Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ КЛАСТЕРИЗАЦІЇ ПОСЛІДОВНОСТЕЙ ЗА НЕПРЯМИМИ ВИМІРАМИ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-20-1
Богуцька Т.В.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Машталір В.П.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Богуцькій Тетяні Володимирівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження кластеризації послідовностей за непрямими вимірами

затверджена наказом по університету від « 22 » жовтня 2021 року № 1574Ст
2. Термін подання студентом роботи до екзаменаційної комісії 26 листопада 2021 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, мова програмування Python, теоретичні відомості про особливості часових рядів, теоретичні відомості про методи кластеризації послідовних даних.

4. Перелік питань, що потрібно опрацювати в роботі
1. Огляд загального процесу кластеризації послідовностей.
2. Огляд існуючих алгоритмів кластеризації.
3. Огляд використання непрямих характеристик часових рядів.
4. Побудова системи кластеризації часових рядів за непрямими вимірами.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність дослідження, об'єкт та мета дослідження, постановка задачі дослідження, вихідні дані дослідження, етапи розроблення, апробація результатів роботи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандарт та норм	Доцент Бєлова Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	22.10.2021	
2	Аналіз завдання, підбір літератури	22.10.21–24.10.21	
3	Аналіз літератури з досліджуваної проблеми	25.10.21–26.10.21	
4	Аналіз технічних засобів	27.10.21–28.10.21	
5	Розробка методу кластеризації	29.10.21–30.10.21	
6	Програмна реалізація	01.11.21–10.11.21	
7	Оформлення пояснювальної записки	11.11.21–16.11.21	
8	Перевірка на плагіат	18.11.2021	
9	Рецензування	19.11.2021	
10	Підготовка презентації та доповіді	25.11.2021	
11	Занесення роботи в електронний архів	04.12.2021	
12	Попередній захист кваліфікаційної роботи	06.12.2021	

Дата видачі завдання 22 жовтня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Машталір В.П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 62 с., 20 рис., 40 джерел.

ЧАСОВИЙ РЯД, ЗАДАЧА КЛАСТЕРИЗАЦІЇ, НЕПРЯМІ ОЗНАКИ, PYTHON, ПОСЛІДОВНІСТЬ ДАНИХ.

Об’єктом дослідження є послідовність даних у вигляді часового ряду.

Метою дослідження є розробка онлайн методу кластеризації, що базуються на використанні непрямих ознак часового ряду, які дозволяють розподілити вхідні дані на класи – групи однотипних екземплярів вибірки для подальшого прогнозування.

Проведено дослідження видів та особливостей послідовних даних, досліджено основні проблеми у кластеризації часових рядів. У ході роботи було розглянуто загальний процес кластеризації часових рядів. Досліджено основні алгоритми кластеризації, різноманітні методи отримання статистичних ознак та їх вплив на ефективність кластеризації.

У результаті роботи здійснена програмна реалізація системи для онлайн кластеризації часового ряду використовуючи непрямі характеристики даних.

TIME-SERIES DATA, CLUSTERING, INDIRECT FEATURES, PYTHON, DATA STREAMS.

The object of the research is the sequence of data streams in the form of time-series.

The aim of the research is to develop clustering methods based on the use of indirect features of time-series, which allow to divide input data in the groups of similar examples of data to synopsis them in future.

The research of types and particular qualities of data streams was carried out, the main problems of time-series data clustering were investigated. In the process of work common process of clustering was discussed. The algorithms of clustering, main types of features and their impact on clustering results were investigated.

As a result was implemented software implementation of the system to cluster time-series data online using indirect features.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 7
Вступ 8
1 Аналіз предметної області та постановка задачі 10
1.1 Часові ряди 12
1.1.1 Компоненти часового ряду 16
1.1.2 Недоліки часових рядів в задачі кластеризації 18
1.2 Кластеризація часових рядів 18
1.3 Класифікація методів кластеризації часових рядів 20
1.4 Постановка задачі дослідження 23
2 Аналіз алгоритмів попередньої обробки часових рядів та алгоритмів on-line кластеризації 24
2.1 Перевірка статистичних гіпотез властивостей часового ряду у процесі попереднього аналізу вибірки 24
2.1.1 Перевірка гіпотези о наявності аномальних спостережень .. 24
2.1.2 Перевірка гіпотези о наявності невипадкової величини 25
2.2 Формування векторів-ознак на основі статистичних методів дослідження часових рядів 26
2.3 Методи зменшення розмірності 29
2.4 Міри відстані для послідовних даних 31
2.4.1 Евклідова відстань, середньоквадратична відстань та відстань Міковського 31
2.4.2 Динамічна деформація часу (DTW) 32
2.5 Алгоритми on-line кластеризації 34
2.5.1 Stream on-line алгоритм 34
2.5.2 CluStream on-line алгоритм 36
3 Експериментальні дослідження 37
3.1 Алгоритм обчислення середніх Велфорда 38

3.2 Кластеризація за непрямими вимірами 39
3.2.1 Формування векторів-ознак за непрямими вимірами 39
3.2.2 Експоненційне згладжування 40
3.2.3 On-line кластеризація 41
3.3 Обґрунтування вибору середовища програмної реалізації 42
3.3.1 NumPy 45
3.3.2 Pandas 46
3.3.3 Matlotlib 47
3.4 Опис вхідних даних 48
3.5 Програмна реалізація 52
3.6 Отримані результати та їх аналіз 55
Висновки 57
Перелік джерел посилання 59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

МНК – метод найменших квадратів

PCA – Principal Component Algorithm – метод головних компонент

DTW – Dynamic Time Wrapping – динамічна деформація часу

CSV – формат даних для представлення табличних даних

ВСТУП

Стрімке зростання обробляємих об’ємів інформації та посилення потоків даних в навколишньому світі підштовхує сучасні технології до назрілого завдання підвищення ефективності пошуку необхідної інформації в всесвітньому інформаційному просторі. Це завдання потребує дослідження та розроблення методів і алгоритмів розподілу інформаційних моделей об’єктів на певні групи і класи. Задачі зазначеного роду з’являються в таких сучасних інформаційних технологіях як Data Mining.

Основні етапи аналізу даних: отримання даних, попередня обробка, отримання описових ознак та прогнозування.

В цьому напрямку наукову цікавість представляють методи кластеризації. Важливість використання саме цих методів полягає в тому, що вони дають можливість виділити групи інформаційних об’єктів, близьких за певними ознаками, без будь-якої попередньої інформації про розподіл інформаційних об’єктів на групи [1].

Більшу частину інформації, що пов’язана з обробкою великих обсягів даних, містять часові ряди. Часовий ряд (або динамічний ряд) – це дані, що накопичуються з плином часу швидкими темпами [2]. Необхідність використання аналізу послідовностей даних можна зустріти у таких різноманітних областях як економіка, фінанси та медицина. Так саме до таких даних можливо віднести транзакційні дії за карткою людини, тобто накопичення інформації про використання кредитної карти у різних цілях. Це може також бути корисним, наприклад для подальшого аналізу або прогнозування витрат людини. За допомогою застосування даного методу невпинно розвивається аналіз активності користувачів вебсторінок.

Часові ряди, як набір послідовних спостережень, наряду з великим обсягом ще мають певні особливості. До саме таких особливостей можна віднести: дані можуть надходити у дискретному вигляді, тобто можуть мати

велику кількість оригінальних значень; можуть бути зашумлені викидами, мати так званий «прокльон розмірності», що є недоліком для використання більшості методів кластеризації.

Дослідження кластеризації часових рядів описані у багатьох роботах науковців, тому не зважаючи на достатню кількість недоліків даних в такому вигляді, на сьогоднішній час, актуальною є розробка нових та покращення існуючих методів аналізу послідовностей даних.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

Стрімкий розвиток технологій призвів до того, що дотепер достатньо велика кількість інформації переводиться до зберігання в електронному вигляді. Це значно спрощує процес зберігання та обробки інформації, а також отримання потрібної інформації за певним запитом змістовної інформації. Зі збільшенням кількості інформації, що зберігається на електронних носіях, збільшується необхідність використовувати більш швидкі методи та технології для обробки такої інформації. Постійний розвиток таких інформаційних технологій (хмарні або різноманітні технології обробки великих даних (Big Data)) призвели до необхідності дослідження та застосування інтелектуального аналізу даних [3].

Термін «data mining» з'явився у 1990-х роках, але саме обробка даних виникла у 18 столітті, основуючись на теоремі Байеса, чуть пізніше на регресійному аналізі [4]. Інтелектуальний аналіз даних, що застосовують для обробки великих обсягів інформації, дає змогу зрозуміти різноманіття процесів для її подальшого використання у різноманітних сферах життєдіяльності, що супроводжують людину. Наприклад, у бізнесі, науці, техніці, медицині, телекомунікаціях. Аналіз таких даних як транзакційні дії за кредитними картками, дослідження акцій компаній – лише невелика частка можливих варіантів використання «data mining».

Фундаментально інтелектуальний аналіз даних грунтується на трьох поняттях:

– статистика – є основою більшості таких технологій, як кластерний аналіз, регресійний аналіз та інші;

– штучний інтелект – відтворення нейронної мережі мислення людини в цифровому вигляді;

– машинне навчання – сукупність штучного інтелекту та статистики.

У «data mining» найчастіше використовуються наступні класи задач:

– виявлення аномалій – знаходження даних, які відрізняються від загальної маси;

– кластеризація – групування наборів даних без заздалегідь відомих шаблонів;

– класифікація – узагальнення відомого шаблону для застосування до нових даних.

Значну частину інформації, пов’язану з обробкою великих обсягів даних, містять часові ряди. Часовий ряд (або динамічний ряд) – це дані, що накопичуються з плином часу швидкими темпами.

Необхідність аналізу послідовностей даних можна зустріти у таких областях як економіка, фінанси та медицина. Невпинно розвивається аналіз активності користувачів вебсторінок. Це може бути також корисним, наприклад для подальшого аналізу або прогнозування вподобань людини. Тобто розширюється сфера використання часових рядів та дослідження аналізу часових рядів стає досить актуальним питання на сьогоднішній час.

Однією з перспективних задач аналізу часових даних можна виділити кластеризацію. Задача кластеризації полягає у розбитті вхідної сукупності аналізованих об’єктів на окремі групи (кластери) таким чином, щоб відмінності між об’єктами всередині груп були мінімальними, а відмінності між групами максимальними. Нижче приведені лише декілька практичних задач потребуючих кластерного аналізу:

– знаходження фінансових показників зі схожою динамікою;

– об’єднання пацієнтів в однорідні групи по формі електрокардіограми;

– класифікація типів активності людини за показниками розумних часів.

Метою даного дослідження є аналіз існуючих методів та розробка методу кластеризації часових рядів.

Для досягнення поставленої мети необхідно розглянути такі питання:

– розгляд поняття «часовий ряд» та його особливостей;

– розгляд основних етапів кластеризації послідовних даних;

– розгляд підходів кластеризації часових рядів.

Реалізація даної задачі потребує таких етапів:

Етап 1. Вивчення поняття «часовий ряд», визначення його особливостей.

Етап 2. Розбір та аналіз основних етапів кластерізації послідовних даних.

Етап 3. Аналіз різноманітних підходів кластеризації часових рядів.

1.1 Часові ряди

Часовий ряд – це множина спостережень, що генеруються послідовно у певний період часу. Якщо час неперервний, часовий ряд називається процесом, а якщо час змінюється дискретно – дискретним часовим рядом. Спостереження дискретного часового ряду, зроблені в моменти часу t_1, t_2, ... t_N, можуть бути позначені як $x(t_1)$, $x(t_2)$, ... $x(t_N)$. На рисунку 1.1 зображено приклад неперервного часового ряду.

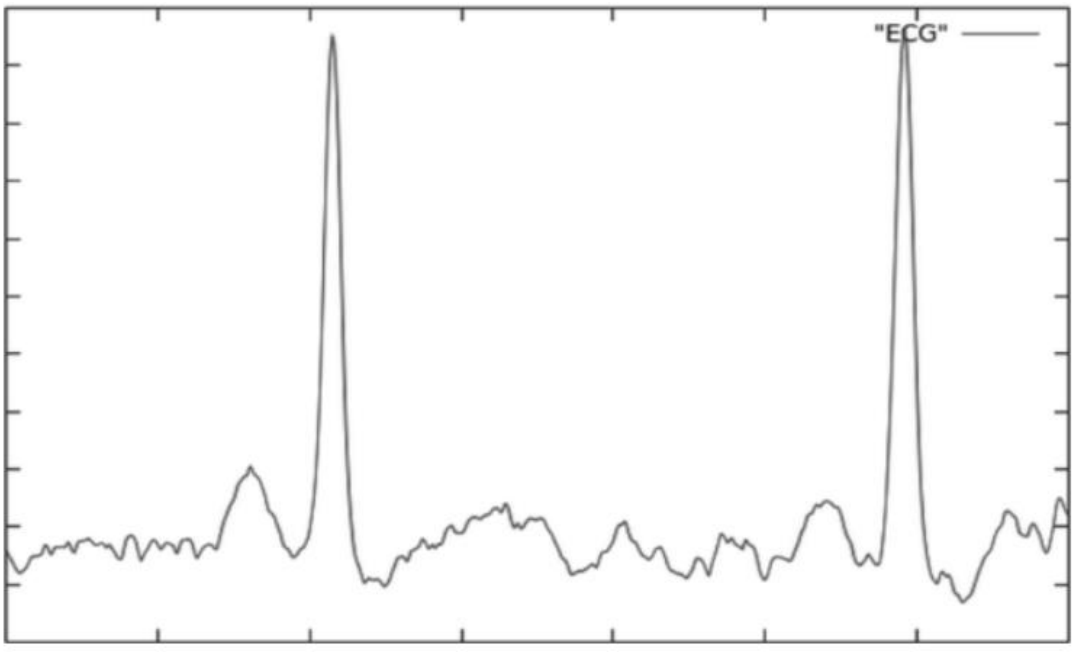

Рисунок. 1.1 – Приклад неперервного часового ряду у вигляді електрокардіограми пацієнта

На рисунку 1.2 зображено приклад дискретного часового ряду у вигляді зміни курсу валюти протягом 100 днів, де спостереження були отримані кожен день.

