УДК 004.652, 004.896

Р.Ю. Грінчак
В.В. Кабачій

ВИКОРИСТАННЯ КЛАСТЕРНОГО АНАЛІЗУ ДЛЯ ІДЕНТИФІКАЦІЇ ЗАЛЕЖНОСТЕЙ НА ЧАСОВИХ РЯДАХ

Вінницький національний технічний університет

Анотація

У даному досліджені було розглянуто застосування Fuzzy C-means метода кластеризації до часових рядів фінансових інструментів. Описана розроблена системи прийняття рішень з використанням результатів кластерізації, яка показала гарні результати у визначенні моментів зміни тенденцій, що дають змогу здійснювати операції купівлі або продажу фінансових інструментів. Представленні отримані результати моделювання.

Ключові слова: *часові ряди, кластерний аналіз, трейдинг, система прийняття рішень.*

Abstract

In this study, the application of the Fuzzy C-means clustering method to the time series of financial instruments was considered. The developed decision-making system using the results of clustering is described, which has shown good results in determining the moments of change in trends that allow to carry out operations of buying or selling financial instruments. The modeling results are presented.

Keywords: *time series, cluster analysis, trading, decision-making system.*

ВСТУП

Оскільки головною метою трейдингу завжди є отримання стабільного прибутку та його підвищення в довгостроковій перспективі, необхідно враховувати те, що успіх в торгівлі фінансовими інструментами залежить від того, наскільки краще та швидше трейдер розумітиме зміни на ринку, тобто в тому числі зміни тенденцій на фінансових часових рядах. Володіння ефективними інструментами аналізу надає велику перевагу, оскільки той, хто володіє найкращими засобами аналізу, отримує конкурентну перевагу і, отже, може розраховувати на більш передбачуваний прибуток.

Для досягнення найкращих результатів потрібно комбінувати існуючі методи аналізу ринків, що призводить до необхідності опрацювання обширного спектру ознак для ухвалення рішення. Популярні методи технічного аналізу, які широко використовуються для вивчення фінансового ринку, не завжди можуть враховувати всі аспекти, або цей процес може бути надто складним. У таких обставинах особливо важливим стає використання кластерного аналізу, який дозволяє ефективно зменшувати розмір великих обсягів соціально-економічної інформації, здійснювати їх конденсацію та зробити їх більш доступними.

Протягом останніх років кластерний аналіз і нейронні мережі стали широко використовуваними для аналізу фінансових показників [1, 2], оскільки вони ефективно опрацьовують великий обсяг різноманітних показників і ознак одночасно. Після огляду літературних джерел та наукових публікацій, виявлено, що обмежена увага приділяється використанню кластерного аналізу та нейронних мереж для ідентифікації конкретних закономірностей на фінансовому часовому ряді, у той час як їх головне використання в цій сфері зосереджується на прогнозуванні. Таким чином, є важливим проведення подальших досліджень в цьому контексті.

Метою дослідження є підвищення ефективності передбачення динаміки фінансового часового ряду шляхом ідентифікації залежностей його поведінки. Цей аналіз здійснюється на основі використання кластерного аналізу та розробленої системи прийняття рішень.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для успішного використання векторів-ознак у статистичних дослідженнях важливо провести ретельну обробку та аналіз початкових даних. Цей етап включає в себе виявлення потенційних грубих помилок у дослідженні, а також виявлення помилок, що можуть виникнути при кодуванні і трансформації даних. Додатково, важливо виявити можливі шуми або аномалії у спостереженнях.

Одним із ключових етапів в цьому процесі є формування векторів-ознак на основі статистичних методів. Це включає в себе визначення релевантних характеристик даних, які найкраще відображають

їхню структуру та властивості [3]. Також, важливо враховувати взаємозв'язки між ознаками та відбирати ті, які найбільше впливають на цільову змінну. Це може включати аналіз кореляцій, використання методів відбору ознак, таких як дерева рішень чи алгоритми важливості ознак, для визначення вагомості кожної ознаки в моделі.

Першим кроком був збір достовірних валютних даних з відкритих та достовірних джерел. В цьому випадку історичні дані валютної пари було отримано з веб-сайту «investing.com», який, зокрема, надає безкоштовні котирування активів фінансових ринів. Завантажені дані валютної пари NZDUSD було конвертовано в формат CSV для подальшої обробки.

