УДК 004.75 doi:10.20998/2413-4295.2024.04.04

ДОСЛІДЖЕННЯ ТА ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

А. М. БОЙКО, О. М. МАРУСЕНКО, В. О. МЄТЄЛЬОВ*, В. В. ОВСЯНІКОВ

Кафедра комп'ютерного моделювання процесів та систем, НТУ «ХПІ», Харків, УКРАЇНА
**e-mail: volodymyr.mietielov@khpi.edu.ua*

АНОТАЦІЯ *Розглядається проблема ефективної обробки часових рядів з метою прогнозування, використовуючи технології розподілених обчислень у хмарному середовищі. Акцентовано увагу на адаптації сучасних підходів до аналізу часових рядів для роботи з великими обсягами даних та їх інтеграції з інфраструктурою хмарних обчислень. Особливий акцент зроблено на обробці ультра-довгих часових рядів, які відзначаються низьким сигнал-шум співвідношенням, складною структурою та довготривалими трендами. Проаналізовано широкий спектр методів прогнозування, включаючи класичні статистичні моделі, такі як авторегресійні інтегровані моделі з рухомим середнім, та сучасні підходи машинного навчання, зокрема нейронні мережі з довгою короткотривалою пам'яттю. Продемонстровано переваги використання паралельних обчислень у забезпеченні значного прискорення обробки великих обсягів даних. Зокрема, у роботі підтверджено ефективність запропонованого підходу з використанням хмарної інфраструктури Amazon Web Services, що дозволяє оптимізувати ресурси та підвищити точність прогнозування. Розроблено програмний пакет на основі технологій Apache Spark для аналізу часових рядів у розподілених середовищах. Проведено тестування продуктивності програмного забезпечення, результати якого свідчать про доцільність його використання у реальних умовах для вирішення задач прогнозування та виявлення аномалій у великих часових рядах. Зокрема, обґрунтовано застосування адаптованої авторегресійної інтегрованої моделі з рухомим середнім у поєднанні з паралельними обчисленнями для забезпечення ефективності прогнозування часових рядів. Розглянуто виклики, пов'язані із впровадженням паралельних обчислень у задачі прогнозування часових рядів, включаючи необхідність оптимізації алгоритмів та забезпечення масштабованості рішень у хмарному середовищі. Окреслено перспективи подальшого вдосконалення програмного забезпечення, зокрема шляхом впровадження адаптивних алгоритмів і розширення їх застосування у сферах кібербезпеки, фінансової аналітики, моніторингу інфраструктурних систем, а також прогнозування в економіці та промисловості. Проаналізовано результати численних обчислювальних експериментів, які довели ефективність розроблених алгоритмів у підвищенні точності прогнозів та зниженні часу обробки даних. Отримані результати формують основу для майбутніх досліджень у напрямі створення комплексних систем аналізу часових рядів, що враховують специфіку різних галузей.*
Ключові слова: *часовий ряд; паралельні обчислення; ARIMA; Apache Spark; AWS EMR; хмарні технології; кластер.*

RESEARCH AND FORECASTING OF TIME SERIES USING PARALLEL COMPUTING TECHNOLOGIES

A. BOIKO, O. MARUSENKO, V. MIETIELOV, V. OVSIANIKOV

Department of Computer Modeling of Processes and Systems, NTU "KhPI", Kharkiv, UKRAINE

ABSTRACT *This study addresses the challenge of efficient time series processing for forecasting purposes using distributed computing technologies in a cloud environment. The focus is placed on adapting modern approaches to time series analysis for handling large data volumes and integrating them with cloud computing infrastructure. Particular attention is given to processing ultra-long time series, characterized by low signal-to-noise ratios, complex structures, and long-term trends. A wide range of forecasting methods is analyzed, including classical statistical models such as autoregressive integrated moving average (ARIMA) and modern machine learning approaches, particularly long short-term memory neural networks. The advantages of parallel computing in significantly accelerating the processing of large data volumes are demonstrated. Specifically, the study confirms the effectiveness of the proposed approach using Amazon Web Services cloud infrastructure, enabling resource optimization and improving forecasting accuracy. A software package based on Apache Spark technologies was developed for time series analysis in distributed environments. Performance testing of the software demonstrated its practical applicability for solving forecasting and anomaly detection tasks in large time series. The application of the adapted autoregressive integrated moving average model, combined with parallel computing, is substantiated as an effective method for time series forecasting. The challenges associated with implementing parallel computing for time series forecasting are explored, including the need for algorithm optimization and ensuring scalability of solutions within a cloud environment. The study outlines prospects for further software enhancements, such as integrating adaptive algorithms and expanding their application to fields like cybersecurity, financial analytics, infrastructure monitoring, and forecasting in economics and industry. The results of extensive computational experiments confirm the effectiveness of the developed algorithms in improving forecast accuracy and reducing data processing time. These findings lay the foundation for future research aimed at creating comprehensive time series analysis systems that account for the specific needs of various industries.*
Keywords*: time series; parallel computing; ARIMA; Apache Spark; AWS EMR; cloud technologies; cluster.*

Вступ

З кожним роком аналіз часових рядів, як одна з форм аналізу даних, знаходить все більше застосувань у найрізноманітніших сферах людської діяльності. Зі зростанням сектору IoT та глобальної інтеграції зростає потреба у засобах для моніторингу

ISSN 2079-5459 (print)
ISSN 2413-4295 (online)

кібернетичних систем, а також у засобах реагування та усунення наслідків збоїв та помилок у них. Розпізнавання вторгнень, недозволеного доступу та дефектів у важливих системах безпеки та управління інфраструктурою є серед ключових пріоритетів у глобальному контексті сучасних інформаційних технологій.

Прогнозування ультра-довгих часових рядів становить суттєвий виклик у сфері аналізу даних, оскільки такі ряди часто мають великі обсяги даних та складну структуру. Застосування паралельних обчислювальних технологій у цьому контексті визначається потребою в ефективності та прискоренні процесу обробки великих обсягів інформації.

Однією з головних проблем у прогнозуванні ультра-довгих часових рядів є обробка великої кількості даних, що може призводити до значного сповільнення традиційних аналітичних методів. Використання паралельних обчислювальних технологій дозволяє розподілити обчислювальні завдання між різними обчислювальними ресурсами, що сприяє оптимізації та прискоренню аналізу.

