

УДК 004.8:336.76

Ю.К. Тараненко, Н.О. Різун, Р.М. Мілов

ПРОГНОЗУВАННЯ КОЛИВАНЬ ЦІН НА ФОНДОВОМУ РИНКУ З ВИКОРИСТАННЯМ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

Розглянуто актуальну наукову та практичну проблему розробки моделі прогнозування цін на фондовому ринку із використанням нейронних мереж та вейвлетної фільтрації. Запропоновано програмну реалізацію цієї моделі за допомогою мови обробки статистичних даних міжнародного стандарту – R-language 3.0.2. Отримано та проаналізовано результати прогнозування на базі моделі, що розроблено, а також досліджено ступінь значущості та точності моделі.

Ключові слова: аналіз динаміки цін, нейронні мережі, вейвлети, R-language.

Постановка проблеми. У наш час стрімко поширюється використання інформаційних систем алгоритмічної торгівлі, призначених для автоматичного керування інвестиційними портфелями в умовах біржі в реальному часі. Сьогодні, за різними оцінками, більш ніж 70% операцій на біржах виконуються саме системами алгоритмічної торгівлі. Системи алгоритмічної торгівлі активно використовуються як великими інвестиційними компаніями, так і приватними особами, що інвестують свої кошти, створюючи портфелі фінансових інструментів.

З іншого боку, завдяки розвитку обчислювальної техніки, пов'язаному із зростанням потужності обчислювальних пристроїв і оптимізацією обчислювальних алгоритмів, з'явилась можливість перейти від застарілих систем, які використовують класичні аналітичні інструменти, до систем на основі штучного інтелекту, які хоча й потребують більше обчислювальних ресурсів, але й мають можливість швидко адаптуватись до змін ринку.

Дослідження наукових праць із питання, що розглядається. Інформація про ціну активів на біржі в заданий період звичайно подається у вигляді набору з чотирьох значень: ціна на *початок* періоду – ціна відкриття, ціна на *кінець* періоду – ціна закриття, *максимальна* та *мінімальна* ціна за період. Послідовність значень кожного параметра у часі створює ряд динаміки, який може аналізуватися для пошуку необхідної інформації.

Задача прогнозування зводиться до знаходження деякої функції $F()$, яка ставить у відповідність значенню сигналу, що досліджується, в певний період часу значення сигналу у періоди, що йому передували:

$$x_t = F(x_{t-1}, x_{t-2}, \dots, x_{t-n}). \quad (1)$$

Традиційно в аналізі біржових котирувань для цієї мети використовуються різні форми лінійних адаптивних фільтрів, які мають загальний вигляд:

$$x_t = a_1 x_{t-1} + a_2 x_{t-2} + \dots + a_n x_{t-n}. \quad (2)$$

Прикладами таких фільтрів є різні форми рухомого середнього:
 – просте ковзне середнє:

$$x_t = \frac{1}{n}(x_{t-1} + x_{t-2} + \dots + x_{t-n}); \quad (3)$$

– експоненційне ковзне середнє:

$$x_t = x_{t-1} + (1 - \alpha)x_{t-2} + (1 - \alpha)^2 x_{t-3} \dots + (1 - \alpha)^{n-1} x_{t-n}. \quad (4)$$

Наступним кроком розвитку лінійних адаптивних фільтрів були фільтри із добором параметрів, дослідження яких були започатковані Н. Вінером [1]. Вони полягають у такому: нехай лінійний адаптивний фільтр має вигляд:

$$y(t) = \sum_{i=1}^n a_i x_{t-i}. \quad (5)$$

Розрахуємо похибку:

$$\varepsilon = \sum_t x(t) - y(t). \quad (6)$$

Навчити фільтр означає знайти такі значення коефіцієнтів a , за яких помилка ε набуває *мінімального* значення.

Ця задача може бути вирішена із використанням класу моделей, відомих як «штучні нейрони». Ідея «штучного нейрону» прийшла з біології, де вона використовувалась для моделювання роботи нервової системи, але згодом ці моделі знайшли широке застосування у галузі систем машинного навчання та штучного інтелекту. Схему штучного нейрона з n входами наведено на рис. 1.