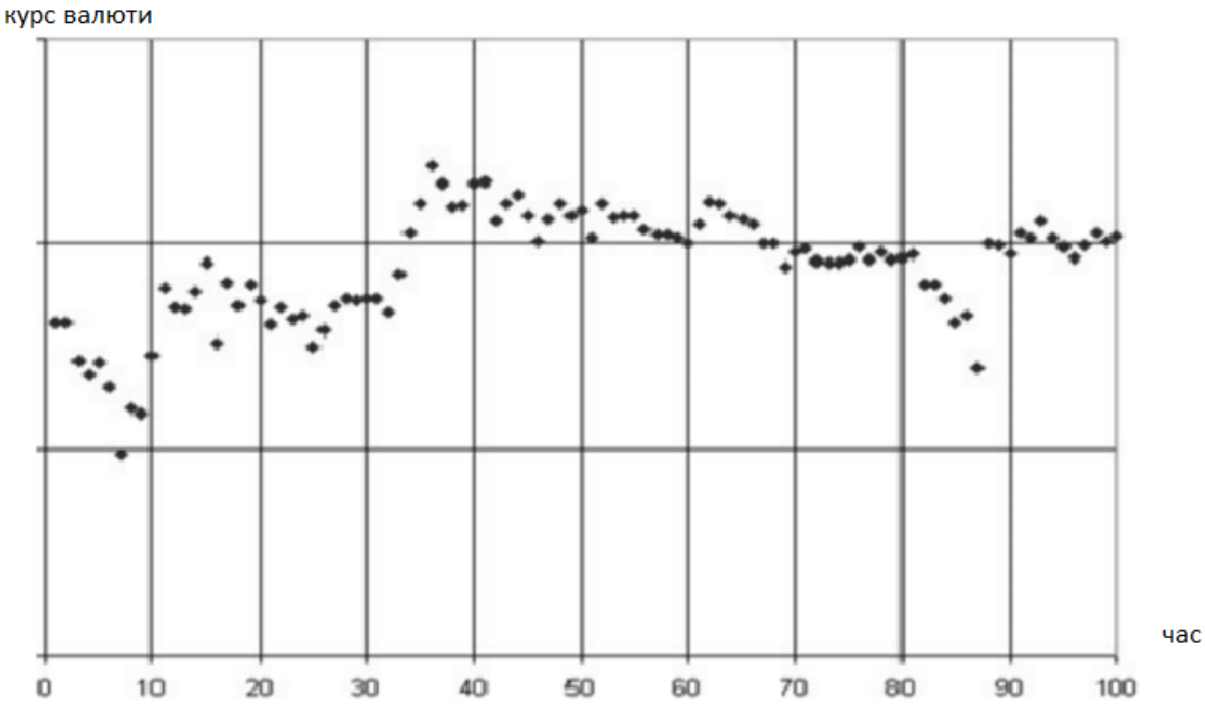

Рисунок 1.2 – Приклад дискретного часового ряду у вигляді зміни курсу валюти

Часовий ряд, який підлягає аналізу, може бути розглянутий як одна часткова реалізація системи, що навчається та генерується прихованим ймовірнісним або іншим механізмом [5].

Розрізняють інтервальні, моментні та довільні часові ряди:

– інтервальним рядом є такий ряд, кожен рівень якого характеризує явище за певний період часу, де рівні характеризують значення показника у ряді за певні відрізки часу. Особливістю інтервальних часових рядів є можливість додавання їх рівнів;

– моментним рядом є такий ряд, кожен рівень якого характеризує виникненням на певний момент часу. На відміну від інтервального ряду рівні моментного ряду не піддаються перетворенням, тобто їх не можна дробити або

підсумовувати для утворення довільних рядів. У моментних часових рядах рівні характеризують значення показника станом на певні моменти часу. Окремі ж рівні даного ряду абсолютних величин містять елементи повторних відліків;

– до часових (довільних) рядів належать ряди середніх показників і ряди відносних величин. Такі ряди утворюються розрахунковим шляхом на основі інтервальних і моментних рядів.

На рисунку 1.3 зображено інтервальний часовий ряд кількості затверджених докторських дисертацій (за даними, опублікованими у бюлетені ВАК України у 2002–2005 рр.) [6].

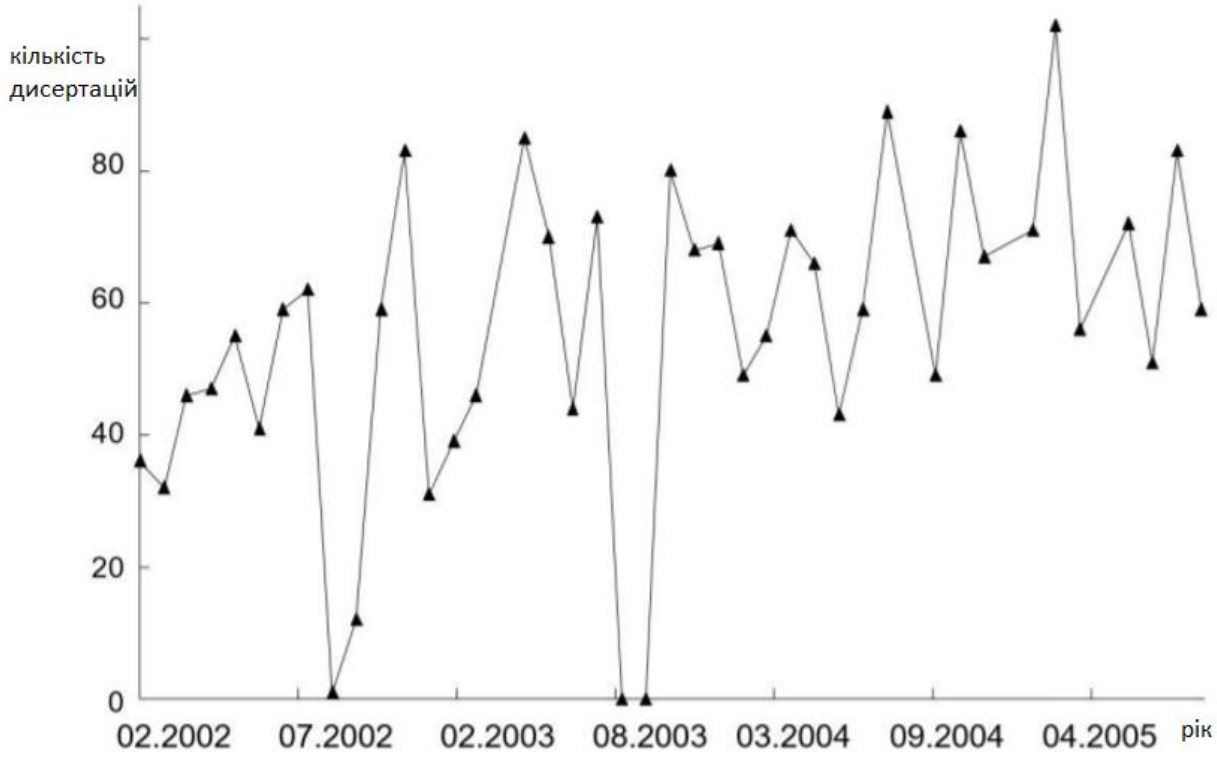

Рисунок 1.3 – Приклад інтервального часового ряду

На рисунку 1.4 зображено приклад моментного часового ряду кроскурсу «гривня – долар США» у 1996−2005 рр. за даними Національного банку України.

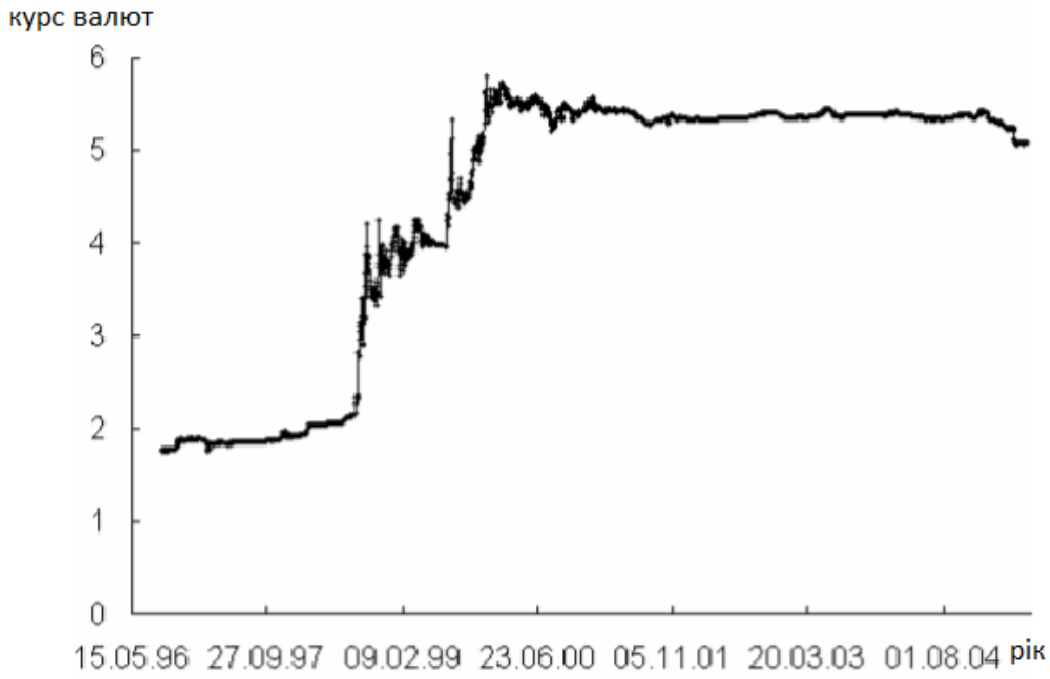

Рисунок 1.4 – Приклад моментного часового ряду

Часові ряди класифікуються:

– за формою подання рівнів:

1) ряди абсолютних показників;
2) відносних показників;
3) середніх показників;

– за кількістю показників, для яких визначаються рівні в кожен момент часу:

1) одновимірні часові ряди;
2) багатовимірні часові ряди;

– за характером часового параметра:

1) моментні часові ряди;
2) інтервальні часові ряди.

За відстанню між датами та інтервалами часу виділяють рівновіддалені– коли дати реєстрації або закінчення періодів слідують один за одним з рівними інтервалами та неповні (нерівномірні) – коли принцип рівних інтервалів не дотримується [7].

За наявністю пропущених значень виділяють повні і неповні часові ряди. Часові ряди бувають детермінованими і випадковими. Перші отримують на основі значень деякої невипадкової функції (ряд послідовних даних про кількість днів у місяцях), другі є результатом реалізації деякої випадкової величини.

Залежно від наявності основної тенденції виділяють стаціонарні ряди, в яких середнє значення і дисперсія постійні, і нестаціонарні, що містять основну тенденцію розвитку.

1.1.1 Компоненти часового ряду

Основними компонентами, що впливають на часовий ряд можна назвати: тренд, сезонність, циклічність і випадковий шум. Тренд є загальною систематичною лінійною або нелінійною компонентою, яка може змінюватися в часі. Сезонна складова – це компонент, що періодично повторюється. Обидва ці види регулярних компонентів часто присутні в ряді одночасно.

На рисунку 1.5 зображено приклад часового ряду з вираженим трендом.

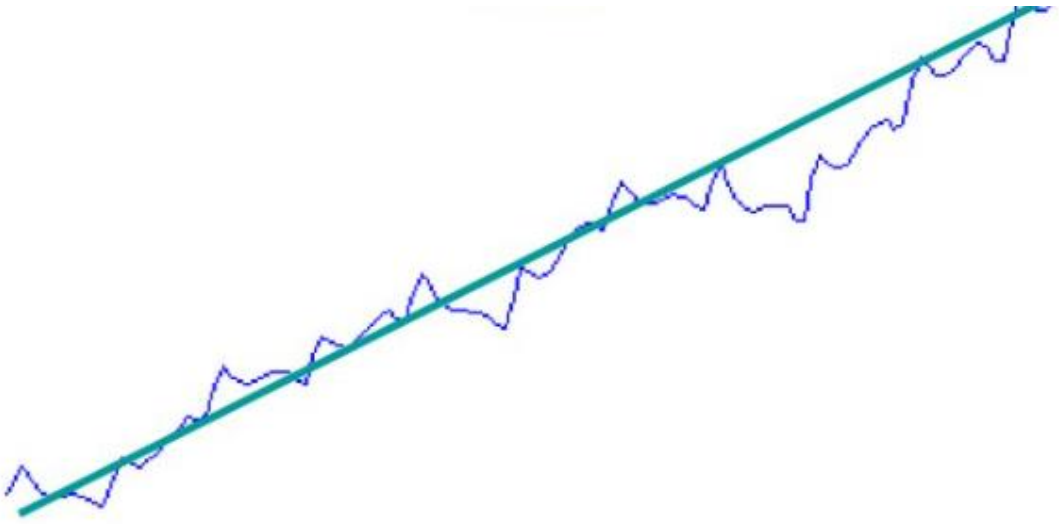

Рисунок 1.5 – Приклад часового ряду з вираженим трендом

На рисунку 1.6 зображено приклад часового ряду з вираженою сезонною компонентною.

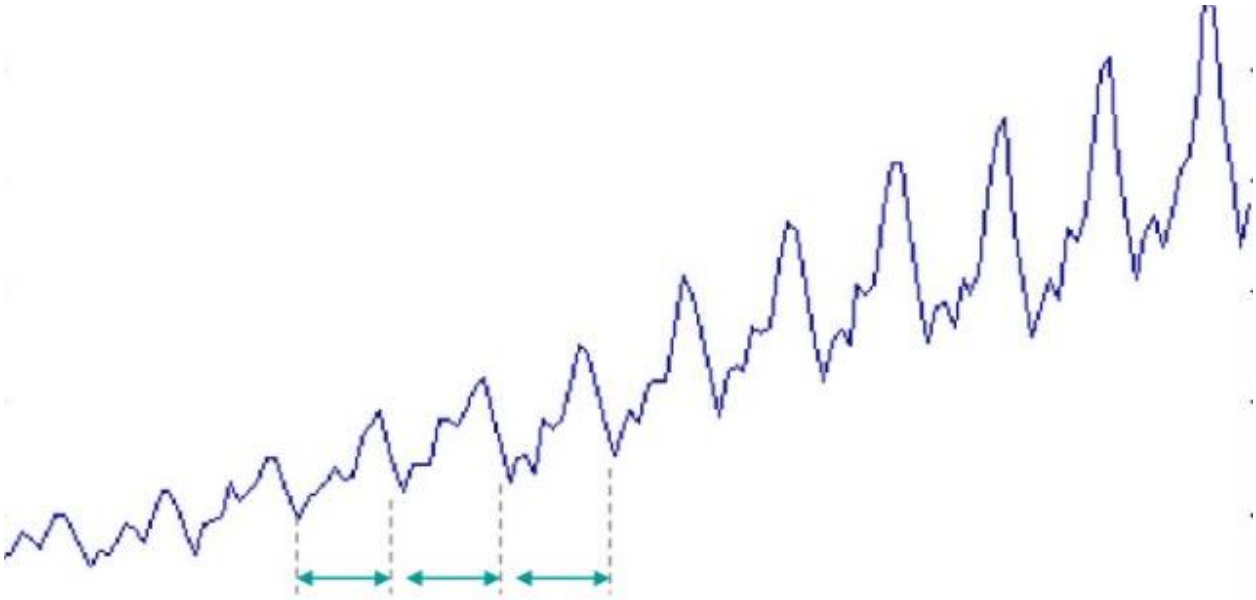

Рисунок 1.6 – Приклад часового ряда з вираженою сезонною компонентою

На основі вище зазначених компонент часового ряду використовують мультиплікативні або адитивні моделі.