Додатковою складністю у прогнозуванні ультра-довгих часових рядів є їхній низький сигнал-шум співвідношення та наявність довготривалих тенденцій та циклів. Паралельні обчислювальні технології можуть бути використані для ефективної роботи з такими складними величинами, сприяючи точнішому та швидкому прогнозуванню ультра-довгих часових рядів.

У контексті паралельних обчислень для ультра-довгих часових рядів, необхідно також враховувати високу масштабованість та оптимізацію алгоритмів для ефективного використання ресурсів обчислювальних кластерів чи хмарних середовищ. Такі технології можуть виявитися ключовими для успішного вирішення викликів, пов'язаних із складністю та обсягами ультра-довгих часових рядів.

Мета роботи

Метою даної роботи є підвищення швидкості виявлення аномалій у часових рядах за допомогою алгоритмів паралельних обчислень.

Досягнення мети базується на розробці оригінальних програмного апарату та адаптації алгоритмів паралельних обчислень до методик прогнозування часових рядів. Результатом є також реалізація запропонованого підходу застосування методу паралельних обчислень у вигляді програмного пакету.

Об’єктом даного дослідження є часові ряди з аномальними значеннями тобто тими, що не відповідають природі процесу, що досліджується, та прогнозування появи аномалій у майбутньому, ґрунтуючись на історичній поведінці даного процесу.

Опис роботи додатку

В роботі було взято набір даних про споживання електроенергії в побуті, який є багатовимірним набором даних, що описує споживання електроенергії (з частотою вибірки одна хвилина) в одній сім'ї протягом чотирьох років.

Дані містять 2 075 259 спостережень і 8 ознак (включаючи відмітки часу), зібраних у будинку у Франції з грудня 2006 року по листопад 2010 року.

У цьому завданні метою є прогнозування споживання електроенергії в побуті за часовим рядом, на основі історії споживання протягом 2 мільйонів хвилин даних.

Для статистичної обробки даних часового ряду виконано фільтрацію за допомогою вбудованих бібліотек Apache Spark. На рисунку нижче представлений набір даних після фільтрації, а також його основні агреговані метрики (рис. 1).

	count	mean	std	min	25%	50%	75%	max
global_active_power	2075259.0	1.089418	1.054678	0.076	0.308	0.602	1.526	11.122
global_reactive_power	2075259.0	0.123687	0.112593	0.000	0.048	0.100	0.194	1.390
voltage	2075259.0	240.836427	3.240051	223.200	238.990	241.000	242.870	254.150
global_intensity	2075259.0	4.618401	4.433165	0.200	1.400	2.600	6.400	48.400
sub_metering_1	2075259.0	1.118474	6.141460	0.000	0.000	0.000	0.000	88.000
sub_metering_2	2075259.0	1.291131	5.796922	0.000	0.000	0.000	1.000	80.000
sub_metering_3	2075259.0	6.448635	8.433584	0.000	0.000	1.000	17.000	31.000

```
df.replace('?', 'NaN', inplace=True)
df = df.astype('float')
def fill_missing(values):
    one_day = 60 * 24
    for row in range(values.shape[0]):
        for col in range(values.shape[1]):
            if np.isnan(values[row, col]):
                values[row, col] = values[row - one_day, col]
fill_missing(df.values)
```

Рис. 1 – Статистичний аналіз даних часового ряду

Куртозис нормального розподілу майже дорівнює нулю, і якщо куртозис більший за нуль, розподіл має більш важкі хвости.

З іншого боку, є виміряна асиметрія розподілу. Якщо значення асиметрії знаходиться між -0.5 і 0.5, дані вважаються достатньо симетричними. Однак, якщо значення асиметрії знаходиться між -1 і -0.5 або між 0.5 і 1, дані вважаються помірно асиметричними. Нарешті, якщо значення асиметрії менше -1 або більше 1, дані вважаються сильно асиметричними. У цьому конкретному випадку значення асиметрії більше 1, що свідчить про високу асиметрію розподілу (рис. 2).

```
sns.distplot(data.Global_active_power,color='purple')
print( 'Kurtosis of normal distribution: {}'.format(stats.kurtosis(data.Global_active_power)
))
print( 'Skewness of normal distribution: {}'.format(stats.skew(data.Global_active_power)))
```

```
Kurtosis of normal distribution: 4.218671866132123

Skewness of normal distribution: 1.7862320846320832
```

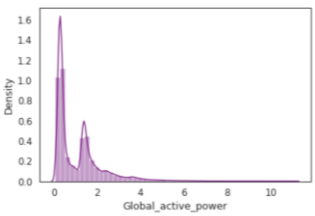

Рис. 2 – Тест на нормальний розподіл даних часового ряду

Для отримання комплексної картини щодо основних метрик набору даних застосуємо операцію групування. Групування даних - це операція, яка найбільш використовуються при аналізі даних. В Pandas за групування даних відповідає метод groupby, який потрібно викликати на об'єкті DataFrame. Також присутня можливість будувати зведені таблиці, аналогічні зведеним таблицям у MS Excel, для цього потрібно використати метод pivot_table.

Для візуалізації даних Pandas використовує бібліотеку Matplotlib. З її допомогою можна з легкістю будувати діаграми. Потрібно лише імпортувати модуль matplotlib.pyplot та викликати метод plot() на об'єкті DataFrame (рис. 3).