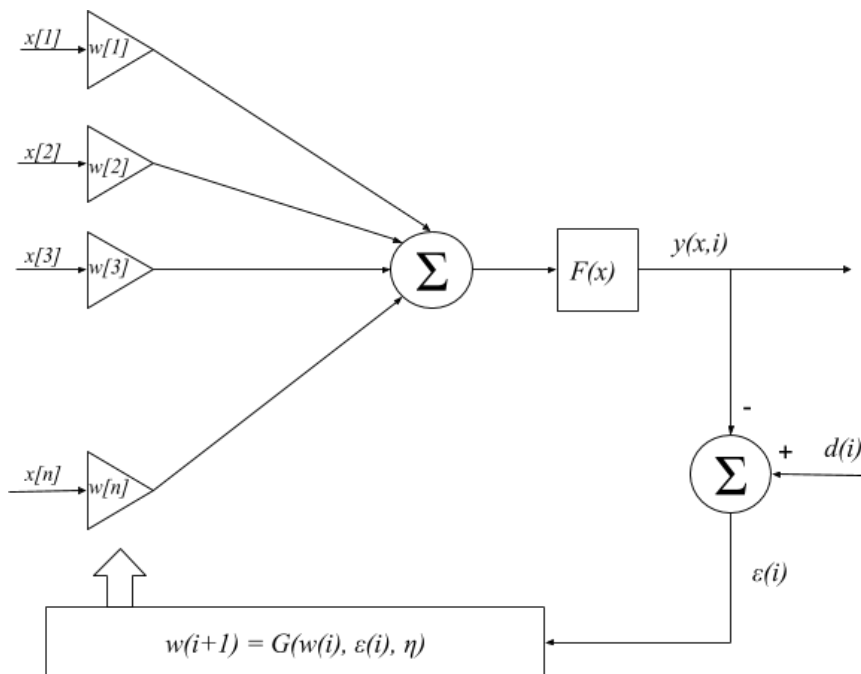


Рис. 1. Модель штучного нейрона

Вхідний n -компонентний сигнал, представлений вектором x , подається на синапси, де кожен елемент вектора x помножується на вагу синапсу, представлену вектором w . Сигнали з виходу синапсів накопичуються і використовуються як аргументи деякої функції $F(x)$, яка називається *функцією активації*. Як функцію активації при вирішенні задач класифікації використовують різні форми *порогових* функцій або, навпаки, при вирішенні задач регресії функція активації може не використовуватись. На виході функції активації ми маємо значення виходу моделі для вектора x .

Але головна ідея «штучних нейронів» полягає в тому, що перед використанням моделі її треба навчити. Навчання відбувається ітераційно: на кожному i -му кроці алгоритму розраховується помилка, $\varepsilon(i) = y(i) - d(i)$, де $d(i)$ – еталонний сигнал, який використовується для навчання. На наступному кроці розраховуються нові значення вектора w із урахуванням: помилки $\varepsilon(i)$, значення w на поточній ітерації, та деякого параметра η , що називається *кроком корекції*.

Процес продовжується, поки величина помилки $\varepsilon(i)$ не стане достатньо малою, чи не буде зроблено наперед задану кількість кроків. У останньому випадку вважається, що рішення задачі за наявного набору даних та кроку корекції не існує.

Подальший розвиток ідеї – поєднання нейронів у нейронні мережі (рис. 2). При побудові нейронної мережі нейрони поєднуються в групи – шари – так, що виходи нейронів кожного шару є входами нейронів наступного шару. Шари, що безпосередньо не з'єднані з входом чи виходом мережі, називаються *прихованими*. Для навчання нейронних мереж використовуються алгоритми з принципом, аналогічним алгоритмам навчання поодиноких нейронів, – зворотного розповсюдження помилки, еластичного зворотного розповсюдження помилки [2] та ін.