Мультиплікативна модель:

$$Y(T) = T(t) \times S(t) \times S(t) \times C(t) \times I(t). \quad (1.1)$$

Адитивна модель:

$$Y(T) = T(t) + S(t) + S(t) + C(t) + I(t), \quad (1.2)$$

де $Y(t)$ – спостереження;

$T(t)$ – тренд;

$S(t)$ – сезонність;

$C(t)$ – циклічність;

$I(t)$ – випадковий шум в часі.

Зазначені вище основні компоненти часового ряду є прямими ознаками, що отримуються у ході статистичного дослідження даних для подальшої кластеризації.

1.1.2 Недоліки часових рядів в задачі кластеризації

Часові ряди, як набір послідовних спостережень, наряду з великим обсягом мають ще певні особливості. До цього можна віднести: дані можуть надходити у дискретному вигляді, тобто можуть мати велику кількість унікальних значень; можуть бути зашумлені викидами, мати так званий «прокльон розмірності», аномальні викиди, що є недоліком для використання більшості статистичних методів аналізу даних.

Існуючі міри схожості, що використовуються для кластеризації статичних даних, не завжди можуть бути використані для часових рядів, адже дані, що надходять на обробку, найчастіше мають різну довжину ряду. Також багатовимірні часові ряди мають дані, що можуть перетинатись.

Усі вище зазначені особливості – велика розмірність, наявність шуму, висока кореляція – призводять до необхідності аналізу та розробки нових підходів для попередньої обробки даних та подальшого аналізу.

1.2 Кластеризація часових рядів

Кластеризація – це процес об’єднання об’єктів у групи, члени яких подібні до певної міри, а кластер – це безліч об’єктів, близьких між собою у смислі деякої міри схожості. На рисунку 1.7 зображено графічний приклад, що обумовлює необхідність кластеризації часових рядів.

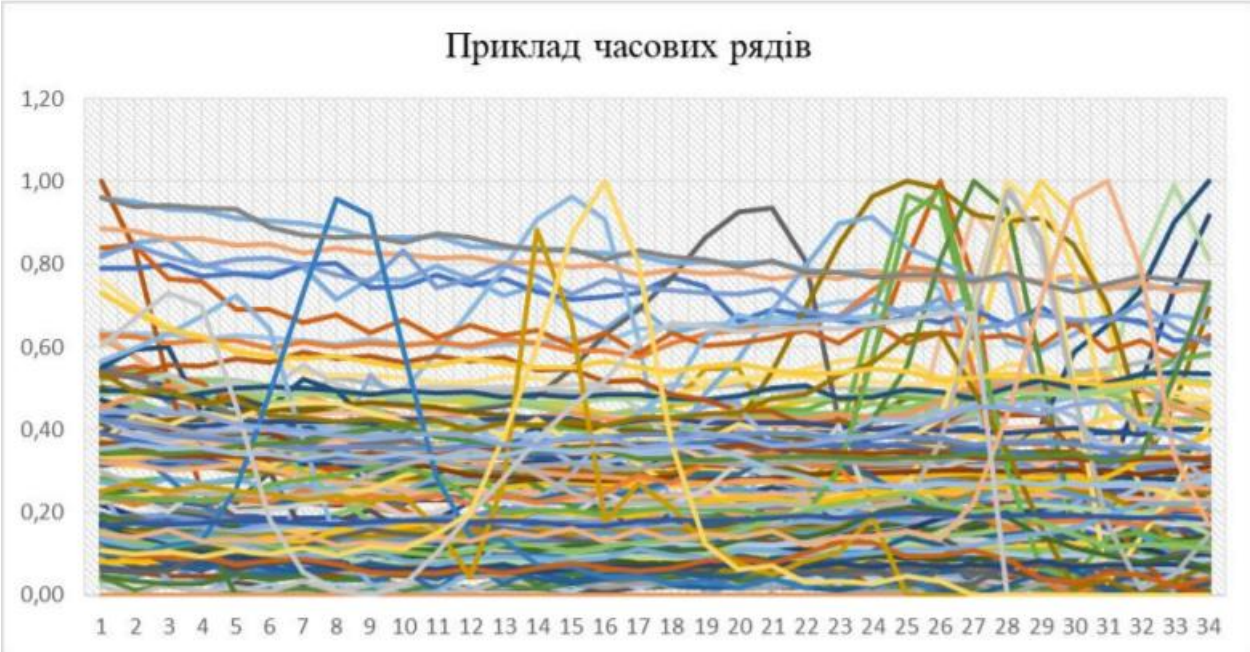

Рисунок 1.7 – Приклад послідовних даних, що подаються на обробку

Завдання кластеризації відноситься до статистичної обробки, а також до широкого класу завдань навчання без вчителя. Головна відмінність кластеризації від класифікації полягає в тому, що перелік кластерів чітко не заданий і визначається в процесі роботи алгоритму. Застосування кластерного аналізу в загальному вигляді зводиться до наступних етапів [8]:

Етап 1. Ідентифікація вибірки об’єктів для кластеризації.

Етап 2. Визначення множини змінних, за якими планується проводити оцінку об’єктів у вибірці. При необхідності нормалізація значень змінних.

Етап 3. Обчислення значень тієї або іншої міри схожості між об’єктами.

Етап 4. Застосування одного з методів кластерного аналізу для створення груп подібних об’єктів (кластерів).

Етап 5. Перевірка вірогідності результатів кластерного розв’язку.

У класичних підходах, що найчастіше використовуються для кластеризації, дані для обробки надходять у повному вигляді, тобто часовий ряд обробляється як вибірка загалом, а не окремими спостереженнями, тобто самі спостереження об’єднані у формі пакету і саме у такому вигляді подаються на обробку. Такий підхід значно збільшує час обробки ряду та витрату великої кількості ресурсів для отримання нових результатів, що є дуже

проблематичним, адже часові ряди в більшій мірі представлені великим обсягом спостережень. Для вирішення цієї проблеми можливим є використання on-line алгоритмів обробки послідовностей даних, які поступово надходять на вхід.

1.3 Класифікація методів кластеризації часових рядів

Стандартний процес кластеризації базується на характеристиках даних та статистичних методах дослідження [9]. Кластеризація часових рядів – це згрупування послідовностей у кластери, де дані мають високий рівень схожості у середині кластера та низький рівень з даними у інших кластерах [10].

Так само, як і статична кластеризація даних, кластеризація часових рядів вимагає алгоритму або процедури кластеризації для формування кластерів. Заданий набір немаркованих об’єктів даних і вибір алгоритму кластеризації залежить від типу доступних даних, конкретної мети та застосування. Дані часових рядів можна розрізняти дискретними чи реальними, однаковими чи нерівномірними вибірками, одновимірними чи багатовимірними, а також чи є ряди даних однакової чи неоднакової довжини [11]. Дані з нерівномірно розподіленими спостереженнями повинні бути перетворені в уніфіковані дані, перш ніж можна буде виконувати операції кластеризації. Цього можна досягти за допомогою широкого спектру методів моделювання.

На сьогоднішній день існує велика кількість алгоритмів кластеризації послідовних даних. Незважаючи на різницю у підходах всі методи направлені на модифікацію існуючих алгоритмів для статичних даних для обробки спостережень у формі часових рядів, або для перетворення послідовних даних у статичні для прямого використання методів кластеризації статичних послідовностей.

Можна виділити базові підходи до кластеризації статичних даних, які безпосередньо або опосередковано використовуються для кластеризації часових рядів:

– методи розподілу;

– ієрархічні методи;

– методи на основі моделей.

Методи розподілу розбивають дані на спочатку задану кількість груп оптимізуючи спочатку заданий критерій, тобто сегментуючі дані, що не перетинаються. Такий підхід в більшості застосовують для одновимірних даних. З прикладів можна навести алгоритм *K*-середніх, що є дуже легким та швидким у реалізації, а також він показує велику ефективність для великої кількості спостережень. Недоліками такого підходу для часових рядів можна виділити низьку ефективність у роботі з багатовимірними послідовностями та зашумленими даними.

Ієрархічні методи кластеризації створюють серію вкладених кластерів, що можна представити графічно (дендограмою або деревом). Гілки дендограми показують схожість між групами кластерів. Ієрархічні алгоритми базуються на виділенні кластерів у підгрупи ітеративно або інтегруванні кластеру в інший. Ієрархічні методи розділяються на агломеративні та дивизійні алгоритми. В агломеративних алгоритмах кожне спостереження стає кластером, де після обробки усіх спостережень формуються інші кластери зі схожими спостереженнями. В дивизійних методах усі спостереження спочатку виступають як один кластер і потім після обробки розділяються по мірі схожості. Ієрархічні методи не тільки групують дані за схожістю, а й мають можливість надавати графічне представлення даних.

Підхід на основі моделі використовує статистичні методи для побудови моделі кластерів. Результатом статичного дослідження є отримання вірогідності приналежності конкретного спостереження до конкретного кластеру. Прикладами статистичних алгоритмів можна назвати метод максимальної правдоподібності, метод моментів, метод найменших квадратів.

Для кластеризації часових рядів можна виділити такі підходи:

– методи на сирих даних;

– методи на ознаках;

– методи на основі моделей.

Спочатку найбільш популярними алгоритмами для кластеризації часових рядів були методи основані на сирих даних. Від кластеризації статистичних даних ці алгоритми відрізнялися вибором метрик подібності більш підходящих для послідовностей. Пізніше стали виділяти алгоритми, що конвертують початкові дані у вектор ознак меншої розмірності або у кількість параметрів моделі. Потім використовується сам алгоритм кластеризації, що і можна назвати методами на ознаках або на основі моделей.

На рисунку 1.8 зображено класифікація методів, які використовуються для кластеризації часових рядів.

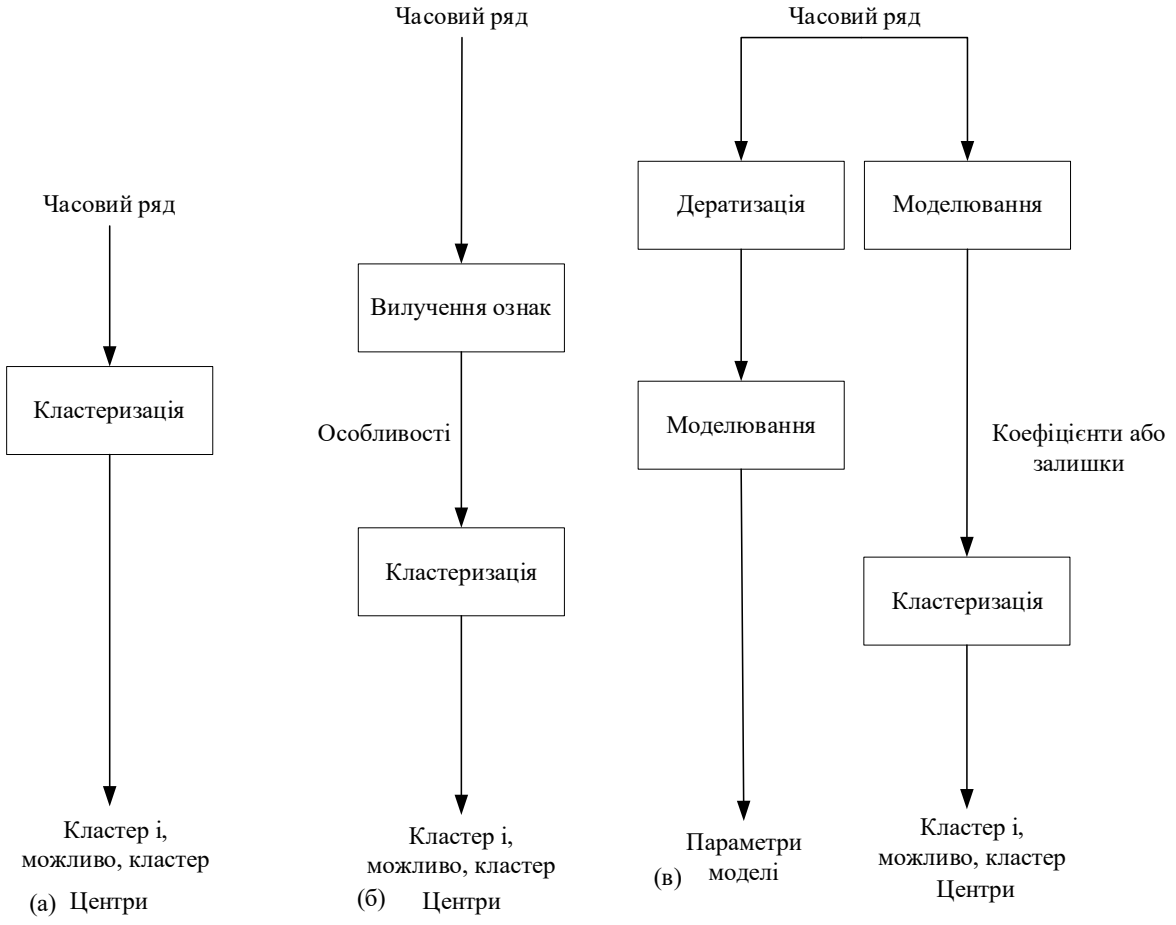

Рисунок 1.8 – Класифікація підходів до кластеризації часових рядів

1.4 Постановка задачі дослідження

Оскільки, як зазначалося раніше, задача кластеризаії часових рядів залишається одною з найактульніших серед задач інтелектуального аналізу даних, тобто зважаючи на описані особливості та проблеми даних, процес аналізу та адаптації існуючих технологій та алгоритмів в сучасному світі потребує уваги.

Об’єктом дослідження є послідовність даних у вигляді часових рядів.

Метою дослідження є розробка онлайн методу кластеризації, що базуються на використанні непрямих ознак часового ряду, які дозволяють розподілити вхідні дані на класи – групи однотипних екземплярів вибірки для подальшого прогнозування.

Для досягнення мети необхідно вирішити такі завдання:

– проаналізувати загальний процес кластеризації послідовностей та його особливості;

– провести аналіз існуючих алгоритмів кластеризації;

– провести аналіз використання непрямих характеристик даних;

– провести імітаційне моделювання обраних методів для побудови системи кластеризації послідовностей даних за допомогою обраних методів;

– проаналізувати якість побудованої системи кластеризації.

2 АНАЛІЗ АЛГОРИТМІВ ПОПЕРЕДНЬОЇ ОБРОБКИ ЧАСОВИХ РЯДІВ ТА АЛГОРИТМІВ ON-LINE КЛАСТЕРИЗАЦІЇ

В процесі аналізу було розглянуто:

– деякі статистичні гіпотези властивостей часового ряду у процесі попереднього аналізу вибірки;

– статичні методи дослідження часових рядів;

– алгоритми зменшення розмірності та формування векторів-ознак;

– міри відстаней послідовних даних;

– алгоритми on-line кластеризації.