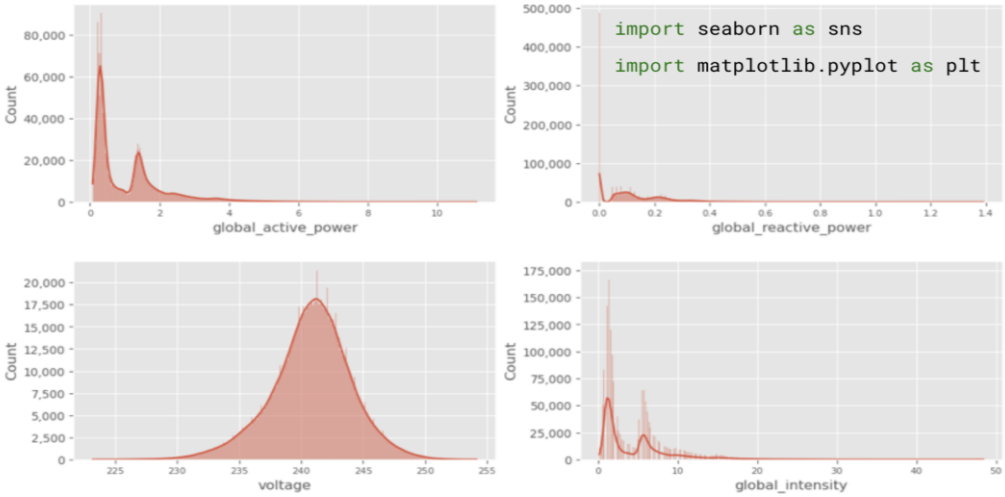

Рис. 3 – Розподіл основних метрик часового споживання електроенергії

Для експериментального дослідження було обрано наступну архітектуру моделі:

1) LSTM з 100 нейронами у першому видимому шарі.

2) Dropout з ймовірністю 20%.

3) 1 нейрон у вихідному шарі для прогнозу Global_active_power.

4) Форма введення буде мати 1 часовий крок з 7 ознаками.

5) Функція втрат середнього абсолютного значення (MAE) та ефективна версія Adam стохастичного градієнтного спуску.

6) Навчання протягом 20 епох тренування з розміром пакету 70.

Для валідації початкових даних проведено тест Дікі-Фуллера. Це статистичний тест, який використовується для перевірки нульової гіпотези про наявність одиничного кореня в часовому ряді, що свідчить про його нестаціонарність та присутність часової залежності. Зворотна гіпотеза - часовий ряд не має одиничного кореня і є стаціонарним, тобто не має часової залежності.

У тесті Дікі-Фуллера, якщо значення p-рівня більше за 0.05, це означає, що ми приймаємо нульову гіпотезу і дані вважаються нестаціонарними. Однак, якщо значення p-рівня менше або дорівнює 0.05, ми відхиляємо нульову гіпотезу і дані вважаються стаціонарними.

Хоча моделі LSTM не вимагають стаціонарності даних, стаціонарний ряд з постійним середнім значенням і дисперсією з часом може забезпечити кращу продуктивність моделі та спростити процес навчання нейронної мережі [1-2].

Нульову гіпотезу, яка вказує на наявність одиничного кореня і, отже, нестаціонарності у часовому ряді, можна відхилити на підставі результатів тесту Дікі-Фуллера. Це означає, що дані не мають часової залежності і є стаціонарними.

Модель вважається ефективною, якщо її продуктивність краща, ніж невдалий підхід, що характеризується середньоквадратичною помилкою прогнозу на рівні близько 465 кіловат для семиденного прогнозу.

Порівняно чотири моделі енкодери яких відповідають за читання і інтерпретацію вхідної послідовності. Вихід енкодера - це вектор фіксованої довжини, який представляє інтерпретацію моделі відносно послідовності. Енкодер є моделлю Vanilla LSTM, а також були використані й інші моделі енкодера, такі як стековані, бідірекційні та CNN-моделі (рис. 4).

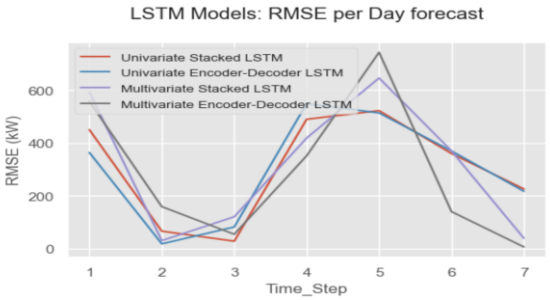

Рис. 4 – Порівняння точності прогнозування по експериментальним результатам застосування інваріантних і багатовимірних моделей

ISSN 2079-5459 (print)
ISSN 2413-4295 (online)

Результати навчання LSTM моделі приведені на наступному рисунку (рис. 5).

Можна дійти до висновку, що за поточних умов (наприклад, кількість кроків введення/виведення, визначення моделі) інваріантні моделі показали трохи кращі результати, ніж багатовимірні моделі.

Науковці часто стикаються з викликом навчання великої кількості моделей за допомогою розподіленого обчислення даних, такого як Apache Spark [3-4]. Використовуючи кластер Spark, окремі робочі вузли кластера можуть паралельно навчати підмножини моделей разом з іншими робочими вузлами, що значно скорочує час, потрібний для навчання всього набору моделей часових рядів.

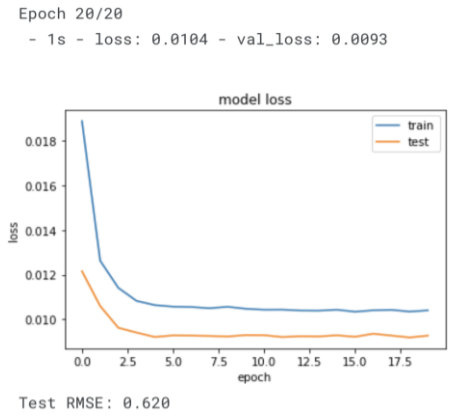

Рис. 5 – Застосування LSTM моделі. Крива навчання

Звичайно, навчання моделей на кластері робочих вузлів (комп'ютерів) вимагає більше хмарної інфраструктури, і це супроводжується витратами. Проте, завдяки легкій доступності хмарних ресурсів на вимогу, компанії можуть швидко виділити необхідні ресурси, навчити свої моделі і так само швидко звільнити ці ресурси, що дозволяє досягти великої масштабованості без довготривалих зобов'язань перед фізичними активами.

Порівняємо швидкість обробки даних за допомогою Apache Spark і Pandas за допомогою двох наборів даних (рис. 6).