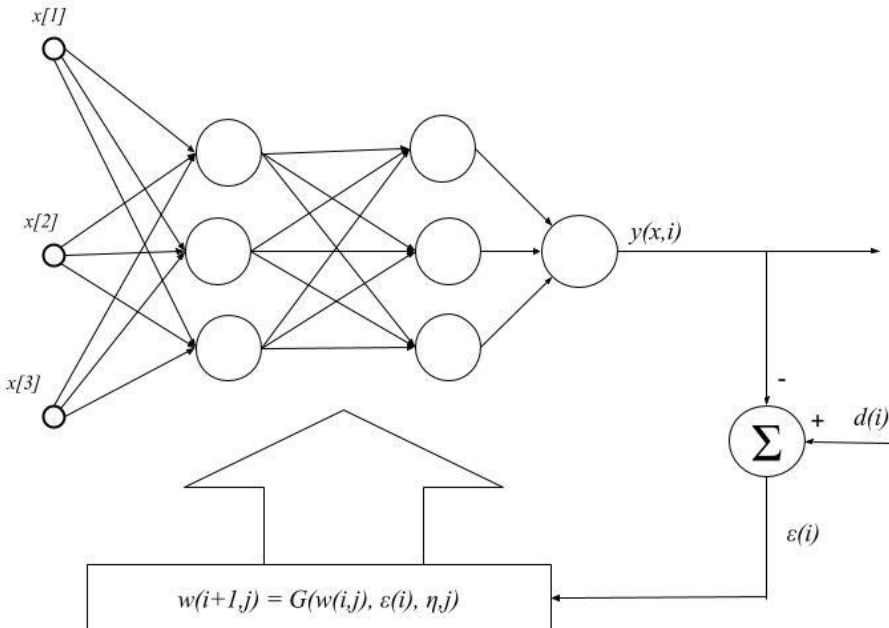


Рис. 2. Двошарова нейронна мережа з прихованим шаром

Таким чином, авторам здається можливим побудувати *адаптивний фільтр* на підставі *навченої* нейронної мережі із **метою** її використання як *автокореляційної* функції.

Важливою частиною досліджень рядів динаміки є очищення сигналу від шуму. Традиційно при аналізі котирувань використовуються лінійні фільтри, що були описані вище – ковзні середні. Суттєвим недоліком цього класу фільтрів є сталість їх частотних характеристик у часі. Натомість ринок є таким, що його частотні характеристики динамічно випадково змінюються у часі під впливом безлічі факторів.

Т. Мейнл порівняв різні методи аналізу рядів динаміки у своїй праці [3] та запропонував для вирішення завдання очищення сигналу від шуму використання фільтрів на основі вейвлетів. Такими фільтрами можуть стати фільтри вейвлетного стиснення, що описано у праці [4].

Суть метода використання фільтрів вейвлетного стиснення полягає у такому: нехай ми маємо деякий сигнал $y = (y_1, y_2, \dots, y_n)$, де $N = 2^j - 1$. Оберемо деякий вейвлетний фільтр, визначимо його параметри та побудуємо ортогональну матрицю $W_{n \times n}$. Наступним кроком отримуємо матрицю коефіцієнтів вейвлета $w = Wy$, аналізуємо величини коефіцієнтів за одним з правил, що мінімізують ризик помилки, та знаходимо поріг η . Прикладом такого правила може бути правило «мінімакс». При використанні цього правила величина порогу обирається така, що мінімізує теоретичну границю асимптотичного ризику.

Далі ті коефіцієнти w , що не перевищують поріг, вважаються незначними і прирівнюються до 0. Така фільтрація називається «жорсткою». Можлива також «м'яка» фільтрація за правилом:

$$w^* = \text{sign}(w) \cdot f(|w| - \eta), \text{ де } f(x) = \begin{cases} x, & x \geq 0 \\ 0 & \end{cases}. \quad (7)$$

Таким чином, ми одержуємо вектор w_{cm} і з ним виконуємо зворотне вейвлетне перетворення $y_{ycm} = W^T \cdot w_{cm}$, де w_{cm} і є згладжений сигнал.

Реалізація алгоритмів вейвлетної фільтрації та побудови моделей на основі нейронних мереж зазвичай є трудомісткими. Одним з ефективних та розповсюджених у міжнародному масштабі є середовище для вирішення статистичних і наукових задач *R-language 3.0.2. R-language*, що одночасно являє собою мову програмування високого рівня та середовище для виконання програм, написаних цією мовою. *R-language* пропонує спеціалізовані інструменти для аналізу даних, побудови і перевірки моделей, в тому числі й з використанням штучного інтелекту.

Для вирішення задачі вейвлетної фільтрації авторами пропонується використовувати функцію *wavShrink()* пакета *wmtsa* [5]. Функція дозволяє виконувати вейвлетну фільтрацію широким набором вейвлетних фільтрів: вейвлети Добеші, симлети, коефлети та інші; використовувати різні функції визначення порога фільтрації та методів фільтрації. Для навчання нейронних мереж використовується пакет *neuralnet* [6], що дозволяє навчати багат шарові нейронні мережі різними методами зворотного розповсюдження помилки, а також використовувати отримані моделі для прогнозування.

Результати досліджень. Для вирішення задачі прогнозування створено програму, яка виконується на платформі *R-language*. Розроблена авторами програма реалізує такий *алгоритм*: зчитує вхідні дані з csv-файлу у вигляді матриці; згладжує дані; ділить дані на дві підмножини – навчальну і контрольну; навчає нейрону мережу на навчальній підмножині; перевіряє навчену мережу на контрольній підмножині.

Завантажені дані мають вигляд матриці $5 \cdot N$, де N – кількість точок даних, а кожний стовпець відповідає одній із величин, які характеризують стан ринку в певний проміжок часу – ціну на початок періоду, ціну на кінець періоду, максимальну та мінімальну ціну за період, а також обсяг торгів.