Загальний процес on-line кластеризації за непрямими вимірами можна описати у вигляді таких ключових етапів:

– формування векторів-ознак за непрямими вимірами;

– попередня обробка послідовностей, а саме згладжування шумів, аналіз тренду;

– подання даних на on-line кластеризацію;

– оцінювання ефективності кластеризації за непрямими вимірами.

2.1 Перевірка статистичних гіпотез властивостей часового ряду у процесі попереднього аналізу вибірки

2.1.1 Перевірка гіпотези о наявності аномальних спостережень

Рівні часових рядів можуть мати аномальні значення. Поява таких значень може бути викликано помилками під час збирання інформації, запису чи передачі – це помилки технічного порядку, чи помилки першого ряду, тобто. помилки, що не відображають жодної тенденції. Однак, аномальні значення можуть відображати реальні процеси, наприклад, стрибок курсу долара або його падіння, такі аномальні значення відносять до помилок

другого ряду, вони не підлягають усуненню. Для виявлення аномальних рівнів часових рядів можна використовувати метод Ірвіна.

Для спостережень часового ряду формуються такі гіпотези:

– H_0: i-е спостереження не є аномальним;

– H_1: i-е спостереження аномальне.

Щоб перевірити дані гіпотези, потрібно обчислити значення критерію за такою формулою:

$$I_i = \frac{|y_i - y_{i-1}|}{S_y}, \tag{2.1}$$

де

$$S_y = \overline{\frac{\sum_{i=1}^{n}(y_i - y)^2}{n-1}}, y = \frac{1}{n}\sum_{i=1}^{n} Y_i. \tag{2.2}$$

S_y представляє собою середнєквадратичне відхилення, яке підраховано для вибірки n.

Якщо величина I_i попадає в критичну точку (перевищує граничне значення), то гіпотеза H_0 відкидається з ймовірністю помилки першого роду та приймається гіпотеза H_1 (визначається аномальним спостереженням).

2.1.2 Перевірка гіпотези о наявності невипадкової величини

Наявність невипадкової величини часового ряду, тобто тренду, можна отримати перевіривши гіпотезу про постійность середнього значення послідовності.

Розглянемо дві статичні гіпотези:

$$H_0: M\ Y\ i = const, \tag{2.3}$$

$$H_1: M\ Y\ i \neq const. \tag{2.4}$$

При перевірці цих гіпотез початкова вибірка поділяється на дві рівномірні частини. Обидві частини розглядаються як самостійні сукупності, що нормально не розподілені, як більшість часових рядів. Для таких вибірок використовують непараметричні критерії оцінювання: критерій Мана-Уітні для математичного очікування [12] та критерій Сіджела-Тюки для дисперсії.

Якщо отримане значення критерію більше критичного значення (знаходиться за таблицею критерію), то основна гіпотеза відхиляється, що означає вибірки не рівні між собою, тобто часовий ряд має виражений тренд [13].

2.2 Формування векторів-ознак на основі статистичних методів дослідження часових рядів

Для будь-яких статистичних досліджень важливо, щоб дані були відредаговані та опрацьовані перед тим, як вони будуть використовуватися в подальшому аналізу. Така перевірка початкових даних необхідна для виявлення грубих помилок у дослідженні, а також помилок, допущених при кодуванні і перетворенні даних; виявлення можливих шумів або аномальних спостережень.

Можна виділити такі основні методи формування ознак, в основі яких лежать статистичні дослідження: один з методів регресійного аналізу – метод найменших квадратів, метод максимальної правдоподібності та методи згладжування.

Метод найменших квадратів призначується для оцінки невідомих величин за вимірами, що можуть мати випадкову похибку. Цей метод є простим і ефективним для вибраної структури моделі функціональної залежності та оптимальних значень параметрів; вибрати структуру та модель може сам аналітик; отримання оптимальних значень коефіцієнтів за класом заданих структур; отримати оптимальну структуру моделі [14, 15].

Метод максимальної правдоподібності – це метод оцінки невідомого параметру шляхом максимізації функції правдоподібності:

$$L\left(X_1, \dots, X_n; \widehat{\theta_{\text{МП}}}\right) = maxL(X_1, X_2, \dots, X_n\theta), \quad (2.5)$$

де $L\left(X_1, \dots X_n; \hat{\theta}\right) = f(X_1; \theta) \times f(X_2; \theta) \times \dots \times f(X_n; \theta)$ – функція правдоподібності;

$f(X_1; \theta)$ – закон розподілу вірогідності.

Зазначений метод має такі недоліки:

– потрібне точне визначення типу закону розподілу;

– погано працює з вибірками великих обсягів [16, 17].

Методи згладжування часового ряду:

– метод ковзкої середньої;

– метод експоненціального згладжування.

Розглянемо ці методи більш детально.

Метод ковзкої середньої – один із емпіричних методів для згладжування та прогнозування часових рядів. Суть полягає в тому, що абсолютні значення ряду динаміки змінюються на середні арифметичні значення у певні інтервали.

Формула цього методу може бути представлена у вигляді:

$$S_i = \frac{1}{2k+1} \sum_{j=-k}^{k} x_{i+j}, \quad (2.6)$$

де x_i – вихідний ряд;

S_i – згладжений ряд.

До основних недоліків використання методу ковзкої середньої належать:

– ковзке середнє неефективне у обчисленні, для кожної точки середнє необхідно перераховувати за новою – не можна повторно використати результат, обчислений для попередньої точки;

– ковзке середнє не можна продовжити на перші та останні точки ряду, тобто це може спричинити проблему, якщо нас цікавлять саме ці точки.

Метод експоненціального згладжування – це метод , що є більш якісним та позбавляє зазначених недоліків у попередньому методі.

Метод експоненціального згладжування обчислює значення згладженого ряду шляхом оновлення значень, розрахованих на попередньому кроці, використовуючи інформацію з поточного кроку. Інформація з попереднього та поточного кроків береться з різними вагами, якими можна керувати.

В простішому варіанті одинарного згладжування відношення таке:

$$S_i = ax_i + (1 - a)S_{i-1}, \tag{2.7}$$

де $0 \leq \alpha \leq 1$.

Параметр α визначає співвідношення між незгладженим значенням на поточному кроці та згладженим значенням попереднього кроку. При *α=1* братимемо лише точки вихідного ряду, тобто. ніякого згладжування не буде. При *α=0* ряд братимемо лише згладжені значення з попередніх кроків, тобто. ряд перетвориться на константу.

Однак, якщо в даних є тренд, просте згладжування буде «відставати» від нього (або доведеться брати значення близькими до 1, але тоді згладжування буде недостатнім). Потрібно використовувати подвійне експонентне згладжування.

Подвійне згладжування використовує вже два рівняння. Одне рівняння оцінює тренд як різницю між поточним та попереднім згладженим значеннями, потім згладжує тренд простим згладжуванням. Друге рівняння виконує згладжування як у випадку простого варіанта, але у другому доданку використовується сума попереднього згладженого значення та тренду.

Потрійне згладжування включає ще один компонент – сезонність і використовує ще одне рівняння. При цьому розрізняються два варіанти

сезонного компонента – адитивний та мультиплікативний. У першому випадку амплітуда сезонного компонента стала і з часом не залежить від базової амплітуди ряду. У другому випадку амплітуда змінюється разом із зміною базової амплітуди низки.

Враховуючі усі особливості використання, реалізації та недоліки для використання дослідження часових рядів, що послідовно надходять на обробку кращим є використання саме методу експоненціального методу згладжування, адже він оброблює кожну точку ряду та призначений для on-line аналізу даних [18-21].

На рисунку 2.1 зображено різницю у згладжуванні двома вищеописаними методами у порівнянні.

Рисунок 2.1 – Приклад згладжування методом ковзкої середньої та експоненціальним

2.3 Методи зменшення розмірності

Зважаючи на те, що послідовні дані зазвичай надходять до аналізу та обробки у великих обсягам та мають корельовані дані доцільним є проводити зменшення розмірності без втрати корисної інформації для можливості використання більшої кількості можливих алгоритмів та кращої ефективності.

Одним з методів, що найчастіше використовується можна назвати метод головних компонент(PCA) – метод, в якому за допомогою лінійного перетворення початкових величин здійснюється перехід до нової системи вже некорельованих ознак, які називають основними компонентами. У разі нормального розподілу таке перетворення призведе до системи незалежних ознак. Питання зниження розмірності вирішуватиметься у цій новій системі ознак у припущенні, що інформативність ознаки безпосередньо пов’язані з його мінливістю [22].

Таким чином із сукупності вихідних характеристик для подальшого аналізу залишаються лише ті, яким властивий «суттєво» більший розкид значень, ніж іншим.

Популярність PCA визначається двома важливими властивостями. По-перше, це оптимальна (по середній квадратичній помилці) лінійна схема для стиснення множини векторів високої розмірності у безліч векторів нижчої розмірності, а потім їх відновлення. По-друге, параметри моделі можуть бути обчислені безпосередньо з даних.

Однак, незважаючи на ці привабливі особливості, модель PCA має декілька недоліків. Один з них полягає в тому, що прямі (стандартні, звичні) методи знаходження напрямів головних компонентів за допомогою розкладання матриці других моментів призводять до проблем при великому обсязі даних високої розмірності. Ще один недолік стандартних підходів до PCA полягає в тому, що неясно, як правильно поводитися з неповними даними. Зазвичай вони або відкидаються, або заповнюються виходячи з наявних значень. Однак такий підхід виявляється вкрай неефективним, коли частка пропущених спостережень значна.

Додаткові труднощі при застосуванні стандартних методів PCA доставляють відхилення від прийнятої моделі даних: зокрема, стандартний алгоритм PCA заснований на припущенні, що дані не зіпсовані викидами.

Незважаючи на вище перераховані недоліки використання методу головних компонент він є доволі популярним методом для зменшення

розмірності часових рядів у нових покращених алгоритмах, що вирішують проблему, як прокльону розмірності, так і добре працюють з пропущеними та зашумленими даними.

2.4 Міри відстані для послідовних даних

Одним з ключових компонентів кластеризації є функція, яка використовується для вимірювання подібності між двома даними, що порівнюються. Ці дані можуть бути в різних формах, включаючи вихідні значення однакової або неоднакової довжини, вектори–ознак значень, матриці переходів тощо.

Нижче будуть розібрані деякі з мір відстані, які найчастіше використовуються у алгоритмах on-line кластеризації.

2.4.1 Евклідова відстань, середньоквадратична відстань та відстань Міковського

Нехай x_i та v_j є P-вимірним вектором. Евклідова відстань обчислюється як:

$$d_E = \sqrt{\sum_{k=1}^{P}\left(x_{ik} - v_{jk}\right)^2}. \tag{2.8}$$

Середньоквадратична відстань (або середня геометрична відстань) є простою:

$$d_{rms} = d_E/n. \tag{2.9}$$

Відстань Мінковського є узагальненням евклідової відстані, який визначається як:

$$d_M = \sqrt[q]{\sum_{k=1}^{P}\left(x_{ik} - v_{jk}\right)^2}. \quad (2.10)$$

У наведеному вище рівнянні q – ціле додатне число. Нормована версія може бути визначена, якщо вимірювані значення нормалізуються шляхом ділення на максимальне значення в послідовності.

2.4.2 Динамічна деформація часу (DTW)

Міра динамічної деформації часу – це узагальнення класичних алгоритмів порівняння дискретних послідовностей з послідовностями безперервних значень. Ця міра використовуються у ситуаціях, коли ряди мають різну довжину. На рисунку 2.2 зображено графічний приклад відмінності з евклідовою відстанню.

Враховуючи два часові ряди, $Q = q_1, q_2,...,q_i,...,q_n$ і $R = r_1, r_2,...,r_j ,...,r_m$, DTW вирівнює два ряди так, щоб їхня різниця була зведено до мінімуму. Для цього створена матриця $n \times m$, де елемент (i, j) матриці містить відстань $d(q_i, r_j)$ між двома точками q_i і r_j. Зазвичай використовується евклідова відстань. Шлях деформації, $W = w_1, w_2,...,w_k,...,w_K$, де $max(m, n) \leq K \leq m + n - 1$, являє собою набір елементів матриці, що задовольняє трьом обмеженням: гранична умова , безперервність і монотонність.

Обмеження граничної умови вимагає, щоб шлях деформації почався і закінчився в діагонально протилежних кутових клітинках матриці. Тобто $w_1 = (1, 1)$ і $w_K = (m, n)$.

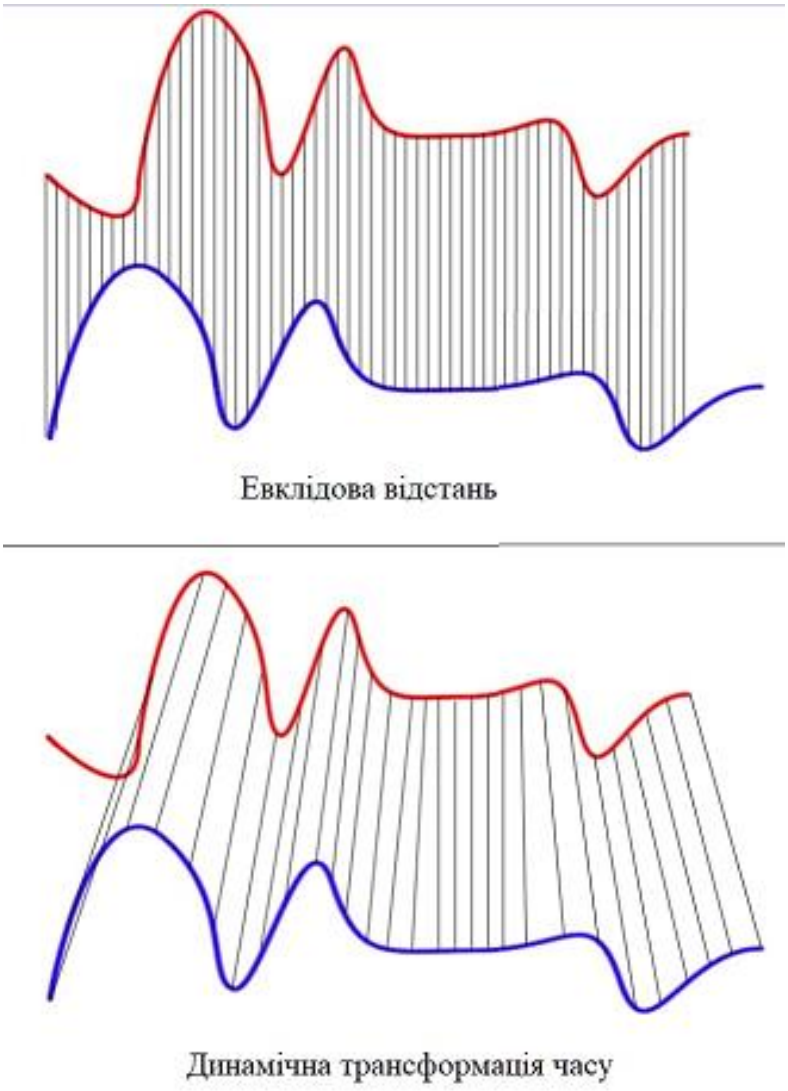

Рисунок 2.2 – Графічний приклад різниці мір

Обмеження безперервності обмежує допустимі кроки сусідніми осередками. Обмеження монотонності змушує точки на шляху деформації бути монотонно рознесеними в часі.