Експеримент № 1	
Тип файлів	CSV
Кількість файлів	12
Середній розмір файлу	383,161 MB
Загальний розмір датасету	2.2 GB
Кількість рядків	23,838,931
Pyspark Runtime	0:00:21.042511
Pandas Runtime	0:00:27.491613
Різниця в хвилинах	6.449102
Різниця у відсотках	23%

Експеримент № 2	
Тип файлів	CSV
Кількість файлів	12
Середній розмір файлу	1,282,542 MB
Загальний розмір датасету	7.3 GB
Кількість рядків	84,152,418
Pyspark Runtime	0:00:51.244250
Pandas Runtime	0:01:40.545144
Різниця в хвилинах	49.3008939
Різниця у відсотках	49%

Рис. 6 – Порівняння швидкості обробки даних за допомогою різних програмних пакетів

Основний механізм для досягнення розподіленого обчислення даних в Spark – це DataFrame. Завантажуючи дані в Spark DataFrame, дані розподіляються по робочим вузлам кластера. Це дозволяє цим робочим вузлам обробляти підмножини даних паралельно, скорочуючи загальний час, потрібний для виконання роботи.

Групуючи дані за ключовими значеннями, у цьому випадку за комбінаціями напруга і споживча потужність, ми об'єднуємо всі дані часових рядів для цих ключових значень на конкретному робочому вузлі.

Після правильного групування наших даних часових рядів, нам потрібно навчити одну модель для кожної групи. Для цього ми можемо використовувати функцію користувача (User-Defined Function, UDF) з бібліотеки Pandas, яка дозволяє застосовувати власну функцію до кожної групи даних у нашому DataFrame.

Ця UDF не лише навчатиме модель для кожної групи, але також генеруватиме набір результатів, який представлятиме передбачення цієї моделі. Однак, хоча функція буде навчати і передбачати для кожної групи незалежно від інших, результати, повернені з кожної групи, зручно збираються в один результуючий DataFrame. Це дозволить нам не тільки генерувати прогнози, але представляти наші результати аналітикам і керівникам як єдиний вихідний набір даних.

Для прогнозування споживання електроенергії в майбутньому використовується бібліотека Prophet, що забезпечує створення високоякісних прогнозів, як для експертів, так і для некваліфікованих користувачів.

Типові аспекти "масштабованості", такі як обчислення та зберігання, не викликають особливих проблем у галузі прогнозування. Виявлено, що обчислювальні та інфраструктурні проблеми при прогнозуванні великої кількості часових рядів розв'язуються досить просто - зазвичай процедури прогнозування легко паралелізуються, а прогнози нескладно зберігати у реляційних базах даних, таких як MySQL, або в системах зберігання даних, наприклад, Hive.

Проблеми масштабування, які ми спостерігаємо на практиці, пов'язані зі складністю, що вводиться різноманітністю проблем прогнозування і забезпеченням довіри до великої кількості прогнозів після їх створення.

Багатоагентні моделі моделюють роботу системи різнорідних агентів (одиниць, компаній), що взаємодіють один з одним, і формують цінової процес шляхом зіставлення попиту і пропозиції на ринку. Цей клас включає моделі, засновані на витратах, рівноважні або теоретико-ігрові підходи. Багатоагентні моделі зазвичай зосереджені на якісних питаннях, а не на кількісні результати.

При практичній реалізації фундаментальних моделей виникають дві основні проблеми: доступність даних і включення стохастичних коливань фундаментальних факторів.

Моделі в зменшеній формі (кількісні, стохастичні) характеризують статистичні властивості досліджуваних параметрів в часі з кінцевою метою оцінки похідних інструментів і управління ризиками.

Статистичні (економетричні, технічні аналізи) методи прогнозують поточні показники, використовуючи математичну комбінацію попередніх показників і поточних значень зовнішніх факторів. Дві найбільш важливі категорії – адитивна і мультиплікативна моделі. Перші більш популярні, але обидва тісно пов'язані – мультиплікативна модель для цін може бути перетворена в адитивну модель для логарифмічних цін.

Ключовим моментом в моделюванні і прогнозуванні цін на електроенергію є врахування сезонності. Ціна на електроенергію демонструє сезонність на трьох рівнях: щоденний і щотижневий, а в деякій мірі – річний. При короткостроковому прогнозуванні річна або довгострокова сезонність зазвичай ігнорується, але щоденні і щотижневі моделі (включаючи окрему обробку свят) мають першорядне значення.

Ультра-довгі часові ряди (тобто дані часових рядів, спостережені протягом тривалого часового інтервалу) стають все більш поширеними. Прикладами є годинні потреби в електроенергії, що охоплюють кілька років, індекси акцій, спостережені щохвилини протягом кількох місяців, щоденні максимальні температури, записані протягом сотень років, та поточні потокові дані, що постійно генеруються в режимі реального часу. Спроби прогнозування цих даних відіграють важливу роль у прийнятті рішень з питань інвестицій, організації виробництва, управління сільськогосподарським господарством та ідентифікації бізнес-ризиків. Проте важко працювати з такими довгими часовими рядами за допомогою традиційних методів прогнозування часових рядів.

У роботі визначили три значущі виклики, пов'язані з прогнозуванням ультра-довгих часових рядів. По-перше, оптимізація параметрів при навчанні алгоритмів прогнозування є часовим споживачем через часову залежність самої природи часових рядів. По-друге, обробка часових рядів, що охоплюють такий тривалий інтервал, потребує значних обсягів зберігання, особливо в процесі навчання алгоритмів, що важко виконати на самостійному комп'ютері. Третя і найбільш серйозна складність полягає в тому, що стандартні моделі часових рядів погано працюють для ультра-довгих часових рядів. Однією з можливих причин є те, що зазвичай нереалістично припускати, що процес генерації даних (DGP) часового ряду [5-6] залишався незмінним протягом ультра-довгого часу. Таким чином, існує очевидна різниця між моделями, які ми використовуємо, і фактичним DGP. Більш реалістичним є припущення, що DGP залишається локально стійким для коротких часових параметрів, так званих вікон.

Прогнозисти зробили спроби подолати ці обмеження при прогнозуванні ультра-довгих часових рядів. Простим підходом є відкидання початкових спостережень і використання скороченого часового ряду для підгонки моделі. Але цей підхід працює добре лише для прогнозування кількох майбутніх значень і не є ефективним використанням наявних історичних даних. Кращим підходом є дозволити моделі розвиватися з часом. Наприклад, моделі ARIMA (AutoRegressive Integrated Moving Average) [7-9] та ETS (ExponenTial Smoothing) можуть вирішити цю проблему, дозволяючи змінювати тренд та сезонні компоненти з плином часу [10-11]. Запропонованою альтернативою [12] є застосування прогнозу без моделі з припущенням, що ряд змінюється повільно і плавно з плином часу. Проте вищезазначені методи вимагають значного обчислювального часу для підгонки моделі та оптимізації параметрів, що робить їх менш практично використовуваними в сучасних підприємствах.

