, рекурентна нейронна мережа, машинне навчання, k-nearest neighbors, Long short term memory, Root mean square error.

1. Вступ. В своєму принципі РНМ можуть використовувати свою внутрішню пам'ять для зберігання представлень останніх вхідних подій у формі активацій (короткочасної пам'яті), а також для обробки довільних послідовностей входів. Це робить їх потенційно важливими для багатьох застосунків, зокрема розпізнавання мовлення, розпізнавання несеgmentованого неперервного рукописного тексту та композиції музики.

Не зважаючи на свої переваги, в таких традиційних РНМ, як зворотному поширенню в часі (Werbos, 1988) або рекурентному навчанні в реальному часі (Robinson & Fallside, 1987) існує проблема зникання або вибухання зворотно поширюваних похибок [1, 3]. Це в свою чергу спричиняє повільну роботу програми або взагалі її припинення, коли часові затримки між важливими подіями є занадто великими.

Для усунення даних недоліків, було запропоновано новий метод побудови РНМ: довгої короткочасної пам'яті (Sepp Hochreiter & Jürgen Schmidhuber, 1997) [1, 5]. Будова ДКЧП призначена для подолання цих проблем із зворотним потоком помилок. Дана мережа може навчитися асоціювати важливі події при часових затримках невідомої тривалості, які перевищують 1000 кроків, навіть у випадку шумних, нестиснутих вхідних послідовностей. Це досягається ефективним, заснованим на градієнті, алгоритмом для даної архітектури, який забезпечує постійний потік помилок через внутрішні стани спеціальних одиниць ДКЧП.

Ефективність даної мережі, а також її нечутливість до довжини часових проміжків між важливими подіями дає ДКЧП перевагу в численних застосуваннях над традиційними РНМ, прихованими марковськими моделями та іншими методами навчання послідовностей. Всі переваги мережі ДКЧП привели до її застосування в багатьох сферах інтелектуального аналізу даних, а саме: визначення аномалій в часових рядах, прогнозування: фінансового ринку, швидкості дорожнього руху, температури поверхні моря і т.д. Все більшої популярності набувають технології узагальнення по тексту чи відеозаписі, адже кожного дня генерується така кількість відео, що для їх перегляду необхідно понад 82 роки. Необхідність автоматичних інструментів для аналізу та розуміння великої кількості даних стверджують про актуальність обраної теми, вивчення якої в даній роботі надасть новий ракурс аналізу мереж ДКЧП.

Метою виконання даного дослідження полягає у ефективності мереж ДКЧП для застосувань навчання з досвіду з метою передбачення часових рядів, а саме пошук тенденції в даних часових рядах. Для здійснення заданої мети служать наступні **завдання**:

- З'ясувати способи використання ДКЧП в різних методах машинного навчання (МН).
- Розглянути приклади задач з використанням ДКЧП.
- Провести порівняння обробки та передбачення часових рядів з іншими відомими методами.

2. Огляд літературних джерел. Для розуміння аспектів та принципів мереж ДКЧП, аналізу та підсумовування існуючих підходів та методів, які використовуються при побудові даних мереж, було проведено аналіз літературних джерел, результати якого будуть використанні при проведенні досліджень, описаних в наступних розділах.

Стаття “Long Short-Term Memory” дозволила дізнатись детальний опис переваг ДКЧП та результати практичних досліджень, які дали можливість чітко побачити переваги та користь даної архітектури РНМ (Sepp Hochreiter & Jurgen Schmidhuber) [2, 14]. Зміна способу побудови блоку рекурентних нейронних мереж а також створення нового алгоритму заснованого на градієнті, створило цілком нову архітектуру, яка значно перевершує в точності традиційні РНМ, а також усуває проблему із зворотним потоком помилок, що робить її нечутливою до довжини часових проміжків між важливими подіями.

Якісно новим стало звернення до статті “Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks” [3, 5, 15], що дозволило дізнатись про покращену будову ДКЧП, де замість базової структури лінійного ланцюга, було представлено узагальнення ДКЧП до деревоподібної структури топології мережі (Kai Sheng Tai, Richard Socher, Christopher D. Manning) [4, 5, 10]. Це в свою чергу, призвело до того, що дана мережа перевершила всі існуючі системи та сильні базові показники звичайних ДКЧП.

3. Методи та засоби дослідження. Мережа ДКЧП є штучною нейронною мережею, яка завдяки своїй будові, яка містить вузли ДКЧП, не має проблеми зникання градієнту, на відміну від традиційних РНМ. Вузол ДКЧП – це вузол рекурентної нейронної мережі, який виділяється запам'ятовуванням значень для довгих, або коротких проміжків часу. Ключем до цієї здатності є наявність у будові вузла - рекурентного вентиля, який називають “забувальним”

[4, 8, 18]. Таким чином, значення, що зберігаються, не розплющуються ітеративно з плином часу, коли між значеннями великий проміжок, і члени градієнту не мають схильності розмиватися, коли для їх тренування застосовується зворотне поширення в часі.

Мережі ДКЧП часто втілюють у блоках, які складаються з вузлів ДКЧП. Така конструкція є типовою для глибоких багат шарових нейронних мереж, адже вона дає можливість реалізації на паралельному апаратному забезпеченні [2, 7, 17].

Типовий блок ДКЧП містить три вентиля (gates), які використовуються для керування потоку інформації до або з їхньої пам'яті. Ці вентиля реалізовані із застосуванням логістичних функцій, а саме сигмоїдної для обчислення значень між 0 та 1. Наприклад, "вхідний" вентиль керує мірою, до якої нове значення входить в пам'яті. "Забувальний" вентиль керує мірою, до якої значення залишається в пам'яті. А "вихідний" вентиль керує мірою, до якої значення в пам'яті використовується для обчислення активування виходу з блоку [6, 10, 12].