Для подальшого дослідження обираємо один із стовпців. Із цього стовпця необхідно виділити вектори $P_{навч}$ і $P_{контр}$, які будуть репрезентувати навчальну та контрольну множини. Нехай розмір навчальної множини $M < N$, а кількість членів функції авторегресії L . Тоді $P_{навч} = (P_1, P_2, \dots, P_M)$, а $P_{контр} = P$. Надалі навчальний і контрольний вектори згладжуються за допомогою функції *wavShrink()*, пакету *wmtsa*. Як параметри функції використовуються чотирикомпонентні вейвлети Добеші з використанням мінімаксного методу визначення порога та «важкого» алгоритму фільтрації.

Для подання на вхід нейронної моделі згладжені вектори перетворюються на матриці $X_{навч}$ і $X_{контр}$, кожний рядок яких репрезентує L аргументів автокореляційної функції у певний проміжок часу:

$$X_{i,j} = P_{i+j-1}, \text{ де } j = 1..L. \quad (8)$$

З використанням функції *neuralnet()* пакета *neuralnet* виконується навчання моделі нейронної мережі із заданою кількістю шарів та кількістю входів, яка дорівнює обраній глибині авторегресії. Як вхідний сигнал використовується матриця $X_{навч}$, як вихідний – $(P_{L+1}, P_{L+2}, \dots, P_M)$. Як алгоритм навчання – алгоритм еластичного розповсюдження помилки. Крок навчання обирається $\eta = 1 \cdot 10^{-5}$.

Наступним кроком є прогнозування за допомогою отриманої моделі з використанням функції *compute()* пакета *neuralnet*. На вхід функції подається матриця $X_{контр}$ а на виході ми отримуємо вектор $P_{прог}$, який репрезентує прогнозовані значення для періодів $t = (L + 1, L + 2, \dots, N)$.

Якість отриманої моделі перевіряється за допомогою коефіцієнта Стьюдента для рівня значущості $p=0,05$.

У ході експерименту на вхід створеної моделі подавалися дані про курс акцій, які містять 331 точку даних (рис. 3–5). Для навчання моделі використовувались перші 128 точок даних. Симуляція проводилась для різних значень глибини авторегресії L . Якість моделі перевірялася коефіцієнтом кореляції Пірсона R .

Висновки. Запропоновано авторегресійну модель прогнозування цін на фондовому ринку, розроблену із використанням інструментів та алгоритму вейвлетних фільтрів та нейронних мереж. У ході виконання експерименту визначено, що найкращі результати можуть бути отримані у випадку, коли кількість точок даних у минулому, що ми використовуємо у моделі, не перевищує 5. Доведено, що середовище *R-language 3.0.2* та пакети *wmtsa* та *neuralnet* є якісними та зручними інструментами для вирішення задач про-

гнозування рядів динаміки і можуть бути використані для побудови більш складних імітаційних моделей аналізу економічних показників.