У промисловості розподілені обчислювальні платформи зазвичай не мають модулів для прогнозування. Наприклад, відомо, що Spark погано підтримує прогнозування часових рядів, особливо багатошарове прогнозування. Щоб підтримувати прогнозування часових рядів великого масштабу на таких платформах, практики зазвичай змушені використовувати недостатні, але наявні методи на розподілених платформах [13]. Наприклад, вони повинні використовувати моделі регресії в MLlib Spark для реалізації регресії типу авторегресії і штучно перетворювати проблему багатошарового прогнозування в проблему багато підзадач, щоб відповідати платформі Spark для прогнозування часових рядів з покращеною обчислювальною ефективністю.

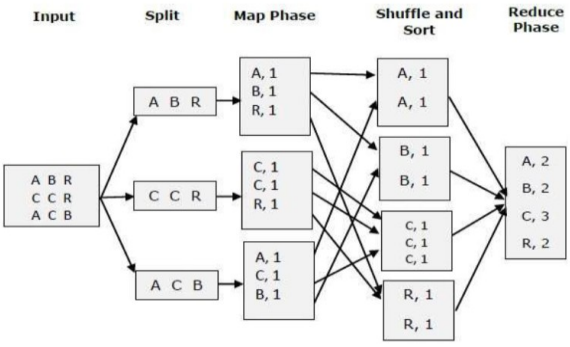

Рис. 7 – Фізичний план роботи MapReduce

У цій роботі вдосконалено ідею "розділяй і володарюй" і пропонується новий підхід до вирішення проблем прогнозування часових рядів великого масштабу в розподілених середовищах. Зокрема, ми пропонуємо розподілений фреймворк для прогнозування часових рядів, в якому довгий часовий ряд спочатку розбивається на декілька підчасових підрядків, охоплюючих короткий період часу, і моделі можуть бути застосовані до кожного підчасового підрядку за розумним припущенням, що DGP залишається незмінним на короткий час (рис. 7).

Запропонований метод зберігає локальну часову залежність і досягає простого розділення зразків, щоб зробити розподілене прогнозування можливим для ультра-довгих часових рядів з одним раундом комунікації. З цієї точки зору наш

ISSN 2079-5459 (print)
ISSN 2413-4295 (online)

фреймворк має риси "моделі змінних коефіцієнтів" для довгого часового ряду. Однак, на відміну від моделей змінних коефіцієнтів, ми поєднуємо локальні оцінювачі, навчені на кожному підчасовому підрядку, використовуючи метод найменших квадратів зі зваженою функцією втрати. Наш фреймворк може бути природно інтегрований в індустріальні розподілені системи з архітектурою MapReduce. Для такого алгоритму MapReduce потрібен лише один раунд комунікації "мастер-робочий" для кожного робочого вузла та уникнення подальших ітераційних кроків. Не потрібна пряма комунікація між робочими вузлами. З цього погляду, це дуже ефективно з точки зору комунікації.

Зазвичай моделі ARIMA входять до числа найбільш використовуваних моделей прогнозування через те, що вони можуть обробляти нестаціонарні та сезонні закономірності, моделі ARIMA часто служать як базові методи завдяки їх чудовим характеристикам. Проте такі моделі важко масштабувати на поточній розподіленій платформі Spark через природу часової залежності, що робить їх непридатними для прогнозування часових рядів великого масштабу.

У реальних застосуваннях даних і симуляціях ми демонструємо, що наш підхід систематично досягає покращення точності прогнозування порівняно з традиційними глобальними моделями часових рядів, як у точкових прогнозах, так і в інтервалах прогнозування. Одержані покращення продуктивності стають більш помітними зі збільшенням горизонту прогнозування. Крім того, наш підхід забезпечує значно покращену обчислювальну ефективність для ультра-довгих часових рядів.

Основною ідеєю є можливість легкого розширення обробки даних на кількох обчислювальних вузлах з високою обчислювальною ефективністю.

Ми ділимо ультра-довгий часовий ряд на кілька підчасових підрядків з реалістичним припущенням про DGP для кожного підчасового підрядку, як це показано на (рис. 8). Таким чином, проблема оцінки параметрів перетворюється на K підзадач та одну комбіновану задачу.

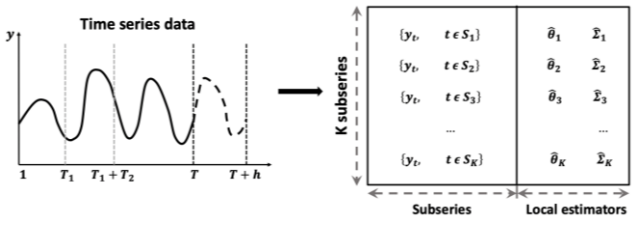

Рис. 8 – Розподілення естіматорів на суб-пакети

Запропонований фреймворк складається з етапів відображених на рисунку (рис. 9).

Крок 1: Попередня обробка. Розділення всього часового ряду на K підчасових підрядків, як показано на рисунку, що виконується автоматично з розподіленими системами.

Крок 2: Моделювання. Навчання моделі для кожного підчасового підрядку через робочі вузли з припущенням, що DGP підчасового підрядку залишається незмінним протягом коротких часових вікон.

Крок 3: Лінійне перетворення. Перетворення навчених моделей на K лінійних представлень.

Крок 4: Комбінування оцінювачів. Об'єднання локальних оцінювачів, отриманих на кроці 3, шляхом мінімізації глобальної функції втрат.

Крок 5: Прогнозування. Прогнозування наступних H спостережень за допомогою об'єднаних оцінювачів.