Єдині ваги, які є в блоці ДКЧП (W та U), використовуються для спрямування дії вентилів. Ці ваги застосовуються між значеннями, які надходять в блок (включно з вхідним вектором x_t та виходом з попереднього моменту часу h_{t-1}) та кожним із вентилів (Формула 1) [11, 13, 16]. Отже, блок ДКЧП визначає, яким чином підтримувати свою пам'ять як функцію від цих значень, яка мінімізує втрати. Дані блоки зазвичай тренують за допомогою алгоритму зворотного поширення в часі.

Традиційна ДКЧП із "забувальними" вузлами представлена на Рис. 1.

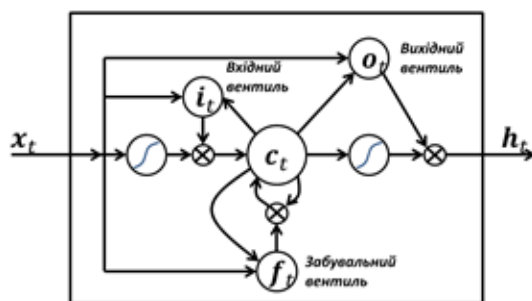


Рис. 1. Будова блоку ДКЧП

$$\begin{aligned}
 f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
 i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
 o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
 c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
 h_t &= o_t \circ \sigma_h(c_t),
 \end{aligned}
 \tag{1}$$

де \circ - позначає добуток Адамара (по-елементний матричний добуток);

Змінні (нижній індекс t вказує на час):

x_t - "вхідний" вектор;

h_t - "вихідний" вектор ($h_0 = 0$);

c_t – вектор стану комірки ($c_0 = 0$);

W, U і b – матриці та вектор параметрів;

f_t, i_t та o_t – вектори вентелів;

f_t – вектор “забувального” вентиля. Вага пам’ятання старої інформації;

i_t – вектор “вхідного” вентиля. Вага отримання нової інформації;

o_t – вектор “вихідного” вентиля, вирішує яку інформацію подавати на вихід.

Функції активації:

σ_g – в оригіналі є сигмоїдною функцією;

σ_c – в оригіналі є гіперболічним тангенсом;

σ_h – в оригіналі є гіперболічним тангенсом.

Експерименти. Аналіз часових рядів відноситься до аналізу зміни тенденції даних протягом певного періоду часу й має різноманітне застосування. Одним із таких додатків є прогнозування майбутньої вартості товару на основі його минулих значень. Прогнозування майбутніх цін на акції є, мабуть, найкращим прикладом такого застосування [5, 9, 10].

Метою виконання даного завдання є побудова мережі довгої короткочасної пам’яті з метою передбачення тенденцій даних за певний період часу та порівняння отриманих результатів з результатами, отриманими за допомогою інших методів машинного навчання [6, 11, 18].

Для програмної реалізації був обраний датасет Apple Inc. (AAPL), який був завантажений з сайту з finance.yahoo.com. Для навчання наших алгоритмів ми будемо використовувати ціни акцій Apple з 2 січня 2013 року по 26 лютого 2021 року. Для прогнозування ми будемо використовувати ціни акцій Apple за березень 2021 року. Зразок даних наведено на Рис. 2.

| | Date | Open | High | Low | Close | Volume |
|------|------------|------------|------------|------------|------------|-----------|
| 0 | 2013-01-02 | 19.779285 | 19.821428 | 19.343929 | 19.608213 | 560518000 |
| 1 | 2013-01-03 | 19.567142 | 19.631071 | 19.321428 | 19.360714 | 352965200 |
| 2 | 2013-01-04 | 19.177500 | 19.236786 | 18.779642 | 18.821428 | 594333600 |
| 3 | 2013-01-07 | 18.642857 | 18.903570 | 18.400000 | 18.710714 | 484156400 |
| 4 | 2013-01-08 | 18.900356 | 18.996071 | 18.616072 | 18.761070 | 458707200 |
| ... | ... | ... | ... | ... | ... | ... |
| 2048 | 2021-02-22 | 128.009995 | 129.720001 | 125.599998 | 126.000000 | 103916400 |
| 2049 | 2021-02-23 | 123.760002 | 126.709999 | 118.389999 | 125.860001 | 158273000 |
| 2050 | 2021-02-24 | 124.940002 | 125.559998 | 122.230003 | 125.349998 | 111039900 |
| 2051 | 2021-02-25 | 124.680000 | 126.459999 | 120.540001 | 120.989998 | 148199500 |
| 2052 | 2021-02-26 | 122.589996 | 124.849998 | 121.199997 | 121.260002 | 164320000 |

Рис. 2. Структура даних статистики цін на акції Apple з 2013 по 2021 роки

Як можемо побачити датасет складається з наступних 6 атрибутів:

- Date – дата (об’єкт дати і часу);
- Open – ціна під час відкриття (неперервне значення);
- High – найвища ціна (неперервне значення);
- Low – найнижча ціна (неперервне значення);
- Close – ціна під час закриття (неперервне значення);
- Volume – обсяг (неперервне значення).

При роботі з такими часовими рядами, як: ціни на акції певних компаній або ж ціни на криптовалюти нас найбільше цікавлять Open та Close ціни.

Як ми знаємо, ціни на фондовому ринку є вкрай непередбачуваними та нестабільними. Це означає, що в даних часових рядах немає відповідних закономірностей, які б дозволили майже ідеально змоделювати ціни протягом певного відрізка часу [13, 15].

Проте, ми принаймні можемо створити таку модель машинного навчання, завдяки якій ми зробимо такі передбачення, які корелюють з фактичною поведінкою даних. Іншими словами, нам потрібно передбачити лише коливання майбутньої ціни акції, а не її точні значення.

Оскільки ми будемо передбачувати ціну акцій під час закриття (close), то нас не цікавлять інші колонки, тому ми видалимо їх з дата сету (Рис. 3).

| | Date | Close |
|---|------------|-----------|
| 0 | 2013-01-02 | 19.608213 |
| 1 | 2013-01-03 | 19.360714 |
| 2 | 2013-01-04 | 18.821428 |
| 3 | 2013-01-07 | 18.710714 |
| 4 | 2013-01-08 | 18.761070 |
| 5 | 2013-01-09 | 18.467857 |
| 6 | 2013-01-10 | 18.696787 |
| 7 | 2013-01-11 | 18.582144 |

Рис. 3. Датасет після видалення зайвих колонок

Як можемо побачити на Рис. `refimage3`, після видалення зайвих колонок ми отримали датасет залежності між датою та ціною за одиницю акції Apple.