Цікавим є шлях деформації, який має мінімальну відстань між двома рядами. Математично це виглядає так:

$$d_{DTW} = min \frac{\sum_{k=1}^{K} \omega_k}{K}. \quad (2.11)$$

Динамічне програмування можна використовувати для ефективного пошуку цього шляху, оцінюючи наступну повторюваність, яка визначає

кумулятивну відстань як суму відстані поточного елемента та мінімуму сукупних відстаней сусідніх елементів:

$$d_{cum}(i,j) = d(q_i, r_j) + min\{d_{cum}(i-1, j-1), \\ d_{cum}(i-1, j-1), d_{cum}(i, j-1)\}. \quad (2.12)$$

2.5 Алгоритми on-line кластеризації

У задачі кластеризації часових рядів, де дані надходять послідовно та у великій розмірності потрібно використовувати on-line алгоритми кластеризації [23-25].

2.5.1 Stream on-line алгоритм

Алгоритм STREAM заснований на методології *k*-медіан кластеризації. Основна ідея полягає в тому, щоб розбити потік на менші резидентні сегменти пам'яті. Таким чином, вихідний потік даних S ділиться на відрізки S_1 ...S_r. Кожен сегмент містить не більше m точок даних. Значення m фіксується на основі попередньо визначеного значення у пам'яті. Оскільки кожен сегмент S_i поміщається в основну пам'ять, можна використовувати більш складний алгоритм кластеризації, не турбуючись про обмеження у неможливості лише одного проходу через дані.

В алгоритмах *k*-медіан набір Y з k представників вибирається з кожного блоку S_i, і кожна точка в S_i призначається до її найближчого представлення. Основна мета полягає в тому, щоб обрати представлення для мінімізації суми квадратів відстані визначених точок даних до цих представлень.

Після обробки першого сегмента S_1 маємо набір з k медіан, які зберігається. Кількість балів, присвоєних кожному представленню,

зберігається як «вага» для цього представлення. Такі представлення вважаються представленнями першого рівня.

Наступний сегмент – S_2 обробляється незалежно, щоб знайти його k оптимальних медіанних представлень. Таким чином, наприкінці обробки другого відрізка буде $2 \times k$ таких представлень. Тобто, потреба в пам'яті для зберігання представлень також зростає з часом та після обробки r сегментів один матиме загалом $r \times k$ представлень. Коли число представлень перевищує m, до цих множин $r \times k$ застосовується другий рівень кластеризації, за винятком того, що збережені ваги представлень також використовуються в кластеризації. Отримані представлення зберігаються як представники другого рівня.

В кінці обробки весь потік даних (або коли виникає конкретна потреба в результаті кластеризації), усі отримані раніше представлення різних рівнів об'єднуються в одну кінцеву програму k-медіан.

Конкретний вибір алгоритму, який використовується для проблеми k-медіан, є критичним для забезпечення якісної кластеризації. Іншим фактором, який впливає на якість кінцевої продукції, є ефект необхідності ієрархічної кластеризації, який проявляються у необхідності декомпозиції на частини.

Основним обмеженням алгоритму STREAM є те, що він не є особливо чутливим до змін в базовому потоці даних. У багатьох випадках потік даних може еволюціонувати і значно змінюватися. Тому це важливо для процесу кластеризації, щоб мати можливість адаптуватися до таких змін і надавати уявлення за різні часові горизонти. У цьому сенс і, алгоритм CluStream здатний надати значно кращу інформацію на різних рівнях часової деталізації.

2.5.2 CluStream on-line алгоритм

Наявність змін в потоці даних, що розвивається, значно змінює кластери з часом. Кластери за минулий день дуже відрізняються від кластерів за останній місяць. У багатьох інтелектуальних задачах аналізу даних аналітики, можливо, з’явиться бажання мати гнучкість для визначення кластерів, основаних на одному або кількох часових горизонтах, які невідомі на початку процесу кластеризації потоків.

Оскільки потокові дані, природно, накладають обмеження на можливість багатопрохідного алгоритму, то при розробці алгоритмів важко обчислити кластери на різних часових етапах за допомогою звичайних алгоритмів. Пряме розширення алгоритму STREAM на такий випадок вимагатиме алгоритму з одночасної підтримки проміжних результатів кластеризації на всіх можливих часових горизонтах. Обчислювальне навантаження такого підходу збільшується з прогресуванням потоку даних і може швидко стати слабким місцем для on-line виконання.

Підходом до вирішення цієї проблеми є застосування процесу кластеризації з двоетапною методологією, включаючи етап мікрокластеризації в режимі онлайн та off-line макрокластеризацію.

Етап мікрокластеризації онлайн обробляє потік в режимі реального часу безперервно та підтримує детальну кластерну статистику потоку. Їх називають мікрокластери.

Етап макрокластеризації в автономному режимі далі підсумовує ці детальні кластери, щоб надати користувачеві більш стисле розуміння кластерів за різними часовими етапами та про рівні часової деталізації. Це досягається шляхом достатнього зберігання детальної статистики в мікрокластерах, щоб можна було повторно кластерувати ці детальні представлення за визначеними користувачем часовими етапами.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Класичний аналіз часових рядів перед процесом кластеризації включає в себе виявлення закономірностей (наприклад, аналіз трендів, сезонності, автокореляцію, моделі авторегресії). Традиційна обробка часових рядів нерідко передбачає фільтрацію, амплітудну нормалізацію, видалення базового рівня часового ряду, лінійного тренду і т.д. Але для деяких даних такі процеси можуть бути обчислювано громоздкими, або, наприклад для обробки відеопотоків взгалі не мають сенса.

Коли маємо справу с великими даними, що послідовно надходять на обробку виникає проблема для дослідження усіх надходжених даних у звязку з кількістю спостережень , високою складністю реалізція такої системи та її підтримки , тому є логічних використання не сирих даних , а непрямих вимирів цих даних (середнє і т.д.), тобто кластеризувати методом за ознаками що забезпечує ефективніші результати та просту реалізацію. Непрямі виміри повністю характеризують початкові дані, краще працюють з on-line алгоритмами. Ще потрібно видиліти те, що після їх отримання дані будуть представлені у меншій розмірності, що також веде до оптимізації кластеризації.

Таким чином, проаналізувавши існуючі підходи кластеризації, статистичні методи дослідження часових рядів та алгоритми кластерізацій, доцільним є дослідження on-line кластеризації за непрямими даними та оцінювання ефективності такого методу. Далі буде представлено поетапний процес досліджованого алгоритму кластеризації з описом використовуваних алгоритмів.

3.1 Алгоритм обчислення середніх Велфорда

У досліджуваній задачі on-line кластеризації за непрямими вимірами отримані після предобробки дані надходять у послідовному режимі за один прохід. Тому потрібно використовувати такий алгоритм формування векторів–ознак, що дозволить послідовне обчислення непрямих вимірів за один прохід алгоритму. Саме таким алгоритмом є метод Велфорда.

Оригінальна стаття Велфорда була опублікована в 1962 році. Це простий та ефективний метод обчислення середніх, дисперсій, коваріацій та інших статистик. Тим не менш, не можна сказати, що алгоритм хоч якийсь широко відомий в даний час [26, 27].

В процесі кластеризації потрібно отримати вектори-ознаки, що будуть представлені у вигляді непрямих вимірів, а саме середнє та дисперсія. Пропонується проводити обчислення за допомогою алгоритму Велфорда. В нашому випадку корисно обчислювати, наприклад, дисперсію за один прохід, перевіряючи кожне спостереження у on-line режимі.

Можна виділити такі переваги для використання цього алгоритму:

– ефективний і корисний коли дані збираються у великій розмірності при недостатній пам’яті;

– ефективний коли витрати на доступ до пам’яті перевищують витрати на обчислення;

– досягає відмінних показників за точністю рішень;

– легкий у реалізації;

– однопрохідний on-line алгоритм, що вкрай корисно у деяких ситуаціях.

3.2 Кластеризація за непрямими вимірами

3.2.1 Формування векторів-ознак за непрямими вимірами

Нехай вихідні спостереження подано у вигляді багатовимірної послідовності:

$$x(1), \dots ., x(k), \dots x(N), \tag{3.1}$$

$$x(k) = (x_1(k), x_2(k), \dots x_n(k))^T, \tag{3.2}$$

де *k = 1, 2, ..., N* позначає поточний дискретний час.

Відповідно до задачі кластеризації необхідно кластеризувати *x(k)* у *k = 1, 2, ..., N* кластерів. Непрямий підхід до кластеризації вирішує це питання визначенням попарної подібності або непрямими ознаками даних, наприклад середнє, дисперсія та автокореляція [28-32].

Тобто для кожного і елементу можна використовувати такі періодичні оцінки:

– для середньої:

$$\overline{x_j}(k) = \overline{x_j}(k-1) + \frac{1}{k}\left(\overline{x_j}(k) - \overline{x_j}(k-1)\right); \tag{3.3}$$

– для дисперсії:

$$\sigma_j^2(k) = \sigma_j^2(k) - \frac{1}{k}\left(x_j(k) - \overline{x}_j(k)^2 - \sigma_j^2(k-1)\right)^2; \tag{3.4}$$

– для автокореляції:

$$r_j(k,\tau) = r_j(k-1,\tau) + \frac{1}{k}\Big(\left(x_j(k) - \overline{x}_j(k)\right)\left(x_j(k-\tau) - \overline{x}_j(k)\right) - \\ -r_j(k-1,)\tau\Big). \tag{3.5}$$

Позначаючи $\bar{x}(k) = (\bar{x}_1(k), \bar{x}_2(k) \ldots \bar{x}_n(k))^T$ та припускаючи, що $\tau = 0,1,2, \ldots \tau_{max}$ означає відставання в часі досить легко ідентифікувати векторно-матрічні аналоги:

$$\overline{x}(k) = \overline{x}(k-1) - \frac{1}{k}\big(x(k) - \overline{x}(k-1)\big), \tag{3.6}$$

$$R(k,\tau) = R(k-1,\tau) + \frac{1}{k}\Big(\big(x(k) - \overline{x}(k)\big)\big(x(k-\tau) - \overline{x}(k)\big)^T - R(k-1,\tau)\Big), \tag{3.7}$$

де діагональні елементи симетричної матриці *R(k,0)* є фактично оцінками дисперсій $r_{ij}(k,0) = \sigma_i^2(k)$ та недіагональні елементи складають коефіцієнти взаємної кореляції $r_{ij}(k,0)$, *i, j = 1, 2, 3, …n.*

3.2.2 Експоненційне згладжування

На сьогодні не існує автоматичного способу знаходження тренду часового ряда. Але якщо тренд є доволі монотонним (стійко зростає або стійко зменшується), то аналізувати послідовні дані стає не так важко. Якщо тимчасові ряди містять значну помилку, першим кроком виділення тренда є згладжування.

Для того, щоб забезпечтити адаптивні властивості рекурентних даних (3.3) – (3.7) є доцільним застосувати модифікацію алгоритму згладжування до характеристик, що будуть подаватися на кластеризацію:

$$\overline{x}_j^{\alpha}(k) = \alpha x_j(k) + (1-\alpha)\overline{x}_j^{\alpha}(k-1), 0 < \alpha < 1, \tag{3.8}$$

$$\left(\sigma_j^{\alpha}(k)\right)^2 = \alpha\left(x_j(k) + \overline{x}_j^{\alpha}(k)\right)^2 + (1-\alpha)\left(\sigma_j^{\alpha}(k-1)\right)^2, \tag{3.9}$$

$$r_j^{\alpha}(k,\tau) = \alpha\left(x_j(k) - \overline{x}_j^{\alpha}(k)\right)\left(x_j(k-\tau) - \overline{x}_j^{\alpha}(k)\right) + (1-\alpha)r_j^{\alpha}(k-1,\tau), \quad (3.10)$$

де $\alpha = 2/(L+1)$ це коефіцієнт забування, який забезпечує згладжування на ковзному вікні, що містить останні L спостережень $x_i(k), x_i(k-1), \ldots x_i(k-L+1)$.

Таким чином, $(2+\tau_{max})\times 1$ вектор-ознак $\tilde{x}_i(k) = \left(\overline{x}_i^{\alpha}(k),\ (\sigma_i^{\alpha}(k))^2,\ r_i^{\alpha}(k,1)\ldots r_i^{\alpha}(k,\tau_{max})\right)^T$ може бути визначеним у відповідності з кожним компонентом часового ряду $x_i(k)$, тобто саме цей вектор представляє кластерні об'єкти.

3.2.3 On-line кластеризація

Після обробки часового ряду вище описаними методами можна подавати отримані згладжені вектори до кластеризації. До векторів-ознак будуть застосовані fuzzy алгоритми кластеризації засновані на fuzzy цільовій функції [33-36].

Стандартний критерій самонавчання використовується як цільова функція:

$$E\left(\mu\left(\tilde{x}_j(k), C_j(I)\right)\right) = \sum_{k=1}^{N}\sum_{j=1}^{m}\mu^{\beta}\left(\tilde{x}_j(k), C_j(I)\right)D^2\left(\tilde{x}_j(k), C_j(I)\right), \quad (3.11)$$

з обмеженнями:

$$\sum_{j=1}^{M}\mu\left(\tilde{x}_j(k), C_j(I)\right) = 1, \forall k = 1,2,\ldots N, \quad (3.12)$$

$$0 < \sum_{k=1}^{M}\mu\left(\tilde{x}_j(k), C_j(I)\right) \le N, \forall j = 1,2,\ldots M. \quad (3.13)$$

Тут $\mu(\tilde{x}_i(k),C_i(I))$ є рівнем вектору $\tilde{x}_i(k)$ визначеного кластеру, $C_i(I)$ – центроїд кластера, що потрібно знайти, β – позитивний параметр, що називається fuzzifier та контролює кількість пересічених кластерів.

$D^2(\tilde{x}_i(k),C_i(I))$ визначає деяку міру, що є відстанню між вектором $\tilde{x}_i(k)$ і кластером $C_i(I)$ використовуючи обрану метрику (найчастіше використовують Евклідову відстань).