Оскільки отримані дані з джерела мали в собі деякі нечислові дані які не мають ніякої цінності в цьому досліджені, їх було видалено. Окрім того, для зручності подальшого маніпулювання даними деякі стовпці було перейменовано.

Для обробки даних було обрано Jupyter Notebook – інтерактивне середовище для програмування, яке дозволяє об'єднувати код, текстові описи та графіки в одному документі. Інтерактивність Jupyter Notebook полягає в тому, що ви можете виконувати окремі частини коду та спостерігати за результатами в реальному часі. Результати виводяться безпосередньо під коміркою коду, що дозволяє легко візуалізувати та розуміти результати обчислень [4]. У Jupyter Notebook можна працювати з різними мовами програмування, але найбільш популярною є Python.

Python підтримує об'єктно-орієнтоване, імперативне та функціональне програмування. Вона має широкий спектр бібліотек і модулів, що полегшують розробку програм та сприяють їх використанню в різноманітних областях, таких як веб-розробка, аналіз даних, штучний інтелект і багато інших. Однією з особливостей Python є його динамічна типізація, що дозволяє змінювати типи даних об'єктів під час виконання програми. Це полегшує розробку та зменшує кількість необхідного коду [5].

На рис. 1 показано графік часового ряду, який побудовано за завантаженими даними, і його згладжування простою середньою для наочного представлення домінуючої тенденції.

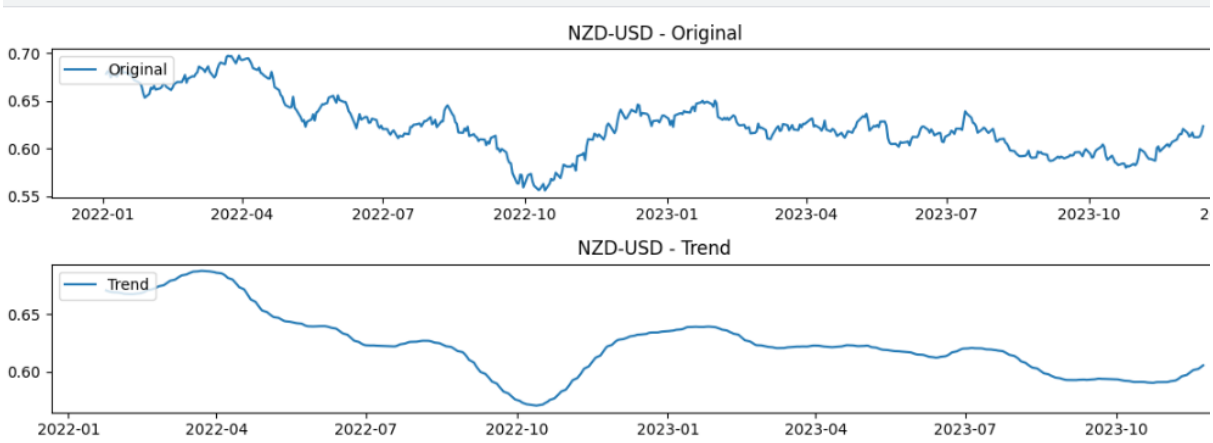

Рисунок 1 – Оригінальний графік часового ряду та графік його тренду.

Цей графік показує курс новозеландського долара до долара США (NZDUSD) протягом останніх двох років. З початку 2022 року курс NZDUSD коливався в діапазоні від 0,55 до 0,70.

Далі було створено коефіцієнт ефективності (K), який базується на розрахованих показниках осциляторів Stochstic та MACD. Вони додають додаткові шари аналізу: Stochstic вказує на ступінь перекупленості чи перепроданості ринку, визначаючи його відносну позицію відносно максимального та мінімального значень цін; MACD використовується для виявлення змін у ринковому імпульсі та генерації сигналів. Розраховані значення двох видів ковзних середніх (MA) використовуються для створення індикаторів короткострокового та довгострокового тренду. Сигнали покупки або продажу генеруються на основі перетину цих середніх [6]. Також, на завершальному етапі визначався найсильніший сигнал серед розглянутих, надаючи ключову інформацію для прийняття рішення щодо входу чи виходу з ринку, формуючи модифікований вектор ефективності K_mod, який доповнив датасет data (рис. 2), який буде використовуватися в результуючій системі прийняття рішень.