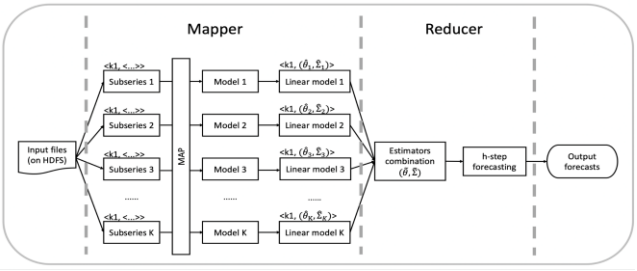

Рис. 9 – Розподілення задачі прогнозування по вузлам Apache Spark

У цій роботі проілюстровано наш підхід за допомогою моделі ARIMA. Оскільки ми розділили часовий ряд інтересу на K підчасових підрядків з послідовними часовими інтервалами, обчислювальна складність моделювання ARIMA для кожного підчасового підрядку зменшується до O(n2T/K), коли порядки моделей вже визначені. В результаті, при прогнозуванні ультра-довгого часового ряду з надзвичайно великим T, наш метод є обчислювально більш ефективним, ніж ARIMA, обчислювальна складність якого становить O(n2T), оскільки він вирішує проблему великомасштабних обчислень у розподіленому вигляді.

Попит на електроенергію може проявляти періодичні закономірності, такі як час доби, день тижня і місяць року. Незважаючи на те, що день тижня може легко включатися до нашої запропонованої моделі як коваріанти, наша передбачувальна аналітика показує відсутність суттєвих відмінностей у закономірностях між днями тижня: включення закономірності дня тижня у моделі ARIMA або розподіленій моделі ARIMA майже не впливає на точність прогнозування як у точкових, так і в інтервальних прогнозах. Крім того, підчасовий підрядок недостатньо довгий для можливості розглядувати місячну сезонність в умовах розподіленого обчислення, тоді як місячну сезонність можна обробляти за допомогою попередніх кроків, таких як розкладання часового ряду. Щоб краще сфокусуватися на оцінці переваг запропонованих розподілених моделей ARIMA над звичайними моделями ARIMA, ми розглядаємо лише вплив часу доби в наступному аналізі, використовуючи сезонні компоненти моделей ARIMA для щотижневих підчасових підрядків (m = 24).

Тестування

Щодо середовища системи, експерименти проводилися на кластері на базі AWS Elastic Map Reduce, оскільки він поєднує у собі дистрибутив Hadoop (з native підтримкою Spark) та систему хмарних обчислень. Першим кроком було створення кластеру. Кожен вузол мав 32 логічних ядра, 64 ГБ оперативної пам'яті і два локальних твердотільних накопичувачі об'ємом 80 ГБ (рис. 10).

Рис. 10 – Налаштування створеного кластеру

У експерименті ми розділяємо кожний часовий ряд на 150 підчасових підрядків з довжиною кожного підчасового підрядка приблизно 800. Для часових рядів такої довжини традиційні моделі прогнозування працюють добре на окремому комп'ютері, і час, витрачений на автоматичний процес моделювання ARIMA, приблизно 5 хвилин, що є емпірично прийнятним.

Далі у роботі досліджується продуктивність запропонованих розподілених моделей ARIMA на наборі даних порівняно з моделями ARIMA і моделями ETS за показниками MASE та MSIS. Час виконання також розглядається як важлива метрика, що описує обчислювальну ефективність алгоритмів. З метою стислості наш алгоритм, розподілена модель ARIMA, надалі буде називатися DARIMA.

У роботі розроблена розподілена модель ARIMA з метою спростити моделювання ARIMA для ультра-довгих часових рядів у розподіленому режимі з високою обчислювальною ефективністю та, можливо, покращеною точністю прогнозування, скоріше ніж проводити змагання між запропонованим методом та іншими методами прогнозування. Тому основним порівнянням, яке нас цікавить, є DARIMA проти ARIMA. Крім того, найбільш поширені методи прогнозування, розроблені з використанням розподілених систем, погано масштабовані для великих горизонтів прогнозування, що робить неможливим застосування цих підходів для прогнозування декількох майбутніх значень. У цьому контексті ми порівнюємо запропонований підхід лише з ARIMA-моделями для всього часового ряду, а також з однією стандартною моделлю для порівняння: моделями ETS (рис. 11).

Argument	ARIMA	DARIMA
`max.p; max.q`	5	5
`max.P; max.Q`	2	2
`max.order`	5	5
`fitting method`	CSS	CSS
`parallel` (multicore)	True	False
`stepwise`	False	True

Рис. 11 – Параметри порівнюваних моделей

Також порівнюється прогностична ефективність DARIMA порівняно з ARIMA і ETS для всього часового ряду за показниками середнього, медіани і стандартного відхилення (SD) значень MASE та MSIS. Очікувано, що DARIMA завжди перевершує бенчмаркові методи незалежно від точкових прогнозів чи інтервалів прогнозів. Зокрема, щодо точкового прогнозування DARIMA досягає значних поліпшень продуктивності порівняно з бенчмарковими методами, приблизно принаймні 9,3% для середнього значення MASE та 8,1% для медіани, із меншим ступенем варіації. Тим часом, DARIMA надає статистично значуще поліпшення (принаймні 23,6%) порівняно з бенчмарковими методами за середнім значенням MSIS.

	MASE			MSIS		
	Mean	Median	SD	Mean	Median	SD
DARIMA	**1.297**	**1.218**	**0.284**	**15.078**	**14.956**	**1.021**
ARIMA	1.430	1.325	0.351	19.733	16.498	7.446
AR representation	1.430	1.325	0.351	19.733	16.498	7.446
ETS	1.491	1.338	0.408	53.783	49.109	15.834

Рис. 12 – Порівняння помилок моделей

На рис. 12 представлені результати MSIS прогнозування з використанням DARIMA, ARIMA та ETS при різних рівнях довіри, які варіюються від 50% до 99%. Ми спостерігаємо, що DARIMA стійко виявляє кращу точність прогнозування, ніж ARIMA та ETS за показником MSIS на різних рівнях довіри. Крім того, перевага DARIMA стає більш суттєвою при збільшенні рівня “довіри”.