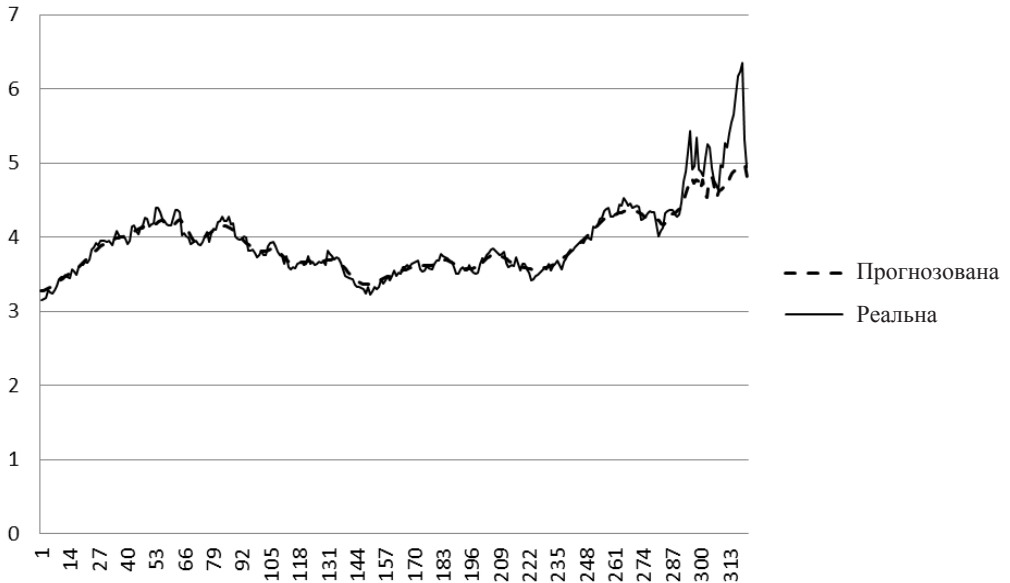


Рис. 3. Реальна та прогнозована ціна $L = 10, r = 0,960$

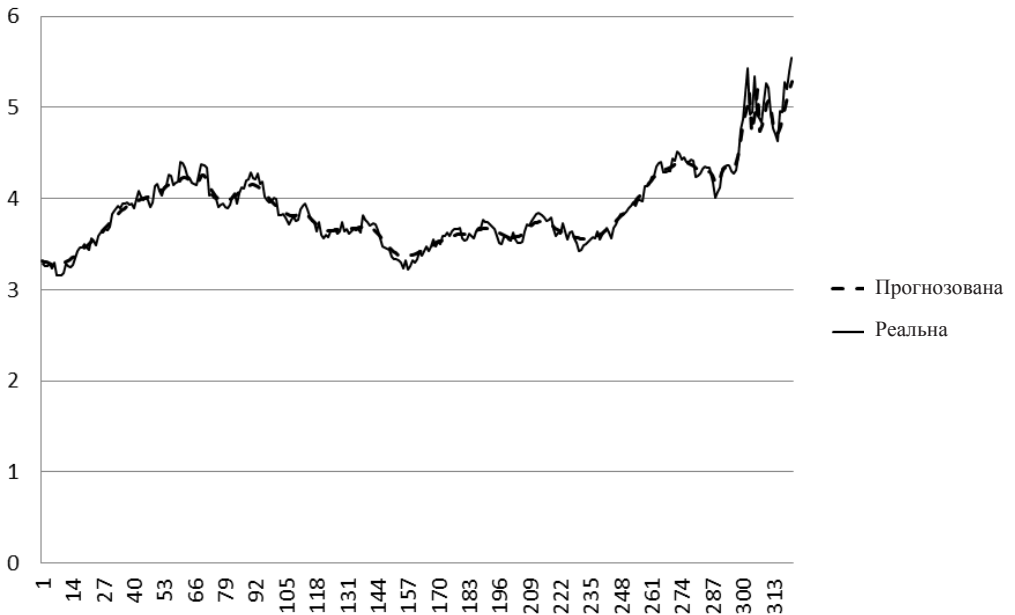


Рис. 4. Реальна та прогнозована ціна $L = 10, r = 0,977$



Рис. 5. Реальна та прогнозована ціна $L = 3$, $r=0,984$

Список використаних джерел

1. Winer N. Extrapolation, interpolation and smoothing of stationary time series / N. Winer. – N.Y., 1949.
2. Riedmiller M. A direct adaptive method for faster backpropagation learning: The RPROP algorithm / M. Riedmiller and H. Braun // Proceedings of the IEEE International Conference on Neural Networks (ICNN) San Francisco, 1993. – P. 586–591.
3. Thomas Meinl. A novel wavelet based approach for time series data analysis: dissertation / Meinl Thomas. – Karlsruhe, 2011.
4. Donoho, D. and Johnstone, I. Ideal Spatial Adaptation by Wavelet Shrinkage. Technical report, Department of Statistics, Stanford University, 1992.
5. Software to book Wavelet Methods for Time Series Analysis, Donald B. Percival and Andrew T. Walden, Cambridge University Press, 2000.
6. S. Fritsch and F. Günther. Neuralnet: Training of Neural Networks. R Journal Vol. 2/1, June 2010. – P. 30–38.

Рассмотрена актуальная научная и практическая проблема разработки модели прогнозирования цен на фондовом рынке с использованием нейронных сетей и вейвлетной фильтрации. Предложена программная реализация данной модели с использованием языка обработки статистических данных международного стандарта R-language 3.0.2. Получены и проанализированы результаты прогнозирования на основе разработанной модели, а также исследованы уровень значимости и точности модели.

Ключевые слова: анализ динамики цен, нейронные сети, вейвлеты, R-language.

The urgent scientific and practical problem of developing the stock prices predictive models using neural networks and wavelet filtering was explored. The software for of this model implementation using the language for statistical data processing of the international standard – R-language 3.0.2 – was suggested. The results on the bases of the developed prediction model was obtained and analyzed, and the level of significance and accuracy of the model was investigated.

Key words: *analysis of price dynamics, neural networks, wavelets, R-language.*

Одержано 28.01.2014.