Для кластеризації був використаний метод Fuzzy C-means (FCM). Він використовує техніку "розмиття" (fuzzification), щоб дозволити об'єктам даних приналежати не тільки одному, а й усім кластерам з певною ймовірністю чи ступенем належності. У порівнянні з традиційними методами, де об'єкт повністю належить одному кластеру, FCM надає більш гнучкий підхід до моделювання невизначеності в даних. Основна ідея FCM полягає в тому, щоб кожен об'єкт належав кожному кластеру, але з різними ступенями належності. Алгоритм мінімізує функціонал, який враховує відстані

між об'єктами і центрами кластерів, враховуючи ймовірність належності [7]. FCM добре підходить для задач, де об'єкти можуть мати неоднозначні чи перехрещені принципи належності до кластерів.

	Date	Price	MA_Crossover	Stochastic_Oscillator	K_mod
504	2022-01-07	0.6780	1	61.722488	19.009947
505	2022-01-06	0.6745	1	33.908046	18.816946
506	2022-01-05	0.6792	1	58.024691	10.472511
507	2022-01-04	0.6812	1	65.714286	17.707492
508	2022-01-03	0.6784	1	33.913043	20.014379

	S%K	S%D	kst	macd1	macd2	kmacd	K
504	67.129564	78.726618	-4.0	0.002889	0.001891	0.0	15.009947
505	52.665375	66.015894	-1.0	0.002498	0.002013	0.0	17.816946
506	51.218408	57.004449	-1.0	0.002539	0.002118	0.0	9.472511
507	52.549008	52.144264	-1.0	0.002701	0.002234	0.0	16.707492
508	52.550674	52.106030	0.0	0.002574	0.002302	0.0	20.014379

Рисунок 2 – Фрагмент створеного датасету data.

Алгоритм починається з ініціалізації параметрів, включаючи кількість кластерів та початкові центри. На кожній ітерації алгоритму визначаються ступені належності для кожного об'єкта до кожного кластера, з урахуванням відстаней між об'єктом і центрами кластерів. Потім центри кластерів оновлюються на основі зважених значень об'єктів, де ваги – це розраховані ступені належності [8].

Процес ітерацій повторюється до досягнення збіжності, тобто до тих пір, поки ступені належності та центри кластерів не залишаться стабільними (рис. 3). Оптимальні значення ступенів належності та центрів кластерів визначаються так, щоб мінімізувати цільову функцію, що враховує відстані між об'єктами та центрами кластерів. Однією з переваг Fuzzy C-means є його здатність враховувати вагомість об'єктів при визначенні центрів кластерів. Ступені належності служать ваговими коефіцієнтами для об'єктів під час оновлення центрів. Це може бути корисним, коли деякі об'єкти мають більший вплив або важливість для кластеризації.

```python
def create_observation_matrix(data):
    n = len(data)
    X = np.zeros((n, 3))

    max_K_mod = np.max(data['K_mod'])  # Знаходимо максимальне значення K_mod

    for i in range(n - 1):
        X[i, 0] = data['K_mod'].iloc[i]
        X[i, 1] = data['K_mod'].iloc[i]
        X[i, 2] = data['K_mod'].iloc[i + 1] - max_K_mod

    X[-1, 0] = data['K_mod'].iloc[-1]
    X[-1, 1] = data['K_mod'].iloc[-1]
    X[-1, 2] = data['K_mod'].iloc[-1]

    return X

# Виклик функції для створення матриці спостережень X
observation_matrix = create_observation_matrix(data)

observation_matrix[np.isnan(observation_matrix)] = 0

def perform_clustering(observation_matrix, c=2):
    cntr, u, _, _, _, _, _ = fuzz.cluster.cmeans(
        observation_matrix.T, c, 2, error=0.005, maxiter=1000)
    return cntr, u

# Кластеризація для покупок
c = 2
cntr_buy, u_buy = perform_clustering(observation_matrix, c)
F_buy = u_buy.T
F_sell = -F_buy  # Перетворення матриці -F
```

Рисунок 3 – Код кластеризації даних.

Як видно на рис. 3, спершу, створюється матриця спостережень (observation_matrix), яка включає у себе значення стовпця 'K_mod' з датасету. Ця матриця готується для подальшого використання у кластеризаційному алгоритмі. Далі викликається функція perform_clustering, яка використовує нечітку кластеризацію (fuzzy c-means) для розподілу спостережень між кластерами. В результаті отримуємо центри кластерів (cntr_buy) і матрицю приналежності (u_buy) для кластера "покупок". Зазначений код також передбачає можливість використання того ж самого алгоритму для іншого кластера, який вказаний як "продажі" (F_sell). Це робиться шляхом зміни знаків матриці приналежності для кластера "покупок" на протилежні.