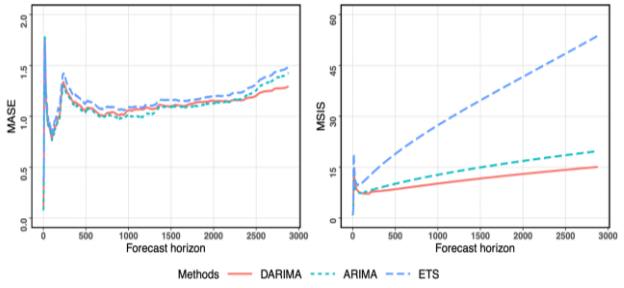

Рис. 13 – Порівняння помилок моделей при збільшенні ряду

ISSN 2079-5459 (print)
ISSN 2413-4295 (online)

Крім того, DARIMA стійко виявляє менший час виконання, ніж ARIMA та ETS при використанні більше ніж двох виконавців/ядер (рис.13). У нашому застосуванні, створення моделі DARIMA для ультра-довгого часового ряду довжиною приблизно 120 000 займає в середньому 1,22 хвилини з 32 ядрами, тоді як моделювання ARIMA займає в середньому 5,16 хвилини, і ETS займає в середньому 5,38 хвилини (рис.14).

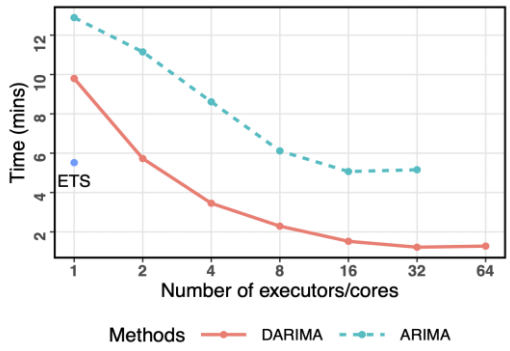

Рис. 14 – Порівняння помилок моделей при збільшенні потужності стенду

Висновки

Створено програмний пакет для аналізу часових рядів за допомогою паралельних обчислень і протестовано його застосування в комп'ютерному кластері хмарного середовища AWS.

Обрано три метрики для оцінки продуктивності нашого запропонованого методу: MASE, MSIS та ACD (абсолютна різниця покриття). Для оцінки інтервалів передбачення ми встановили $\alpha = 0{,}05$ (що відповідає 95% інтервалам передбачення). Як додатковий критерій оцінки, ACD вимірює абсолютну різницю між фактичним покриттям цільового методу та номінальним покриттям, де покриття визначає те, як часто справжні значення знаходяться в інтервалах передбачення, які надає метод.

Відображено точність прогнозування DARIMA, а також трьох методів, розглянутих як бенчмарки в цьому дослідженні. Точність повідомляється окремо для короткострокових (чотири тижні) та довгострокових (решта періодів) горизонтів, а також для всіх горизонтів прогнозування. Крім того, для кожної частоти даних виконується багато порівнянь за кращим, щоб визначити, чи значення середніх рангів для чотирьох розглянутих моделей відрізняються статистично значуще. Якщо інтервали двох методів не перекриваються, це вказує на статистично відмінну продуктивність.

Результати показують, що для щоденних і півгодинних рядів метод DARIMA послідовно досягає найкращої точності прогнозування з точки зору середніх значень MASE та MSIS, особливо для довгострокового прогнозування. Відповідні результати MCB показують, що DARIMA також досягає найвищого рангу продуктивності, за винятком випадку, коли він займає друге місце за MASE для півгодинної частоти, але це не суттєво відрізняється від найкращого.

Список літератури

1. Hou Y. et al. Interpretable CAA Classification Based on Incorporating Feature Channel Attention into LSTM. *Computers & Security*. 2024. P. 104252. doi: 10.1016/j.cose.2024.104252.
2. König T. et al. A LSTM-GAN Algorithm for Synthetic Data Generation of Time Series Data for Condition Monitoring. *Procedia Computer Science*. 2024. Vol. 246, P. 1508–1517. doi: 10.1016/j.procs.2024.09.602.
3. Wang Z. et al. An Empirical Study on the Challenges That Developers Encounter When Developing Apache Spark Applications. *Journal of Systems and Software*. 2022. Vol. 194. P. 111488. doi: 10.1016/j.jss.2022.111488.
4. Reyes-Ortiz J. L., Oneto L., Anguita D. Big Data Analytics in the Cloud: Spark on Hadoop vs MPI/OpenMP on Beowulf. *Procedia Computer Science*. 2015. Vol. 53. P. 121–130. doi: 10.1016/j.procs.2015.07.286.
5. Klopries H., Schwung A. ITF-GAN: Synthetic Time Series Dataset Generation and Manipulation by Interpretable Features. *Knowledge-Based Systems*. 2024. Vol. 283. P. 111131. doi: 10.1016/j.knosys.2023.111131.
6. Dixit A., Jain S. Intuitionistic Fuzzy Time Series Forecasting Method for Non-Stationary Time Series Data with Suitable Number of Clusters and Different Window Size for Fuzzy Rule Generation. *Information Sciences*. 2023. Vol. 623. P. 132–145. doi: 10.1016/j.ins.2022.12.015.
7. Holakouie-Naieni K. et al. Comparative Performance of Hybrid Model Based on Discrete Wavelet Transform and ARIMA Models in Prediction Incidence of COVID-19. *Heliyon*. 2024. Vol. 10, no. 13. P. e33848. doi: 10.1016/j.heliyon.2024.e33848.
8. Singh S., Parmar K. S., Kumar J. Development of Multi-Forecasting Model Using Monte Carlo Simulation Coupled with Wavelet Denoising-ARIMA Model. *Mathematics and Computers in Simulation*. 2024. P. S0378475424004385. doi: 10.1016/j.matcom.2024.10.040.
9. Tosepu R., Ningsi N. Y. Forecasting of Diarrhea Disease Using ARIMA Model in Kendari City, Southeast Sulawesi Province, Indonesia. *Heliyon*. 2024. Vol. 10, no. 22. P. e40247. doi: 10.1016/j.heliyon.2024.e40247.
10. Wang G. et al. Forecasting of Soil Respiration Time Series via Clustered ARIMA. *Computers and Electronics in Agriculture*. 2024. Vol. 225. P. 109315. doi: 10.1016/j.compag.2024.109315.
11. Hyndman R. J., *Athanasopoulos G. Forecasting: Principles and Practice*. Third Print Edition, Melbourne, Australia: Otexts, Online Open-Access Textbooks, 2021.
12. Wang Y., Politis D. N. Model-Free Bootstrap Prediction Regions for Multivariate Time Series. *arXiv*. 2021. doi: 10.48550/ARXIV.2112.08671.
13. Fernández A. M. et al. Automated Deployment of a Spark Cluster with Machine Learning Algorithm Integration. *Big Data Research*. 2020. Vol. 19–20.P. 100135. doi: 10.1016/j.bdr.2020.100135.