Якщо візуалізувати дану залежність: ціни за одиницю акції під час закриття (close) відносно дати, то можна побачити наступний Рис. `refimage4`.

Як можемо побачити на Рис. 4, тенденція є вкрай нелінійною, що робить важким завданням її(тенденцію) вловити, використовуючи дану інформацію. Ще одну річ, яку можемо побачити є те, що ціни за одиницю акції Apple близькі до 2021 року є значно вищими ніж ціни до 2019 року, що ускладнює наше завдання в знаходженні кореляції між даними. Саме для такого завдання нам пригодиться потужність мережі довгої короткочасної пам'яті, а також ми зможемо на наглядному прикладі переконатися в перевагах даних мереж відносно інших методів машинного навчання [7, 9].

Для того, щоб вивчити вплив прогнозуючих сил у фінансових часових рядах ми побудували одну традиційну модель часових рядів, дві базові моделі машинного навчання, а також дві моделі глибинного навчання. Даніми моделями є:

- Рухоме середнє;
- Лінійна регресія;
- K-найближчих сусідів (KNN);
- Стандартна мережа довгої короткочасної пам'яті (LSTM);
- Розширена мережа довгої короткочасної пам'яті, яка має кілька додаткових прихованих шарів ДКЧП (Stacked-LSTM).

Для оцінки даних моделей ми будемо використовувати середньоквадратичну помилку (RMSE) в якості метрики ефективності моделі, а також візуалізовані

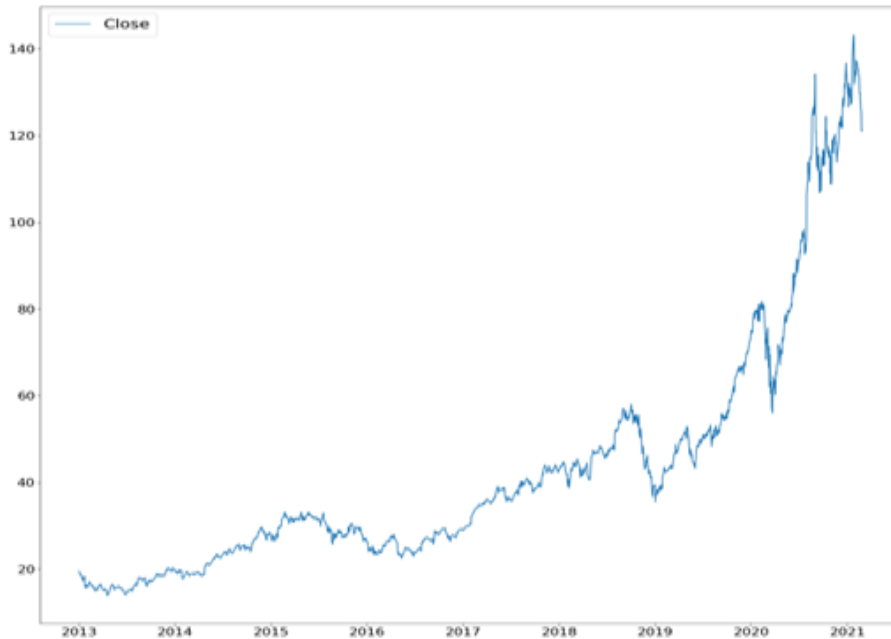


Рис. 4. Тенденція ціни на акції Apple

отримані дані для кращого розуміння точності відтворення тенденції ціни на акції Apple.

Програмна реалізація даних моделей виконувалась за допомогою високорівневої мови програмування Python. Для побудови базових моделей машинного навчання була використана бібліотека `scikit-learn`, а для побудови моделей глибокого навчання – бібліотеки `keras` та `tensorflow`.

“Середнє” є однією з базових арифметичних дій, яку ми часто використовуємо в повсякденному житті. Наприклад знаходження середньої температури за кілька останніх днів, щоб знати приблизну температуру сьогодні. Отже це є хорошою відправною точкою в використанні її на нашому датасеті, щоб передбачити ціну акцій Apple.

Передбачена ціна для кожного дня буде середнім від набору раніше спостережених значень. Замість того, щоб використовувати просте середнє, ми застосуємо алгоритм рухомого середнього, який використовує найновіший набір значень для кожного передбачення [8]. Іншими словами, для кожного наступного кроку передбачувані значення враховуються, вилучаючи найстаріше спостережуване значення з набору. Для більш легкого розуміння даний алгоритм зображений на Рис. 5.

Застосувавши модель рухомого середнього, ми отримали наступні результати: Середньоквадратична помилка дорівнює 7.654, що є не надто хорошим значенням.

Як ми можемо побачити на Рис. 6, передбачення, зроблені за допомогою нашої модель, не корелюють з фактичною поведінкою, що вказує на те, що алгоритм не впорався з відшукуванням тенденції на ціну акцій Apple. Перейдемо до моделей машинного навчання.