Таким чином цільова функція може бути представлена у такому вигляді:

$$E\left(\mu\left(\tilde{x}_j(k), C_j(I)\right)\right) = \sum_{k=1}^{N}\sum_{j=1}^{m}\mu^2\left(\tilde{x}_j(k), C_j(I)\right)\left\|\tilde{x}_j(k) - C_j(I)\right\|^2. \quad (3.14)$$

Зважаючи на те, що нам потрібно отримати алгоритм, що оброблює дані, які послідовно надходять до входу отримуємо оптимізовану функцію кластеризації:

$$\begin{cases} \mu\left(\tilde{x}_j(k), C_j(I,k-1)\right) = \frac{\left\|\tilde{x}_j(k)-C_j(I,k-1)\right\|^{-2}}{\sum_{j=1}^{m}\left\|\tilde{x}_j(k)-C_j(J,k-1)\right\|^{-2}}, \\ C_j(I,k) = C_j(I,k-1) + \eta(k)\mu^{\beta}\left(\tilde{x}_j(k), C_j(I,k-1)\right), \\ \left(\tilde{x}_j(k) - C_j(I,k-1)\right), \end{cases} \quad (3.15)$$

де $0<\eta(k)<1$ параметр швидкості навчання, який загалом обирається емпіричним шляхом.

3.3 Обґрунтування вибору середовища програмної реалізації

При розробці імітаційної реалізації з кластеризації часового ряду за непрямими вимірами було обрано як мову програмування Python.

Python слід розглядати як універсальну та високорівневу мову програмування, основним завданням якого є підвищення продуктивності та

читабельності коду. Python є однією з найпопулярніших мов кодування, що має великий попит на ринку. Йому також притаманні деякі примітні функції, що роблять його достатньо доступним для різних сфер використання.

Python – мова високорівневого типу, яка підтримує такі види програмування:

– об’єктно-орієнтоване;
– структурне;
– функціональне;
– узагальнене;
– аспектно-орієнтоване.

Основний упор зроблений на читання коду та підвищення швидкості роботи програміста. Синтаксис ядра є мінімалістичним, а стандартна бібліотека має широкий набір функцій.

Архітектурні риси:

– підтримка багатопоточності;
– динамічна типізація;
– механізм обробки винятків;
– автоуправління пам’яттю;
– високорівневі структури даних;
– повна інтроспекція;
– розбиття програм на модулі, які можна об’єднувати в пакети.

Еталонна реалізація – інтерпретатор CPython – його підтримує більшість активно використовуваних платформ. Розповсюдження мови здійснюється під вільною ліцензією. Існують реалізації інтерпретатора для JVM, CLR, LLVM та інші. Проект PyPy використовує JIT – компіляцію, що збільшує швидкість виконання програм, написаних на Python.

Основні переваги:

– низький поріг входження. При адекватних зусиллях Python зрозуміє навіть новачок (підходить першою мовою);

– синтаксис простий, лаконічний, зрозумілий, тобто швидкість розробки підвищується;

– кросплатформність. Програмувати можна майже всіх платформах;

– велика та активна професійна спільнота, безліч доступної технічної та навчальної інформації;

– потужна підтримка великих компаній IT-індустрії;

– велика кількість додаткових фреймворків і бібліотек, що ще більше розширюють можливості використання мови (для вирішення того чи іншого комерційного завдання не треба щоразу винаходити велосипед);

– затребуваність Python-фахівців на сучасному ринку праці;

– широке застосування в різних сферах. Ця перевага є закономірним наслідком вищезазначених плюсів.

Python використовується у таких цілях:

– розробка систем штучного інтелекту, машинного навчання;

– розробка програм для мобільних пристроїв. Так, для гаджетів з ОС Android більше потрібний Java або Kotlin, для iOS – потрібна мова Swift або Objective-C. Однак на Python можуть розробляти серверну частину програм;

– створення десктопних програм для ПК (для розробки графічного інтерфейсу можна скористатися, наприклад, бібліотекою Tkinter);

– розробка вбудованих систем для різних цифрових пристроїв. Класичний приклад – Raspberry Pi. Також це можуть бути системи управління банкоматами та ін.;

– створення ігор. Тут слід згадати кілька серйозних проектів, де задіявся Python: Battlefield 2, World of Tanks. Також за його допомогою було створено Civilization IV. Якщо ж створювати ігри – це просто особисте хобі, зверніть увагу на фреймворк PyGame.

Python – мова програмування з динамічною типізацією, і нехай швидкість написання коду вище та в силу синтаксису – писати простіше, але за продуктивністю Python все ж таки поступається компілюваним мовам,

такими, як наприклад C++. Тому продуктивність можна віднести до слабких сторін Python.

В даній роботі були використані такі бібліотеки як NumPy, Pandas та Matplotlib.

3.3.1 NumPy

Пакет NumPy є незамінним помічником Python. Він тягне на собі аналіз даних, машинне навчання та наукові обчислення, а також суттєво полегшує обробку векторів та матриць. Деякі провідні пакети Python використовують NumPy як основний елемент своєї інфраструктури. До них відносяться scikit–learn, SciPy, pandas і tensorflow. Крім можливості розібрати по кісточках числові дані, вміння працювати з NumPy дає значну перевагу при налагодженні складніших сценаріїв бібліотек.

Предок NumPy, Numeric, був розроблений Джимом Хугуніном. Також було створено пакет Numarray із додатковою функціональністю. В 2005 році Тревіс Оліфант випустив пакет NumPy, додавши особливості Numarray в Numeric. Це проект із вихідним кодом, і у його розвитку взяло участь вже багато людей.

NumPy або Numerical Python – це бібліотека Python, яка пропонує таке:

– потужний *N*-мірний масив;

– високорівневі функції;

– інструменти для інтеграції коду C/C++ та Fortran;

– використання лінійної алгебри, Перетворень Фур’є та можливостей випадкових чисел.

Вона також пропонує ефективний багатовимірний контейнер загальних даних. За її допомогою можна визначати довільні типи даних.

3.3.2 Pandas

Для реалізації системи кластеризації обрано бібліотеку Pandas.

Pandas – це бібліотека Python для обробки та аналізу структурованих даних, її назва походить від «panel data» («панельні дані»). Панельними даними називають інформацію, отриману в результаті досліджень та структуровану у вигляді таблиць. Для роботи з такими масивами даних створено Pandas.

Pandas – це відкрита – бібліотека, тобто її вихідний код у відкритому доступі розміщено на GitHub. Користувачі можуть додавати туди свій код: вносити пояснення, доповнювати методи роботи та оновлювати розділи. Для роботи буде потрібний компілятор (програма, яка перекладає текст з мови програмування в машинний код) C/C++ та середовище розробки Python.

За допомогою Pandas фахівці можуть групувати та візуалізувати дані, створювати зведені таблиці та робити вибірку за певними ознаками.

Щоб аналізувати дані за допомогою Pandas, потрібно знати, як улаштовані структури цих даних усередині бібліотеки. Насамперед розберемо, що таке DataFrame та Series.

Pandas Series (серія) – це одномірний масив. Візуально він схожий на пронумерований список: ліворуч у колонці знаходяться індекси елементів, а праворуч самі елементи.

Pandas DataFrame – це двовимірний масив, схожий на таблицю/аркуш Excel (до речі, дані з Excel можна читати за допомогою команди pandas.read_excel(file.xls)). У ньому можна проводити такі ж маніпуляції з даними: об’єднувати групи, сортувати за певною ознакою, проводити обчислення. Як будь-яка таблиця, датафрейм складається зі стовпців і рядків, причому стовпцями будуть вже відомі об’єкти Series.

3.3.3 Matlotlib

Matplotlib – бібліотека, що призначена для візуалізації даних двовимірною (2D) графікою (3D графіка також підтримується).

Це бібліотека з відкритим вихідним кодом, яка допомагає у побудові графіків. Спочатку він був написаний Джоном Д. Хантером, який виявився нейробіологом. Він був автором Matplotlib під час його постлікарських досліджень в галузі нейробіології. Мета цієї бібліотеки полягала в тому, щоб вивчити діяльність, що відбувається в корі головного мозку пацієнтів, які страждають на епілепсію, шляхом побудови графіка цих дій на графіку. Єдиною метою побудови графіків було покращення візуалізації та вивчення загальних закономірностей у них. Перший випуск Matplotlib був у 2003 році.

З часом Matplotlib виявився однією з найбільш широко використовуваних бібліотек для побудови графіків поряд з мовою програмування Python для побудови графіків для даних та обчислювальних графів. Він не залежить від платформи і може працювати на Windows, Mac OS та Linux.

В цій роботі було використано модуль Pyplot у бібліотеці matplotlib. Pyplot – це інтерфейс прикладного програмування, який складається з функцій/методів, які допомагають обробляти дані для візуалізації. Подібність до Matlab полегшує роботу людям, які мають досвід роботи з MATLAB і навпаки. Деякі з них: сюжети, зображення, гістограми, таблиці, кругові діаграми, потокові графіки [37-39].

Можна виділити такі основні переваги використовування цієї бібліотеки:

– легка у використанні;

– надає високоякісні зображення та сюжети у різних форматах, таких як png, pdf;

– забезпечує керування різними елементами фігури, такими як DPI, колір фігури, розмір фігури.

3.4 Опис вхідних даних

На сьогоднішній день з розвитком інноваційних технологій банківської та фінансові сфери зазнають великих змін. Особливо кардинальні зміни помітні у сфері розрахунків та платежів. Багато людей починають використовувати криптовалюти, як валюту розрахунку за різні послуги. Тому, на мою думку, дуже актуальною є задача розгляду аналізу саме даних криптовалюти, як вибірки послідовних даних.

У проведеному дослідженні використовувався датасет зміни курса 22 криптовалют протягом з 1 січня 2018 року по 6 грудня 2019 року. Дані по криптовалютам були зібрані с сайту CoinMarketCap. Вхідний часовий ряд представлений у форматі CSV.

Досліджувана послідовності даних складається з 15510 спостережень. На рисунку 3.1 зображено приклад записів для криптовалюти bitcoin, яка складається з трьох змінних.

Змінна *y* – це вартість криптовалюти у американських доларах відмічена у день *ds*.

```
## Observations: 15,510
## Variables: 3
## $ y    <dbl> 7547.00, 7448.31, 7252.03, 7320.15, 7321.9...
## $ ds   <date> 2019-12-06, 2019-12-05, 2019-12-04, 2019-...
## $ coin <chr> "bitcoin", "bitcoin", "bitcoin", "bitcoin"...
```

Рисунок 3.1 – Приклад запису для криптовалюти bitcoin

На рисунку 3.2 графічно зображено як змінювався курс криптовалюти bitcoin протягом досліджуваного часу.

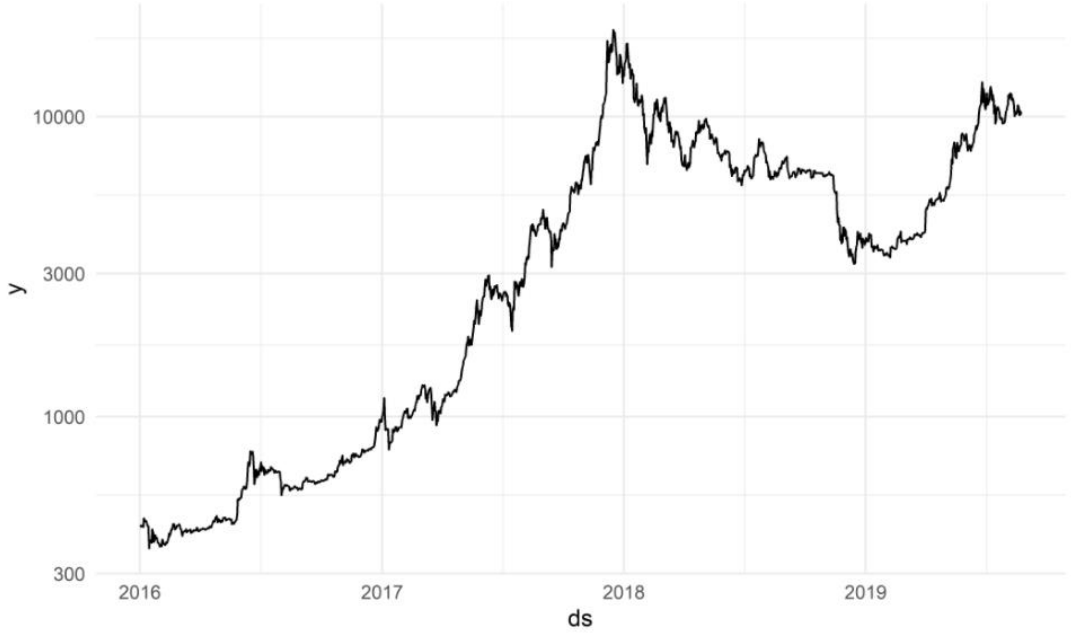

Рисунок 3.2 – Графічний вигляд динаміки зміни курсу криптовалюти bitcoin

На рисунку 3.3 зображено динаміку цін за всіма криптовалютами протягом майже двох років.

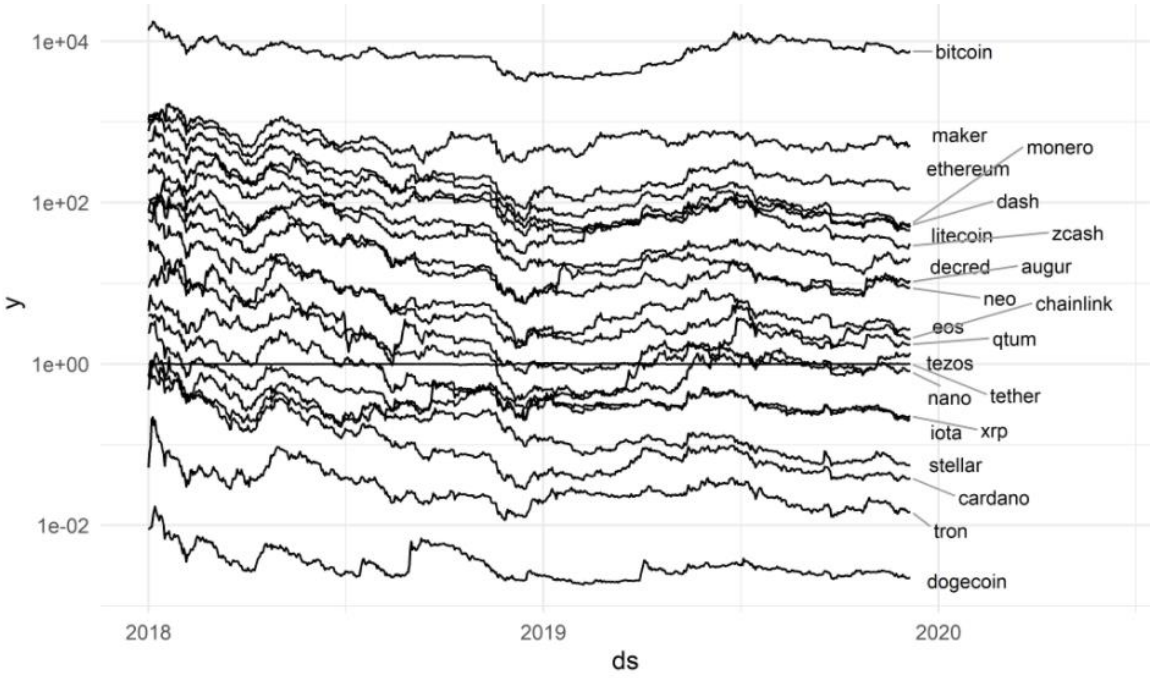

Рисунок 3.3 – Графічний вигляд динаміки зміни цін криптовалют

Можна побачити, що динаміка криптовалюти tether протягом усього часу досліджування була майже незмінною, тобто її часовий ряд представлений у вигляді майже горизонтальної лінії.