References (transliterated)

1. Hou Y. et al. Interpretable CAA Classification Based on Incorporating Feature Channel Attention into LSTM. *Computers & Security*, 2024, p. 104252, doi: 10.1016/j.cose.2024.104252.

2. König T. et al. A LSTM-GAN Algorithm for Synthetic Data Generation of Time Series Data for Condition Monitoring. *Procedia Computer Science*, 2024, vol. 246, pp. 1508–1517, doi: 10.1016/j.procs.2024.09.602.
3. Wang Z. et al. An Empirical Study on the Challenges That Developers Encounter When Developing Apache Spark Applications. *Journal of Systems and Software*, 2022, vol. 194, p. 111488, doi: 10.1016/j.jss.2022.111488.
4. Reyes-Ortiz J. L., Oneto L., Anguita D. Big Data Analytics in the Cloud: Spark on Hadoop vs MPI/OpenMP on Beowulf. *Procedia Computer Science*, 2015, vol. 53, pp. 121–130, doi: 10.1016/j.procs.2015.07.286.
5. Klopries H., Schwung A. ITF-GAN: Synthetic Time Series Dataset Generation and Manipulation by Interpretable Features. *Knowledge-Based Systems*, 2024, vol. 283, p. 111131, doi: 10.1016/j.knosys.2023.111131.
6. Dixit A., Jain S. Intuitionistic Fuzzy Time Series Forecasting Method for Non-Stationary Time Series Data with Suitable Number of Clusters and Different Window Size for Fuzzy Rule Generation. *Information Sciences*, 2023, vol. 623, pp. 132–145, doi: 10.1016/j.ins.2022.12.015.
7. Holakouie-Naieni K. et al. Comparative Performance of Hybrid Model Based on Discrete Wavelet Transform and ARIMA Models in Prediction Incidence of COVID-19. *Heliyon*, 2024, vol. 10, no. 13, p. e33848, doi: 10.1016/j.heliyon.2024.e33848.
8. Singh S., Parmar K. S., Kumar J. Development of Multi-Forecasting Model Using Monte Carlo Simulation Coupled with Wavelet Denoising-ARIMA Model. *Mathematics and Computers in Simulation*, 2024, p. S0378475424004385, doi: 10.1016/j.matcom.2024.10.040.
9. Tosepu R., Ningsi N. Y. Forecasting of Diarrhea Disease Using ARIMA Model in Kendari City, Southeast Sulawesi Province, Indonesia. *Heliyon*, 2024, vol. 10, no. 22, p. e40247, doi: 10.1016/j.heliyon.2024.e40247.
10. Wang G. et al. Forecasting of Soil Respiration Time Series via Clustered ARIMA. *Computers and Electronics in Agriculture*, 2024, vol. 225, p. 109315, doi: 10.1016/j.compag.2024.109315.
11. Hyndman R. J., *Athanasopoulos G. Forecasting: Principles and Practice*. Third Print Edition, Melbourne, Australia. Otexts, Online Open-Access Textbooks, 2021.
12. Wang Y., Politis D. N. Model-Free Bootstrap Prediction Regions for Multivariate Time Series. *arXiv*, 2021, doi: 10.48550/ARXIV.2112.08671.
13. Fernández A. M. et al. Automated Deployment of a Spark Cluster with Machine Learning Algorithm Integration. *Big Data Research*, 2020, vol. 19–20, p. 100135, doi: 10.1016/j.bdr.2020.100135.

Відомості про авторів (About authors)

Бойко Антон Миколайович – студент кафедри комп'ютерного моделювання процесів та систем, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; e-mail: anton.boiko@infiz.khpi.edu.ua.

Boiko Anton – Student of the Department of Computer Modelling of Processes and Systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; e-mail: anton.boiko@infiz.khpi.edu.ua.

Марусенко Олексій Миколайович – асистент кафедри комп'ютерного моделювання процесів та систем, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; ORCID: https://orcid.org/0000-0001-6911-2500; e-mail: Oleksii.Marusenko@khpi.edu.ua.

Marusenko Oleksii – Assistant of the Department of Computer Modelling of Processes and Systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; ORCID: https://orcid.org/0000-0001-6911-2500; e-mail: Oleksii.Marusenko@khpi.edu.ua.

Мєтєльов Володимир Олександрович – кандидат технічних наук, доцент, доцент кафедри комп'ютерного моделювання процесів та систем, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; ORCID: https://orcid.org/0000-0002-2633-6296; e-mail: volodymyr.mietielov@khpi.edu.ua.

Mietielov Volodymyr – Ph. D., Associate Professor, Associate Professor of the Department of Computer Modelling of Processes and Systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; ORCID: https://orcid.org/0000-0002-2633-6296; e-mail: volodymyr.mietielov@khpi.edu.ua.

Овсяніков Владислав Валерійович – аспірант кафедри комп'ютерного моделювання процесів та систем, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна; e-mail: vladyslav.ovsianikov@khpi.edu.ua.

Ovsianikov Vladyslav – Ph.D. student of the Department of Computer Modelling of Processes and Systems, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine; e-mail: vladyslav.ovsianikov@khpi.edu.ua.

Будь ласка, посилайтесь на цю статтю наступним чином:

Бойко А. М., Марусенко О. М., Мєтєльов В. О., Овсяніков В. В. Дослідження та прогнозування часових рядів із застосуванням технологій паралельних обчислень. *Вісник Національного технічного університету «ХПІ». Серія: Нові рішення в сучасних технологіях*. – Харків: НТУ «ХПІ». 2024. № 4 (22). С. 29-37. doi:10.20998/2413-4295.2024.04.04.

Please cite this article as:

Boiko A., Marusenko O., Mietielov V., Ovsianikov V. Research and forecasting of time series using parallel computing technologies. *Bulletin of the National Technical University "KhPI". Series: New solutions in modern technology*. – Kharkiv: NTU "KhPI", 2024, no. 4 (22), pp. 29–37, doi:10.20998/2413-4295.2024.04.04.

Надійшла (received) 10.11.2024
Прийнята (accepted) 16.12.2024