Далі формується новий вектор ефективності K_mod з нормованими даними (рис. 4).

```
# Функція для створення вектора ефективності K_mod
def create_efficiency_vector(F_buy, F_sell):
    n = min(F_buy.shape[0], F_sell.shape[0])
    K_mod = np.zeros(n)

    for i in range(n):
        K_mod[i] = np.dot(F_buy[i], np.arange(c_buy)) - np.dot(F_sell[i], np.arange(c_sell))

    # Нормалізація вектора K_mod до проміжку [-1, 1]
    K_mod = 2 * (K_mod - np.min(K_mod)) / (np.max(K_mod) - np.min(K_mod)) - 1

    return K_mod

# Виклик функції для створення вектора ефективності K_mod
K_mod = create_efficiency_vector(F_buy, F_sell)

# Додавання вектора K_mod до датафрейму
data['K_mod'] = K_mod

# Вивід датафрейму з оновленим стовпцем K_mod
print(data[['Date', 'K_mod']])
```

Рисунок 4 – Код формування вектору ефективності K_mod та нормування значень.

В коді визначається функція create_efficiency_vector, яка призначена для створення вектора ефективності K_mod на основі купівельних та продажних кластерів. Функція приймає матриці F_buy і F_sell, що, містять дані про купівельні та продажні фактори відповідно. Для кожного рядка обчислюється значення K_mod за допомогою вагового сумування, де ваги визначаються кількістю купівельних та продажних кластерів. Отриманий вектор K_mod нормалізується до інтервалу [-1, 1], щоб забезпечити стандартизацію результатів. Нарешті, створений вектор ефективності K_mod додається до датафрейму data в новий стовпець з назвою 'K_mod'.

Далі була визначенні умови сигналів покупки Buy_Signal та продажу Sell_Signal активу. Вони формуються на основі трендів та вектора ефективності K_mod (рис. 5). Алгоритм перевіряє умову, що тренд на попередній день був рівний або меншим за 2. Це може вказувати на тенденцію до позитивного руху. Далі, перевіряється поточний тренд, чи є він позитивним. Крім того, розглядається вектор ефективності K_mod, і сигнал покупки генерується, якщо значення K_mod більше 0.5. Об'єднання умов за допомогою операції І (&) гарантує, що всі три умови повинні виконуватись одночасно для генерації сигналу покупки. У випадку сигналу продажу алгоритм аналогічно перевіряє три умови. Перша умова перевіряє, чи тренд на попередньому день був рівний або більшим за 0, що може вказувати на тенденцію до негативного руху. Друга умова перевіряє поточний негативний тренд, і, нарешті, вектор ефективності K_mod повинен бути менше або рівний -0.5 для генерації сигналу продажу.

```
# Додамо нові стовпці для сигналів покупки та продажу
data['Buy_Signal'] = ((data['Trend_Direction'].shift(1) <= 2) & (data['Trend_Direction'] > 0) & (data['K_mod'] >
-0.5)).astype(int)
data['Sell_Signal'] = ((data['Trend_Direction'].shift(1) >= 0) & (data['Trend_Direction'] < 0) & (data['K_mod']
<= 0.5)).astype(int)
```

Рисунок 5 – Код формування сигналів покупки та продажу.

При спрацюванні сигналу покупки імітується відкриття позиції купівлі, а при спрацюванні сигналу продажу її закриття. За допомогою цих дій обчислюється прибуток чи збиток від кожної угоди, а також розраховується результуючий баланс після застосування отриманих моделлю рішень (рис. 6). Для

моделювання був використаний початковий баланс у розмірі 10 000 умовних грошових одиниць та на кожну покупку виділялося 10% від поточного балансу. На основі аналізу таблиці угод можна зробити висновок про успішну роботу системи прийняття рішень. Зафіксовано 16 угод, з яких 11 були успішними, в той час як 5 – невдалими. Загальний прибуток від усіх угод склав 341.55 умовних грошових одиниць. На рис. 7 графічно представлені результати роботи системи прийняття рішень.