Найпростішим алгоритмом машинного навчання, який ми можемо застосувати до даних такого типу є лінійна регресія. Модель лінійної регресії повертає

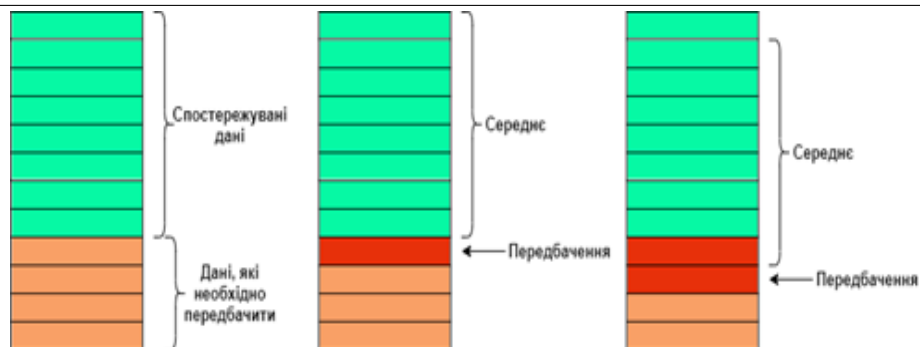


Рис. 5. Алгоритм рухомого середнього

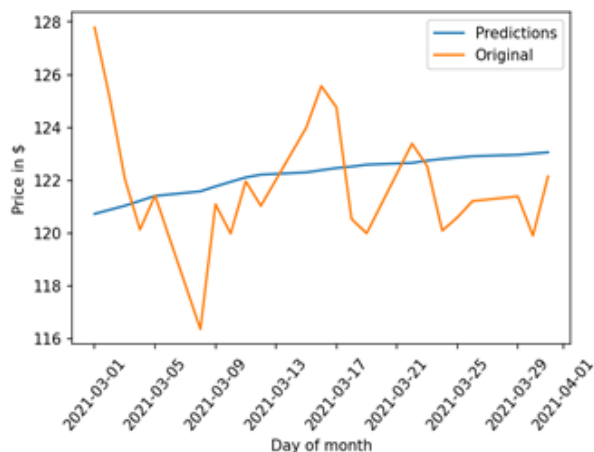


Рис. 6. Передбачення ціни на акції Apple при використанні рухомого середнього

рівняння, яке визначає взаємозв'язок між незалежними змінними та залежною змінною.

Рівняння для лінійної регресії можна зобразити наступним чином (Формула 2):

$$Y = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n, \quad (2)$$

де x_1, x_2, \dots, x_n представляють незалежні змінні, тоді як коефіцієнти $\theta_1, \theta_2, \dots, \theta_n$ представляють ваги.

Для вирішення проблеми ми не маємо набору незалежних змінних, а лише дати і відповідні їм ціни на акції Apple. Тому ми використаємо стовпець дати для вилучення таких характеристик, як – день тижня, початок/кінець місяця, рік тощо, за допомогою бібліотеки `fastai`.

Як можемо побачити на Рис. 7, новоутворений датасет окрім дати та ціни акцій під час закриття (`close`) містить низку додаткових незалежних змінних таких як: день тижня/місяця/року, номер тижня, початок/кінець місяця/четверті/року, за допомогою яких ми будемо будувати нашу модель для передбачень.

| | Close | Year | Month | Week | Day | Dayofweek | Dayofyear | Is_month_end |
|------------|----------------|----------------|------------------|-------------|---------------|-----------|-----------|--------------|
| Date | | | | | | | | |
| 2013-01-02 | 19.608213 | 2013 | 1 | 1 | 2 | 2 | 2 | False |
| 2013-01-03 | 19.360714 | 2013 | 1 | 1 | 3 | 3 | 3 | False |
| 2013-01-04 | 18.821428 | 2013 | 1 | 1 | 4 | 4 | 4 | False |
| 2013-01-07 | 18.710714 | 2013 | 1 | 2 | 7 | 0 | 7 | False |
| 2013-01-08 | 18.761070 | 2013 | 1 | 2 | 8 | 1 | 8 | False |
| | Is_month_start | Is_quarter_end | Is_quarter_start | Is_year_end | Is_year_start | | | |
| | False | False | False | False | False | | | |
| | False | False | False | False | False | | | |
| | False | False | False | False | False | | | |
| | False | False | False | False | False | | | |
| | False | False | False | False | False | | | |

Рис. 7. Утворений датасет з додатковими змінними

Застосувавши модель лінійної регресії, ми отримали наступні результати: Середньоквадратична помилка дорівнює 46.458, що однозначно не є хорошим показником.



Рис. 8. Передбачення ціни на акції Apple при використанні лінійної регресії

Як можемо побачити на Рис. 8, через те, що значення цін сильно відрізняються між роками, наша модель перенавчається відносно колонок дати та місяця, тому наші передбачення не дають суттєвого результату.

Замість того, щоб враховувати попередні значення з точки прогнозування, модель враховує значення з аналогічної дати місяць тому, або тієї ж дати/місяця проте рік тому.

Якщо ж спробувати нормалізувати дані по різних проміжкам часу (batch normalization) і побудувати модель по отриманих даних, то ми отримаємо наступний результат: Середньоквадратична помилка дорівнює 6.61, що вже є кращим показником, ніж минулий результат, проте не є достатньо хорошим.

Як можемо побачити на Рис. 9, ми отримали однозначно кращий результат на нормалізованих даних, в порівнянні з передбаченнями зроблених по звичай-

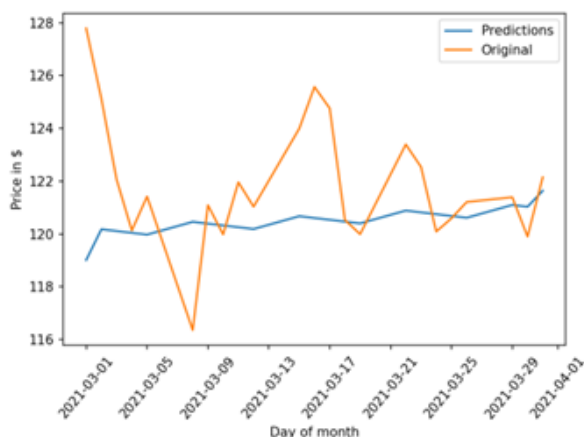


Рис. 9. Передбачення ціни на акції Apple (нормалізовані) при використанні лінійної регресії

них даних. Проте як ми можемо побачити на графіку, передбачення, зроблені за допомогою нашої моделі, не корелюють з фактичною поведінкою, а отже наша модель не справилась з поставленим завданням.