У демонстраційних цілях було проведено графічний аналіз динаміки змін криптовалют у вигляді часових рядів з ціллю отримання представлення про характер початкової вибірки. На рисунку 3.4 зображено ряд криптовалюти tether, що характеризується мінімальною вираженістю тренда.

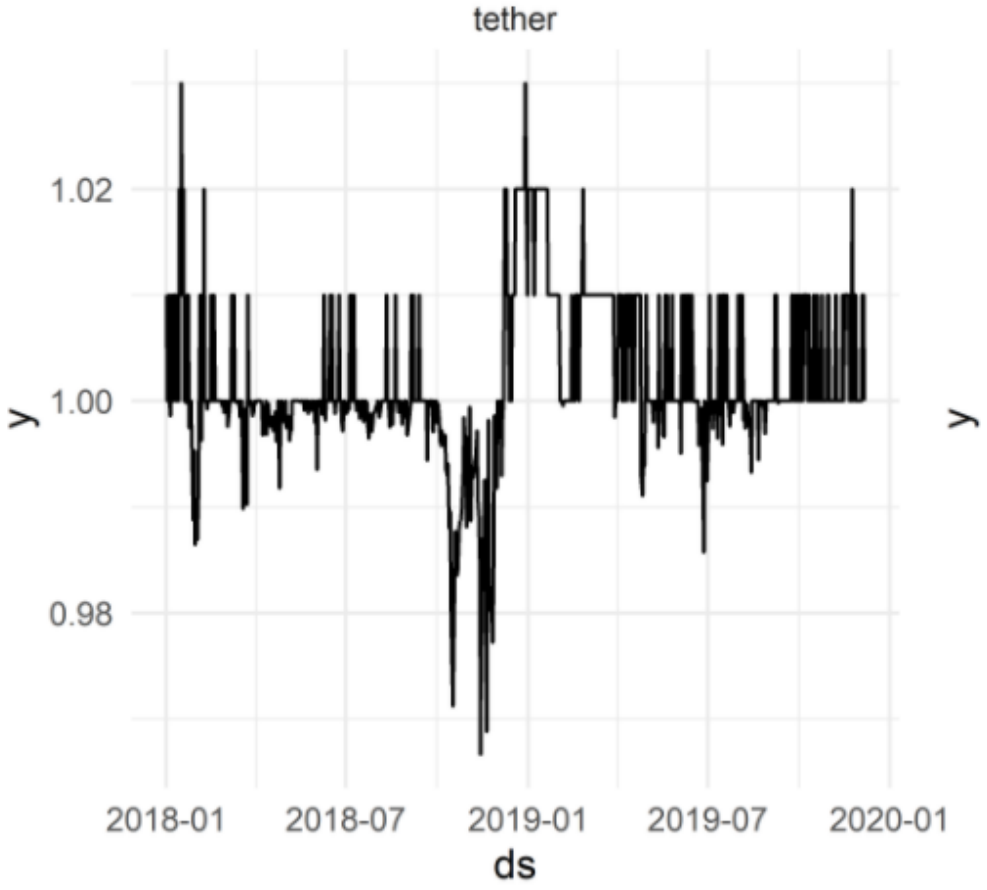

Рисунок 3.4 – Графічний вигляд ряду tether з мінімальною вираженістю тренда

На рисунку 3.5 зображено ряд dash з максимальною вираженістю тренда.

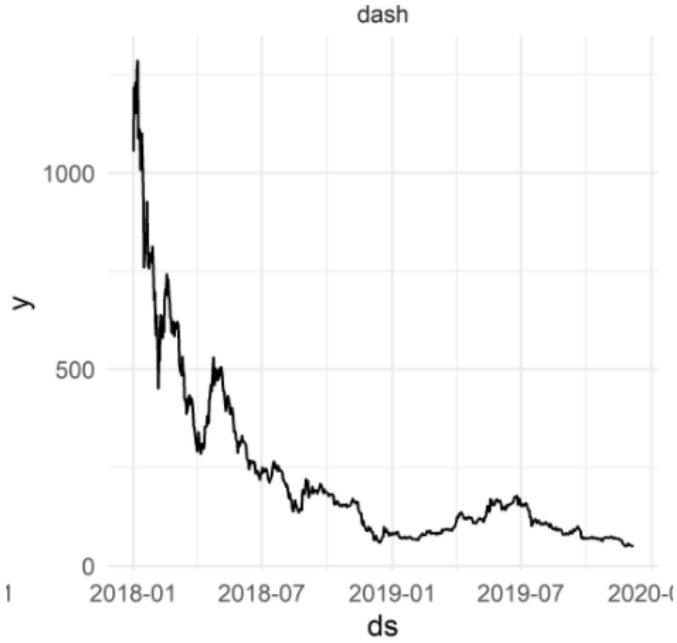

Рисунок 3.5 – Графічний вигляду ряду dash з максимальною вираженістю тренда

На рисунку 3.6 зображено ряд qtum, що характеризується, як вираженістю тренду, так і вираженістю недільної сезонності.

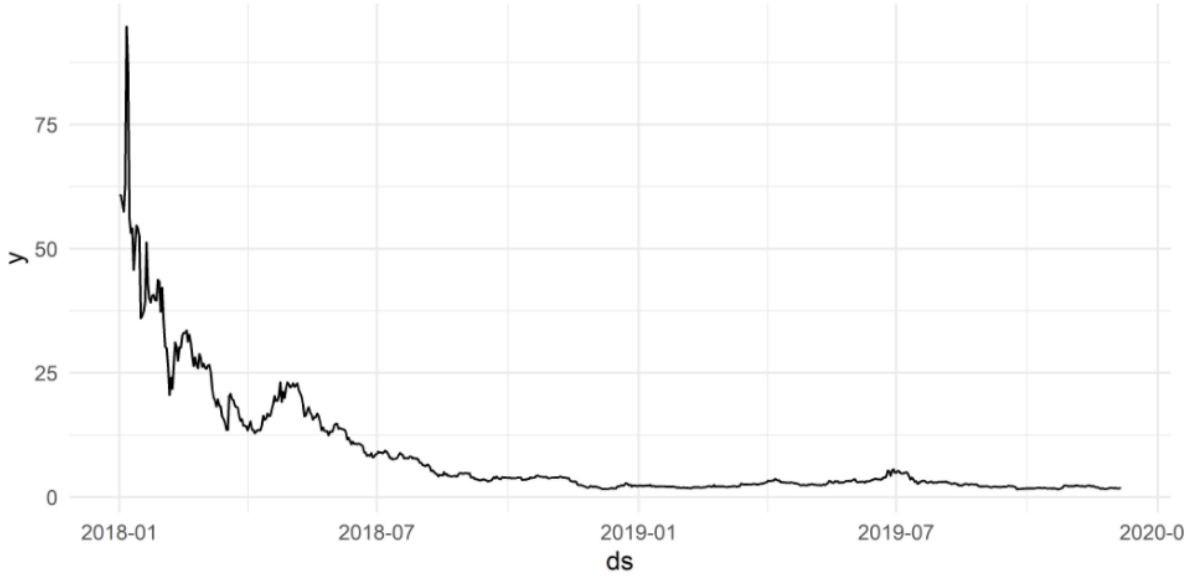

Рисунок 3.6 – Графічний вигляд часового ряду qtum з вираженим трендом та недільною сезонністю

3.5 Програмна реалізація

Як було вказано раніше, при розробці був використан однопрохідний алгорітм Велфорда для отримання векторів-ознак за непрямими вимірами, експоненційне згладжування середніх та fuzzy *c*-середніх алгоримт on-line кластеризації.

Було проведене дослідження роботи алгоритму на тестовому датасеті, особливості якого були описані у підрозділі 3.4.

На рисунку 3.7 зображено блок-схему досліджуваного алгоритму кластеризації послідовних даних.

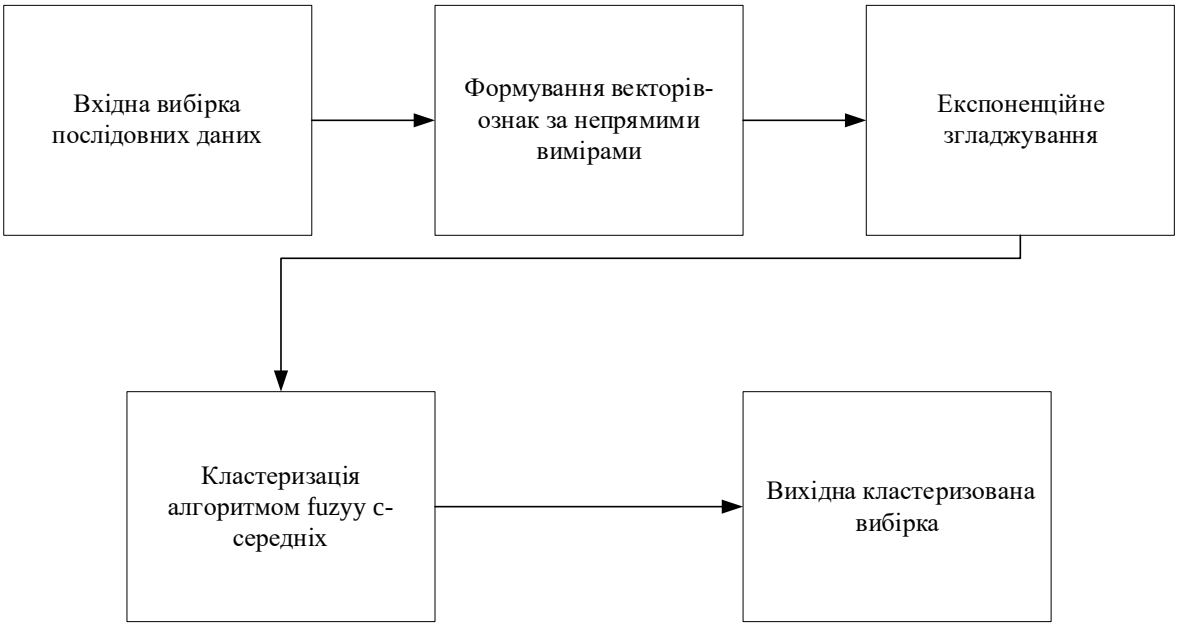

Рисунок 3.7 – Блок-схема роботи досліджуваного алгоритму

Згідно описаних підходів методів дослідження першим етапом є предобробка часових рядів, а саме процес формування векторів–ознак за непрямими вимірами, а саме середнє, дисперсія та автокреляція.

У лістингу 3.1 показано прикладу реалізаціїї однопрохідного алгоритму Велфорда для отримання дисперсії.

На рисунку 3.8 зображено графічне представлення дисперсії на прикладі криптовалюти bitcoin.

Лістинг 3.1 Реалізація обчислення дисперсії послідовних даних:

```
def add_vector(self, vector: np.ndarray, label=None):
current_avg = self.avg.val.copy() if self.avg.val is not None else vector.copy()
self.avg.add_vector(vector)
self.n += 1
if self.val is None:
self.val = np.zeros(vector.shape)
else:
self.val = self.val + ((vector – current_avg) * (vector – self.avg.val) – self.val) / self.n
if label:
self.changes[label] = self.val
```

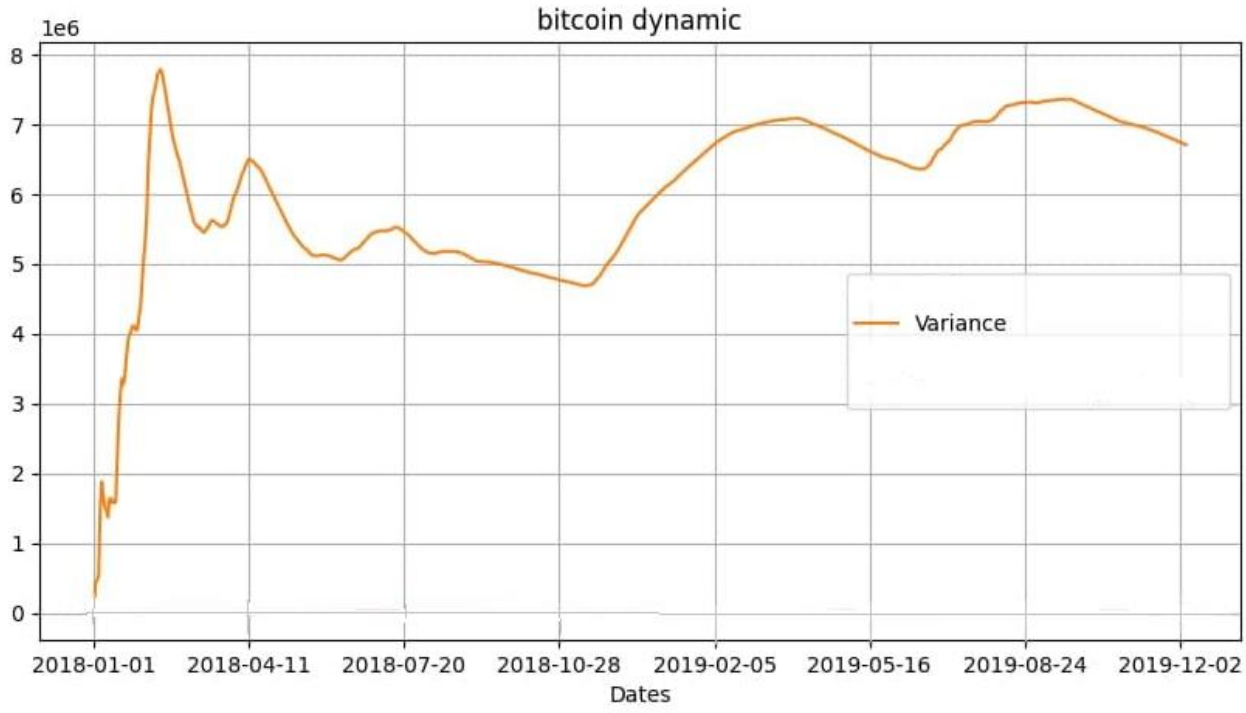

Рисунок 3.8 – Графічне зображення дисперсії на прикладі криптовалюти bitcoin

Сам процес формування дисперсій, середнього та автокореляції супроводжується експоненціальним згладжуванням. У лістингу 3.2 показано приклад реалізації формування середньої та експоненційного згладжування.