Деталі угод з різницею цін:

Покупка (дата)	Ціна покупки	Продаж (дата)	Ціна продажу	Прибуток	Різниця цін
2022-01-04	0.6812	2022-01-12	0.6844	4.697592	0.0032
2022-01-26	0.6651	2022-03-21	0.6884	35.032326	0.0233
2022-05-11	0.6299	2022-05-26	0.6477	28.258454	0.0178
2022-06-10	0.6370	2022-06-15	0.6284	-13.500785	-0.0086
2022-07-04	0.6205	2022-08-12	0.6453	39.967768	0.0248
2022-10-06	0.5656	2022-12-14	0.6454	141.089109	0.0798
2022-12-20	0.6347	2022-12-27	0.6275	-11.343942	-0.0072
2023-01-04	0.6293	2023-01-20	0.6472	28.444303	0.0179
2023-03-06	0.6194	2023-03-31	0.6257	10.171133	0.0063
2023-04-24	0.6166	2023-05-05	0.6293	20.596821	0.0127
2023-05-26	0.6048	2023-06-14	0.6204	25.793651	0.0156
2023-06-27	0.6162	2023-07-14	0.6368	33.430704	0.0206
2023-08-15	0.5949	2023-08-18	0.5921	-4.706673	-0.0028
2023-08-23	0.5979	2023-08-29	0.5971	-1.338016	-0.0008
2023-09-05	0.5884	2023-09-27	0.5921	6.288239	0.0037
2023-09-29	0.5995	2023-10-06	0.5987	-1.334445	-0.0008

Початковий баланс: 10000.0

Успішних угод: 11, Невдалих угод: 5

Баланс після всіх угод: 10341.546239141417

Рисунок 3.20 – Результати імітації здійснення операцій.

Рисунок 3.19 – Графік з ідентифікованими моментами зміни тенденції (зелений кружок – умовне відкриття угоди; червоний – умовне закриття).

ВИСНОВКИ

Проведені експериментальні дослідження ефективності обраного підходу за використанням Fuzzy C-means методу кластеризації на часових рядах. Розроблена системи прийняття рішень з використанням результатів кластерізації, показала гарні результати у визначенні моментів зміни тенденцій, що дають змогу здійснювати операції купівлі або продажу фінансових інструментів. Створені моделі є сенс дослідити для рядів інших фінансових інструментів на різних часових проміжках та масштабах часу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Аналіз часових рядів з використанням кластерного аналізу / А. В. Кушніренко // Вісник НТУУ «КПІ». Серія: Інформатика, управління та обчислювальні системи. – 2022. – № 3. – С. 109-115.Кластеризація часових рядів для прогнозування фінансових ризиків / М. Капулло // Вісник Національного банку України. – 2015. – № 2. – С. 44-48.
2. Аналіз часових рядів з використанням кластерного аналізу / А. В. Кушніренко // Вісник НТУУ «КПІ». Серія: Інформатика, управління та обчислювальні системи. – 2022. – № 3. – С. 109-115.
3. Volosyuk Y. (2020) Analysis of clustering algorithm for Data Mining tasks, pp. 112-119.
4. Schafer, C. (2023). Jupyter Notebooks: The Complete Guide [Udemy video course].
5. NumPy in Python. URL: https://numpy.org/ [Дата звернення: 17.10.2023].
6. Бакай Є. І., Кабачій В. В, Маслій Р. В. Модель прийняття рішень для фінансових часових рядів на основі пари середніх з використанням оцінки різних часових вимірів [Електронний ресурс] – URL: http://ir.lib.vntu.edu.ua//handle/123456789/21777
7. Mantula, E., & Mashtalir, V. (2013, June). An Adaptive Forecasting of Nonlinear Nonstationary Time Series under Short Learning Samples. In ICTERI (pp. 91-98).
8. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.

Грінчак Радислав Юрійович – студент групи 1ІСТ-22м, Факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: rgrinchak@gmail.com;

Кабачій Владислав Володимирович – к. т. н., доцент кафедри Автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, м. Вінниця, e-mail: kabachij.v.v@vntu.edu.ua

Grinchak Radyslav - student of 2IST-22m group, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: rgrinchak@gmail.com;

Kabachy Vladyslav Volodymyrovych – Ph.D., Associate Professor of Automation and Intelligent Information Technologies, Vinnytsia National Technical University, Vinnytsia, e-mail: kabachij.v.v@vntu.edu.ua