Так само як і лінійна регресійна модель, даний алгоритм базується на незалежних змінних – знаходить схожість між новими та старими даними.

Алгоритм KNN може бути використаний як для задач класифікації, так і для регресійних проблем. Даний алгоритм використовує “подібність ознак” для прогнозування значень будь-яких нових даних. Це означає, що новим точкам присвоюються значення на основі того, наскільки вони схожі на точки в навчальному наборі.

Даний алгоритм складається з трьох кроків:

- Першим кроком є розрахунок відстані між новою точкою і всіма точками в тренувальному датасеті. Існують різні методи розрахунку даних відстаней, серед яких найбільш відомими є: Евклідова відстань, Манхеттенська метрика (для безперервних даних) та відстань Хеммінга (для категоріальних даних).
- Другим кроком є вибір значення k , яке визначає кількість найближчих сусідів, які ми будемо розглядати, при присвоєнні значення будь-якому новому спостереженню. Для його виконання була використана функція Grid Search CV, за допомогою якої ми змогли підібрати найкраще значення k для нашої проблеми.
- Третім кроком є передбачення значення нового спостереження, шляхом розрахунку середнього між значеннями k найближчих сусідів, визначених в другому кроці.

Як і для лінійної регресії, для побудови моделі, заснованої на алгоритмі KNN, нам необхідно, щоб наші дані мали набір незалежних змінних, тому ми використовуємо датасет, утворений для побудови лінійної регресії (Рис. 10).

Застосувавши модель К-найближчих сусідів, ми отримали наступні результати: Середньоквадратична помилка дорівнює 103.02, що, як і в випадку лінійної регресії, також є однозначно не хорошим показником.

Як можемо побачити на Рис. 10, через те, що значення цін сильно відрізня-

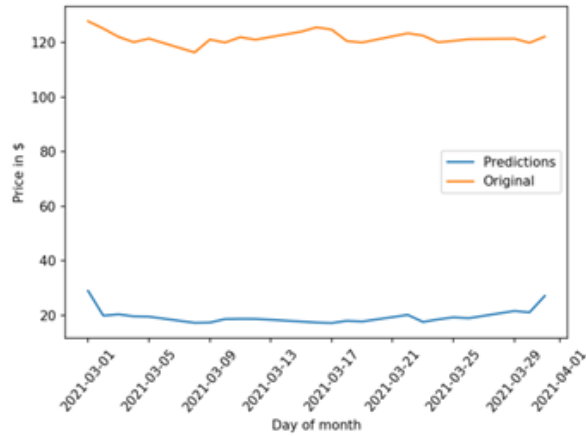


Рис. 10. Передбачення ціни на акції Apple при використанні KNN

ються між роками, наша модель, заснована на алгоритмі KNN, так само як і лінійна регресія, перенавчається відносно колонок дати та місяця, тому наші передбачення не дають суттєвого результату.

Спробуємо, як і для лінійної регресії, нормалізувати дані по різних проміжкам часу (batch normalization) і побудувати модель по отриманих даних: Середньоквадратична помилка дорівнює 6.74, що вже є кращим показником, ніж минулий результат, проте не є достатньо хорошим.

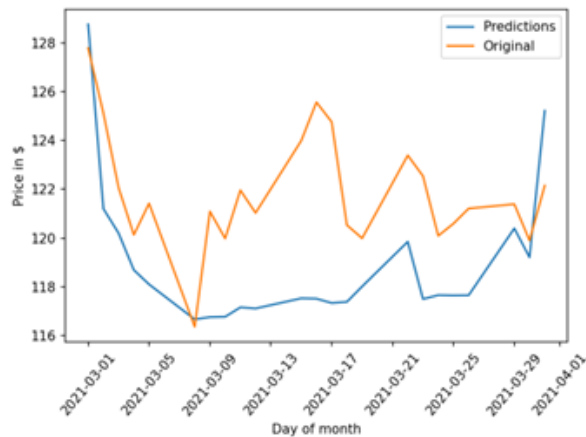


Рис. 11. Передбачення ціни на акції Apple (нормалізовані) при використанні KNN

Як можемо побачити на Рис. 11, ми отримали однозначно кращий результат на нормалізованих даних, в порівнянні з передбаченнями зроблених по звичайних даних.

Також можемо побачити, що на відміну від лінійної регресії, наша модель, заснована на алгоритмі KNN, наближено відтворила тенденцію на ціну акцій Apple. Проте, отримані передбачення не надто добре корелюють з фактичною поведінкою, особливо на середньому проміжку, тому спробуємо добитись кращого результату, за допомогою побудови інших моделей.

Для обох моделей LSTM для прихованих шарів використовувалась функція

активації гіперболічного тангенса, а для вихідних шарів – сигмоїдна функція активації, було практично доведено, що ці функції активації показують кращі результати для даного часового ряду, ніж інші функції активації. Також ми застосували 20% відкидання (dropoutrate) для кожного прихованого шару, як звичайний метод регуляризації, щоб уникнути проблеми перенавчання. Ми використали оптимізаційний алгоритм Адама для вивчення параметрів, а середню квадратичну помилку- як функцію втрат.

Щоб збільшити швидкість навчання та збільшити збіжність, ми використовуємо так звані “mini-batches” під час тренування. Ми спробували велику кількість розмірів “mini-batches” і дослідили, що найкращим розміром, який слід використовувати є 30, оскільки більший розмір не гарантує збіжність і є великий шанс потрапити у сідлову точку.

Також для покращення точності передбачення наших мереж, а також пришвидшення тренування їх тренування, ми застосували функцію GridSearch, завдяки якій ми підбрали найбільш підходящі гіперпараметри для наших моделей.

Через високу стохастичність фінансових часових рядів ми побудуємо дві різні моделі ДКЧП: стандартну мережу ДКЧП та розширену мережу довгої короткочасної пам'яті, яка має кілька додаткових прихованих шарів ДКЧП, а також порівняємо їх продуктивність. Очікувалось, що розширена мережа ДКЧП може краще вловити тенденцію ціни на акції Apple через свою більш складну структуру. Однак наші експерименти показали протилежні результати, що ми обговоримо пізніше.