Лістинг 3.2 Реалізація обчислення дисперсії послідовних даних:

```
def add_vector(self, vector: np.ndarray, label=None):
self.n += 1
if self.val is None:
self.val = vector.copy()
if self.smoothing:
self.smoothed = self.val
else:
self.val = self.val + (vector – self.val) / self.n
if self.smoothing:
smoothing_rate = self.smoothing_rate
self.smoothed = smoothing_rate * self.val + (1 – smoothing_rate) * self.smoothed
if label:
self.changes[label] = self.val
if label and self.smoothing:
self.smoothed_changes[label] = self.smoothed
```

На рисунку 3.9 зображено графічне представлення середнього та експоненційно згладженого середнього на прикладі криптовалюти bitcoin.

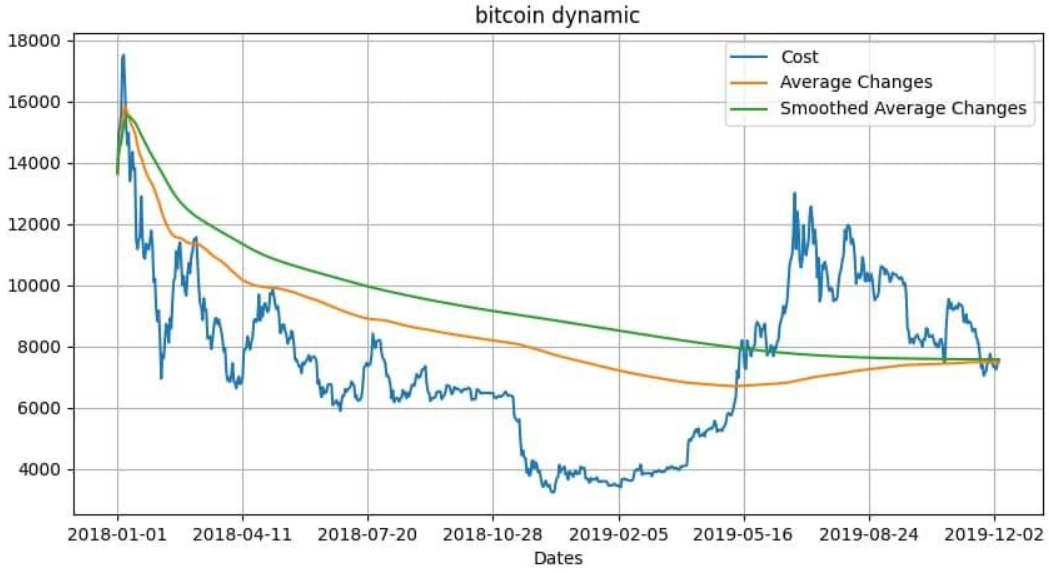

Рисунок 3.9 – Графічне зображення експоненційно згладженої середньої на прикладі криптовалюти bitcoin

Як можна побачити на графіку експоненційна функція показує згладжування тренду та шумів.

Після згладжування сформовані вектори-ознаки подаються на кластеризацію fuzzy алгоритмом за описаними функціями у розділі 3.2.

3.6 Отримані результати та їх аналіз

У результаті дослідження кластеризації за непрямими вимірами та реалізації імітаційного додатку кластеризації вхідні дані були розподілені між чотирма кластерами. Ряд tether був розподілений у кластер без інших криптовалют, що є очікуваним, адже цей ряд має мінімальну вираженість тренду серед інших. У кластері С4, наприклад, обидва ряди криптовалют dash та qtum характеризуються вираженою недільною сезонністю. Також можна побачити, що ряд bitcoin характеризується аномальністю серед інших рядів, тому очікувано він був розподілений у кластер С3.

Таким чином, проаналізувавши результати кластеризації можна зробити висновок, що підхід з використанням кластеризації за непрямими вимірами є ефективним та простим у реалізації.

На рисунку 3.10 графічно зображено результат кластеризації часових рядів динаміки зміни ціни криптовалют протягом майже двох років.

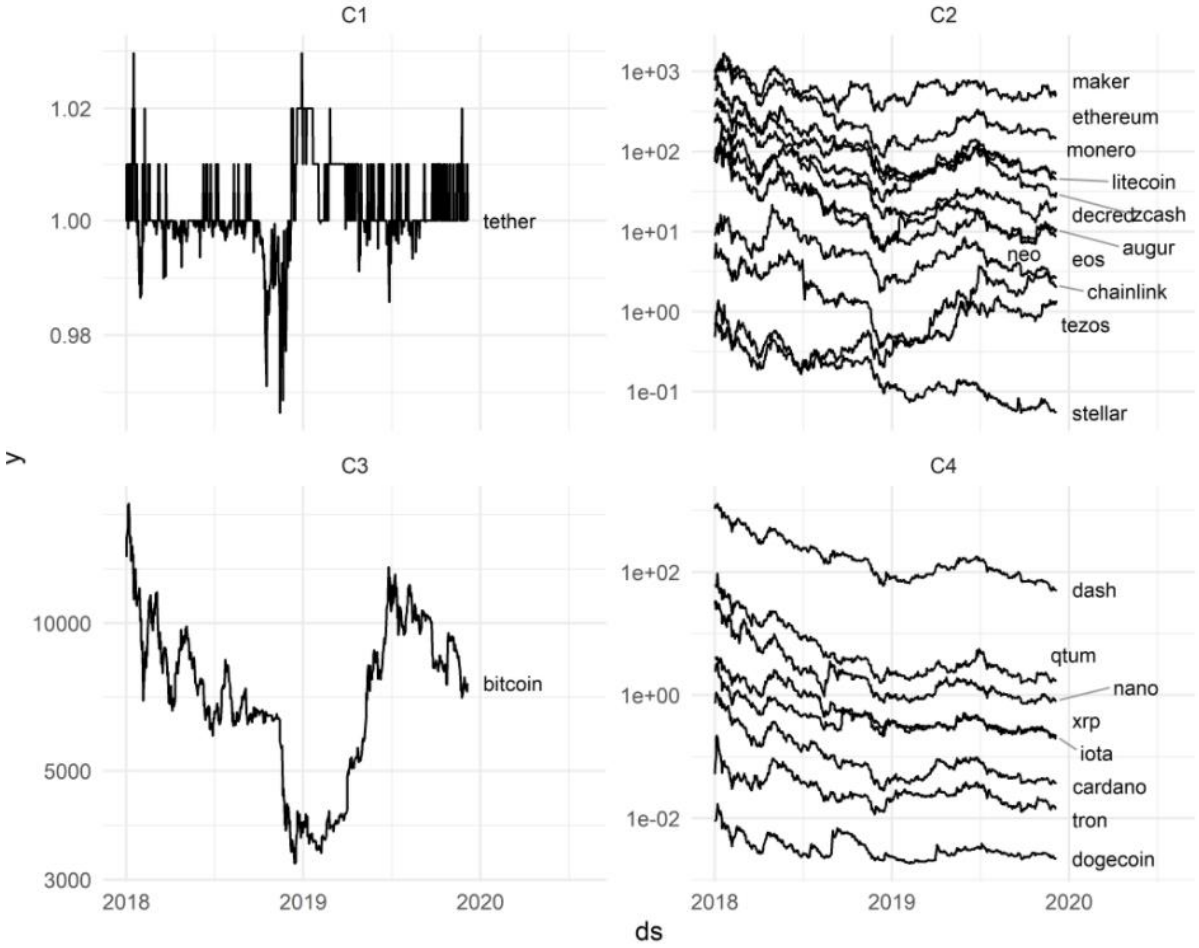

Рисунок 3.10 – Графічне зображення результату кластеризації криптовалют

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований on-line метод кластеризації за непрямими вимірами послідовних даних.

Був проведений аналіз класифікації існуючих підходів для вирішення цієї задачі та основні процеси, що відбуваються при кластеризації. Після проведеного аналізу з'ясувалось, що найбільшими проблемами та особливостями, які впливають на вибір методів кластеризаціїї для часових рядів можна назвати:

– велика розмірність даних;

– можлива зашумленість;

– нестаціонарність;

– необхідність обробки даних у послідовному режимі, тобто за один прохід;

– корельованість даних;

– наявність «прокльону розмірності»;

– різна довжина рядів;

– можливість перетину даних у кластерах.

Також було більш детально розглянуто найбільш використовувані у існуючих програмних реалізаціях по кластеризації часових рядів методи та алгоритми.

На основі проведеного аналізу можна зробити висновок, що не існує універсального методу для вирішення задачі кластеризації часових рядів. Всі існуючі методи мають свої слабкі сторони й вимагають подальшого розвитку, застосування певного методу повинно бути обґрунтовано вирішенням конкретної задачі, результатами та цілями що є в пріоритеті. Наприклад, обробкою стаціонарних часових рядів, або нестаціонарних, природою часових рядів (фінансові послідовності або відео потоки).

Для експериментального дослідження були прийнято рішення використовувати часовий ряд у вигляді зміни ціни різних криптовалют за 3 роки. Цей ряд є доволі гарним: лише декілька криптовалют мають виражений тренд, дані не мають аномальних викидів, часові ряди мають однакову довжину.

В якості алгоритму для обчислення векторів-ознак було обрано однопрохідний метод Велфорда, що є дуже простим та ефектним підходом. Згладжування шумів та вирівнювання тренду було проведено за допомогою експоненційного згладжування. Отримані сформовані та оброблені вектори-ознаки були подані на on-line кластеризацію методом fuzzy *c*-середніх.

Отримані результати кластеризації показують, що така послідовність алгоритму показує високу якість кластеризації даних з певними особливостями. Також треба зазначити, що певний підхід є ефективним у використанні, як у різноманітних фінансових та погодних послідовних даних, так і у кластеризації відеопотоків за рахунок використання непрямих вимірів, що найповніше описують вхідні дані.

Результати роботи апробовано у вигляді тези доповіді під час першої Міжнародної конференції «Major Achievements in Science and Education» у Стокгольмі [40].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Volosyuk Y. (2020) Analysis of clustering algorithm for Data Mining tasks, pp. 112–119.
2. Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer.
3. Birant, D. (2011). *Data mining using RFM analysis*. IntechOpen.
4. Data Mining технології. URL: https://iot.ru/wiki/data–mining (дата звернення: 27.09.2021).
5. Бокс, Д., Дженкинс, Г., & Левшин, А. Л. (1974). Анализ временных рядов: Прогноз и управление. Вып. 2. Мир.
6. Юрченко М. Є. (2019). Прогнозування та аналіз часових рядів.
7. Kay S. M. (1993), Fundamentals of statistical signal processing.
8. Воронцов К. В. (2007), Алгоритми кластеризації і многомірного шкалювання.
9. Esma E. O. (2020), Clustering of Time – series data.
10. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
11. Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38(11), 1857-1874.
12. U – критерій Мана – Уітні. URL: https://ru.wikipedia.org/wiki/U-критерий_Манна_—_Уитни (дата звернення: 30.09.2021).
13. Метод перевірки гіпотези про існування тренду в часовому ряду, заснований на порівнянні середніх рівнів ряду. URL: https://info.wikireading.ru/250993 (дата звернення: 30.09.2021).
14. Кобилін, І. О. (2019). Нечітка кластеризація часових рядів в інтелектуальному аналізі потоків даних.
15. Mantula, E., & Mashtalir, V. (2013, June). An Adaptive Forecasting of Nonlinear Nonstationary Time Series under Short Learning Samples. *In ICTERI* (pp. 91-98).

16. Айвазян, С. А., Енюков, И. С., & Мешалкин, Л. Д. (2019). Прикладная статистика: Основы моделирования и первичная обработка данных. Финансы и статистика.

17. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.

18. Mashtalir, S., & Mashtalir, V. (2020). Spatio-temporal video segmentation. In *Advances in Spatio-Temporal Segmentation of Visual Data* (pp. 161-210). Springer, Cham

19. Kinoshenko, D., Mashtalir, V., & Shlyakhov, V. (2007). A partition metric for clustering features analysis.

20. Kinoshenko, D., Mashtalir, V., Yegorova, E., & Vinarsky, V. (2005, July). Hierarchical partitions for content image retrieval from large-scale database. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition* (pp. 445-455). Springer, Berlin, Heidelberg.

21. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2020). *Advances in Spatio-Temporal Segmentation of Visual Data*. Springer.

22. Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4.

23. Chupikov, A., Kinoshenko, D., Mashtalir, V., & Shcherbinin, K. (2006, July). Image retrieval with segmentation-based query. In *International Workshop on Adaptive Multimedia Retrieval* (pp. 207-221). Springer, Berlin, Heidelberg.

24. Mantula, E., & Mashtalir, V. (2013, June). An Adaptive Forecasting of Nonlinear Nonstationary Time Series under Short Learning Samples. In *ICTERI* (pp. 91-98).

25. Kinoshenko, D., Mashtalir, V., Shlyakhov, V., & Yegorova, E. (2012). nested Partitions Properties for spatial content Image retrieval. In *Multimedia Storage and Retrieval Innovations for Digital Library Systems* (pp. 240-269). IGI Global.

26. Точне обчислення середніх так коваріацій методом Велфорда. URL: https://habr.com/ru/post/333426/#3-vychislenie-srednih (дата звернення: 02.10.2021).

27. Welford's online algorithm. URL: https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance#Online_algorithm (last accessed: 02.10.2021).

28. Mashtalir V., Mashtalir S. (2020). Video shot boundary detection via sequential clustering.

29. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2018, August). Representative based clustering of long multivariate sequences with different lengths. In *2018 IEEE second international conference on Data Stream Mining & Processing (DSMP)* (pp. 545-548). IEEE.

30. Bogucharskiy, S., & Mashtalir, V. (2014). On matrix modification of clarans clustering method in large video surveillance databases.

31. Bogucharskiy, S., Mashtalir, V. (2015). Image segmentation via X-means under overlapping classes.

32. Bobrowski, L., Mashtalir, V., Topczewska, M. (2007) Pattern discovery through separable data projections.

33. Богучарский, С. И., & Машталир, В. П. (2014). Сегментация изображений в больших базах данных с использованием плотности распределения информации. *Электротехнические и компьютерные системы*, (14), 119-123.

34. Mashtalir, V., & Bogucharskiy, S. (2015). Covering Image Segmentation via Matrix X-means and J-means Clustering. *Sensors & Transducers*, 195(12), 56.

35. Mashtalir, V. P., Shlyakhov, V. V., & Yakovlev, S. V. (2014). Group structures on quotient sets in classification problems. *Cybernetics and Systems Analysis*, 50(4), 507-518.

36. Kagramanyan, A., Mashtalir, V., & Shlyakhov, V. (2014). Metrical evaluation of spatial content for segmentationbased image retrieval.

37. Машталир, В. П., Шляхов, В. В., & Яковлев, С. В. (2014). Групповые структуры на фактор-множествах в задачах классификации. *Кибернетика и системный анализ*, (50,№ 4), 27-41.

38. Matplotlib in Python. URL: https://ru.education–wiki.com/8072875–matplotlib–in–python (last accessed: 02.10.2021).

39. Maulik, U., & Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on pattern analysis and machine intelligence,* 24(12), 1650-1654.

40. Богуцька Т. (2021). Дослідження кластеризації послідовностей за непрямими вимірами.