Застосувавши стандартну мережу довгої короткочасної пам'яті, ми отримали наступні результати: Середньоквадратична помилка дорівнює 2.617, що є достатньо малим значенням, що вказує на високу подібність даних.

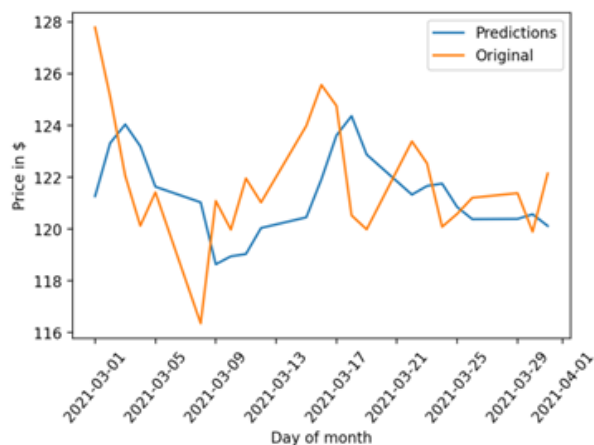


Рис. 12. Передбачення ціни на акції Apple при використанні стандартної мережі ДКЧП

Як можемо побачити на Рис. 12, на відміну від всіх попередніх моделей, наша стандартна мережа ДКЧП, достатньо точно відтворила тенденцію на ціну акцій Apple, а саме передбачила, де наша ціна буде підніматись, а де опускатись. Також ми можемо дізнатись приблизні міри падіння і піднімання ціни і робити відповідні висновки з цього. Отримані передбачення добре корелюють

з фактичною поведінкою, що вказує на те, що наша мережа добре впоралась з поставленим завданням.

Розширена мережа довгої короткочасної пам'яті, яка має кілька додаткових прихованих шарів ДКЧП (Stacked-LSTM).

Для побудови даної розширеної моделі ми добавили 3 додаткових прихованих шарів ДКЧП. Застосувавши розширену мережу довгої короткочасної пам'яті, ми отримали наступні результати: Середньоквадратична помилка дорівнює 3.14, що, як і в випадку зі стандартною мережею, є достатньо малим значенням, проте одразу можна зробити висновки, що дана мережа гірше передбачила ціну на акції Apple.

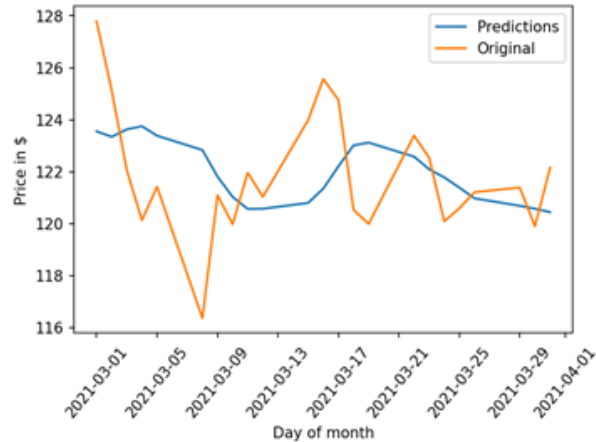


Рис. 13. Передбачення ціни на акції Apple при використанні розширеної мережі ДКЧП

Як можемо побачити на Рис. 13, дана розширена мережа не так точно відтворює тенденцію ціни на акції Apple, як стандартна мережа, проте в порівнянні з іншими методами, застосованими в даній роботі вона показала достатньо хороші результати. За допомогою даної мережі ми досі можемо передбачувати чи ціна підніметься, чи знизиться, проте не можемо робити передбачення про міру падіння/піднімання ціни. Отримані передбачення, за допомогою даної розширеної мережі, хоч і не настільки добре як зі стандартною мережею, проте корелюють з фактичною поведінкою, що вказує на те, що наша мережа достатньо добре впоралась з поставленим завданням.

Результати дослідження. Основною метою даної роботи було дослідження ефективності мереж ДКЧП в передбаченні часових рядів, а саме пошук тенденції в даних часових рядах, а також порівняння її ефективності з іншими відомими методами.

Для виконання даного завдання було побудовано п'ять моделей часових рядів, застосувавши наступні методи:

1. Рухоме середнє (традиційна модель часових рядів)

Алгоритмом створення даної моделі є: передбачена ціна для кожного дня буде середнім від набору раніше спостережених значень (вилучаючи найстаріше спостережуване значення з набору), що можна представити наступною формулою 3:

$$\bar{a}_{SM} = \frac{x_{n-M} + x_{n-(M-1)} + \dots + x_{(n-1)}}{M} = \frac{1}{M} \sum_{i=0}^{M-1} x_{n-(M-i)}, \quad (3)$$

де x_n – значення за певний проміжок часу (в нашому випадку за один день);

M – проміжок часу, відносно якого ми будемо вираховувати середнє.

Перевагою даного алгоритму є швидке виконання передбачення (0.7 с).

Недоліком даного алгоритму є те, що передбачення, зроблені за допомогою нього, показують лише наближену ціну на акції, яка не корелює з фактичними даними.

2. Лінійна регресія (базова модель МН)

Рівняння для лінійної регресії представлено у формулі 3. Перевагою використання даної моделі є швидке виконання передбачення (0.003 с).

Недоліками її використання є: передбачення, зроблені за допомогою неї, як і в випадку з алгоритмом рухомого середнього, показують лише наближену ціну на акції, яка не корелює з фактичними даними.

Для того, щоб уникнути проблеми з перенавчанням, потрібно додатково нормалізувати дані по певним проміжкам часу (batch normalization).

3. К-найближчих сусідів (базова модель МН)

Алгоритмом створення даної моделі є:

- На першому кроці потрібно розрахувати відстані між новою точкою і всіма точками в тренувальному датасеті.
- Другим кроком є знаходження k найближчих сусідів (k -визначене наперед).
- Третім кроком є передбачення значення нового спостереження, шляхом розрахунку середнього між значеннями k найближчих сусідів, визначених в другому кроці.

Перевагами використання даного алгоритму є:

- Швидке виконання передбачення (0.004 с).
- Наближене відтворення тенденції на ціну.

Недоліками використання даної моделі є:

- Як і в випадку з лінійною регресією, для того, щоб уникнути проблеми з перенавчанням, потрібно додатково нормалізувати дані по певним проміжкам часу (batch normalization).
- Хоч тенденція і є наближено відтвореною, проте недостатньо точно, а отже ми можемо використовувати її лише як орієнтир для подальших досліджень.

4. Стандартна мережа ДКЧП (модель глибинного навчання)

Перевагою використання даної мережі є те, що в порівнянні з іншими методами, дана мережа відтворює тенденцію на ціну досить точно, що і є найважливішим критерієм в даній роботі.

Хоча час тренування склав 109 с, проте це не є недоліком, адже це відносно невеликий час для тренування моделей МН.

5. Розширена мережа довгої короткочасної пам'яті, яка має кілька додаткових прихованих шарів ДКЧП (Stacked-LSTM)

Перевагою використання даної мережі є те, що дана мережа досить добре відтворює тенденцію на ціну (краще впоралася з цим, порівнюючи з традиційною моделлю часових рядів і базовими моделями МН).

Недоліками використання даної мережі є:

- Не настільки хороша точність відтворення тенденції в порівнянні зі стандартною мережею.
 - Довший час тренування, в порівнянні зі стандартною мережею ДКЧП (911 с).
- Отже, давайте підсумуємо усі результати, отримані при виконанні попереднього розділу у наступній таблиці 1.

Таблиця 1.

Результати передбачень різними моделями

| Модель | Час виконання, с | RMSE | Кореляція передбачених даних з фактичними |
|------------------------|------------------|-------|---|
| Рухоме середнє | 0.7 | 7.654 | Відсутня кореляція (Рис. 6) |
| Лінійна регресія | 0.003 | 6.61 | Відсутня кореляція (Рис. 8) |
| KNN | 0.004 | 6.74 | Часткова кореляція (Рис. 11) |
| Стандартна мережа ДКЧП | 109.3 | 2.617 | Хороша кореляція (Рис. 12) |

В Табл. 1 ми можемо переглянути підсумок результатів нашого дослідження. Давайте перейдемо до аналізу отриманих результатів, а також зробимо попередні висновки.

Обговорення результатів. Аналізуючи результати, отримані при виконанні попереднього розділу можна зробити наступні висновки:

Через те, що наші дані мають високу стохастичність, а будь яка прихована залежність є непрямою дана традиційна модель часових рядів – рухоме середнє не впоралась з покладеним завданням – передбачити тенденцію майбутньої ціни на акції Apple. Даний алгоритм передбачує лише наближену майбутню ціну, а тому не підходить для вирішення нашої проблеми (передбачення тенденції на ціну).

Як і рухоме середнє, лінійна регресія не впоралась з покладеним завданням – передбачити тенденцію майбутньої ціни на акції Apple. Недоліком використання даного алгоритму для нашої проблеми є вимога до даних, а саме: наявність незалежних змінних. Для цього ми додатково створювали ряд незалежних змінних, які дістали з колонки дати/час. Проте давайте більш докладно розглянемо чому лінійна регресія не впоралась з даним завданням:

- Оскільки дані, які були використані та містять високу стохастичними, а також будь яка прихована залежність є непрямою, лінійна регресія просто не може її відшукати.
- Як вже було сказано раніше, ціна на акції Apple за різних років значно відрізняються, через що наша модель перенавчається відносно колонок дати та місяця. Тому, щоб зробити нормальне передбачення нам потрібно застосувати “batch normalization” для усунення даної проблеми.

Низка даних обмежень, а також той факт, що дана модель не відтворила тенденцію робить її не підходящою для заданої проблеми.

На відміну від двох попередніх алгоритмів, алгоритм KNN наближено відтворив тенденцію на ціну акцій Apple, проте на певному проміжку дані взагалі

не корелювали з фактичними. Тому передбачення, отримані за допомогою даного алгоритму, ми можемо лише використовувати як довідку для передбачень, зроблених за допомогою методів глибинного навчання.

Як вже було сказано у попередньому розділі, очікувалось, що розширена мережа ДКЧП може краще вловити тенденцію ціни на акції Apple через свою більш складну структуру, проте наші експерименти показали протилежні результати. В порівнянні з іншими моделями, застосованими у цій роботі, наша розширена мережа ДКЧП досить добре відтворила тенденцію ціни на акції Apple, проте не настільки добре, як стандартна мережа ДКЧП.

Як правило, чим “глибша” нейронна мережа (та, яка має більше прихованих шарів), тим краще вона здатна пояснити більш складні проблеми, на відміну від одношарової нейронної мережі. Однак відкриттям, яке ми виявили був той факт, що розширена мережа ДКЧП не перевершує стандартну мережу ДКЧП в контексті прогнозування ціни акцій, а тобто проблем, пов’язаних з часовими рядами. Натомість, розширена мережа ДКЧП показала меншу продуктивність від стандартної мережі. Доведено, що більш складний представник нейронної мережі не обов’язково покращує якість передбачення.

Даний результат ми можемо пояснити наступними причинами:

- Більш складна будова нейронної мережі спричиняє проблему перенавчання, а тобто розширена мережі ДКЧП погано узагальнює приховані залежності у часових рядах, через більшу кількість параметрів, які вона розглядає.
- Розширена мережа ДКЧП більше підходить для розв’язування проблем класифікації, а не неперервних часових рядів, таких як ціни на акції.

Після підсумування та обговорення всіх результатів доходимо до висновку, що найкращою моделлю машинного навчання для передбачення таких стохастичних часових рядів, як ціни на акції, є модель глибинного навчання – стандартна мережа ДКЧП.

4. Висновки. В даній роботі було детально досліджено одну з архітектур рекурентних нейронних мереж – мережу довгої короткочасної пам’яті. На відміну від звичайних РНМ, мережа ДКЧП добре підходить для навчання з досвіду з метою передбачення часових рядів в умовах, коли вони є стохастичними або ж приховані залежності є непрямыми і їх складно віднайти.

Для того, щоб показати усю потужність мереж ДКЧП ми обрали датасет цін на акції Apple з вкрай нелінійною тенденцією, що ускладнило наше завдання – передбачити її (тенденцію) за певний проміжок часу.

Загалом для максимально точної та об’єктивної оцінки ми запропонували, розробили, натренували й протестували 5 моделей, а саме: плаваюче середнє, лінійну регресію, KNN, стандартну мережу ДКЧП та розширену мережу ДКЧП з кількома додатковими прихованими шарами ДКЧП. Найкраще себе показала стандартна мережа ДКЧП, яка достатньо точно відтворила майбутню тенденцію на ціну акцій Apple й значно перевершила у цьому усі інші моделі.

Також ми зробили відкриття й показали, що незважаючи на більш складну структуру моделі розширеної мережі ДКЧП у порівнянні зі стандартною мережею, розширена мережа не перевершує стандартну в контексті прогнозування ціни акцій, а тобто проблем, пов’язаних з часовими рядами, через можливість перенавчання.

Також було досліджено, що при побудові будь якої мережі ДКЧП, важливо

звернути увагу на правильний підбір параметрів моделі, адже навіть невелика її зміна може як значно покращити, так і значно погіршити дану модель.

Подальші дослідження слід проводити зі ще більш стохастичними даними, такими як: ціни на криптовалюти і поглянути наскільки точно наша мережа зможе відтворити тенденцію в такому часовому ряді. Також хорошим варіантом буде побудова більш складної структури мережі ДКЧП, а саме мережа ДКЧП з механізмом уваги й перевірити чи зможе вона краще впоратись з завданням передбачення тенденції в часовому ряді.

Boyko N., Kachmaryk V. Building models for time sequence forecasting using long-term memory networks .

The study is devoted to a comprehensive analysis of long short-term memory (LSTM) networks. This architecture is a type of recurrent neural network (RNN), in which the connections between nodes form a time-oriented graph. In this study, we review the structure and performance of the LSTM networks. Materials and methods are also analyzed. This paper conducts experiments, such as building different models to predict time series. The obtained research results are also analyzed. In this study, we will consider both the value of LSTM from a practical point of view in terms of exploitation, depending on the type of input data, and from a theoretical point of view, which will be based on understanding the structure of the LSTM block.

Keywords: Long short-term memory, recurrent neural network, machine learning, k-Nearest Neighbors, Root mean square error.

References

1. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735> [in English].
2. Brownlee, J. A. (2022). Gentle Introduction to Long Short-Term Memory Networks by the Experts, Machine Learning Mastery. Retrieved from <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/> [in English].
3. Brownlee, J. (2021). Mini-Course on Long Short-Term Memory Recurrent Neural Networks with Keras, Machine Learning Mastery. Retrieved from <https://machinelearningmastery.com/long-short-term-memory-recurrent-neural-networks-minicourse/> [in English].
4. Brownlee, J. (2021). Stacked Long Short-Term Memory Networks, Machine Learning Mastery. Retrieved from <https://machinelearningmastery.com/stacked-long-short-term-memory-networks/> [in English].
5. Özlü, A. (2021). Long Short Term Memory (LSTM) Networks in a nutshell, Medium. Retrieved from <https://ahmetozlu93.medium.com/long-short-term-memory-lstm-networks-in-a-nutshell-363cd470ccac> [in English].
6. Malik, U. (2021). Time Series Analysis with LSTM using Python's Keras Library, StackAbuse. Retrieved from <https://stackabuse.com/time-series-analysis-with-lstm-using-pythons-keras-library/> [in English].
7. Ganegedara, Th. (2021). Stock Market Predictions with LSTM in Python. Retrieved from <https://www.datacamp.com/community/tutorials/lstm-python-stock-market> [in English].
8. Zou, Zh., & Qu, Z. Using LSTM in Stock prediction and Quantitative Trading. *CS230: Deep Learning* [in English].
9. Singh, A. (2021). Stock Prices Prediction Using Machine Learning and Deep Learning Techniques, Analytics Vidhya. Retrieved from <https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learning-and-deep-learning-techniques-python/> [in English].
10. Hongju, Y., & Hongbing, O. (2018). Financial time series prediction based on deep learning. *Wireless Personal Communications*, 102(2), 683–700. <https://doi.org/10.1007/s11277-017-5086-2> [in English].
11. Box, G. E., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. (2015) Time Series Analysis: Forecasting and Control. John Wiley & Sons, Hoboken, 715 [in English].

12. Boyko, N. (2016). Application of mathematical models for improvement of “cloud” data processes organization. *Mathematical Modeling and Computing: scientific journal "Computational problems elektotekhniki"*, 3(2), 111–119. <https://doi.org/10.23939/mmc2016.02.111> [in English].
13. Boyko, N., & Shakhovska, N. (2018). Prospects for Using Cloud Data Warehouses in Information Systems, *International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018*, 2, 136–139. <https://doi.org/10.1109/STC-CSIT.2018.8526745> [in English].
14. Taylor, G. W. (2009). *Composable, distributed-state models for high-dimensional time series*. Toronto: University of Toronto [in English].
15. Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv:1409.1556 [in English].
16. Sarikaya, R., Hinton, G. E., & Deoras, A. (2014). Application of Deep Belief Networks for natural language understanding. *ACM Transactions on Audio Speech and Language Processing*, 22(4), 778–784 [in English].
17. Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. Neural Networks. In *Proceedings of the IEEE-INNS-ENNS international joint conference on IJCNN 2000*, 348–353 [in English].
18. Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, 38, 10389–10397 [in English].

Одержано 15.04.2022