

Міністерство освіти і науки України  
Харківський національний університет імені В. Н. Каразіна  
Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки

«Затверджую»  
Зав. кафедри теоретичної та  
прикладної системотехніки  
\_\_\_\_\_ д.т.н., проф. С. І. Шматков  
«\_\_\_» грудня 2023 р.

## Пояснювальна записка

до кваліфікаційної роботи  
магістра

на тему: «Модель прогнозування навантажень в комп'ютерних мережах за  
допомогою авторегресійних методів»

Захищено на засіданні  
Атестаційної комісії № 40  
протокол № \_\_ від \_\_.12.2023 р.  
Оцінка \_\_\_\_\_ / \_\_\_\_\_  
Голова Атестаційної комісії  
\_\_\_\_\_ **СКОБ Ю. О.**

**Виконав:**  
студент групи КІ– 61  
за спеціальністю 123 – Комп'ютерна  
інженерія.  
Галузь знань: 12 – Інформаційні  
технології  
**КУЛИК Вячеслав Миколайович**

**Керівник:** доцент кафедри, кандидат  
технічних наук  
**БАКУМЕНКО Ніна Станіславівна**

**Рецензент**  
\_\_\_\_\_

## АНОТАЦІЯ

Написанням магістерської кваліфікаційної роботи займався студент групи КІ-61 ФКН ХНУ ім. В.Н. Каразіна Кулик Вячеслав Миколайович.

Робота складається із вступу, трьох розділів, висновків та трьох додатків. Загальний обсяг роботи складає 87 сторінок, 51 з яких це основна частина з 19 рисунками, 7 таблицями і 30 найменуваннями в списку використаних джерел.

**Темою кваліфікаційної роботи** є розробка моделі прогнозування навантажень в комп'ютерних мережах за допомогою авторегресійних методів.

**Об'єкт дослідження** – процес аналізу навантажень в комп'ютерних мережах за допомогою методів моделювання часових рядів.

**Предмет дослідження** – методи аналізу та прогнозування часових рядів, зокрема SARIMA-базованих моделей, які надають можливість врахування тренду та сезонної компоненти для прогнозування навантажень.

**Мета дослідження** полягає в поліпшенні роботи комп'ютерних мереж шляхом прогнозування перенавантажень за допомогою авторегресійних методів.

Новизною даної роботи є використання авторегресійних методів для прогнозування навантажень на даних про кількість користувачів веб-сайту в окремо взятій період часу, який був отриманий із інтегрованого на веб-сайт сервісу Google Analytics. Модель розроблена мовою Python із використанням онлайн сервісу Google Colab.

Робота є актуальною тому, що комп'ютерні мережі зіштовхуються із проблемою недостатньої пропускну здатності в окремі періоди часу. Можливість прогнозування пікових та наднормових навантажень буде сприяти більш стабільній роботі КМ, оскільки об'єкт який приймає рішення

зможе завчасно виділити додаткові ресурси або обмежити доступ до КМ для запобігання перебоїв у системі.

**Ключові слова:** КОМП'ЮТЕРНА МЕРЕЖА, МОНІТОРИНГ, НАВАНТАЖЕННЯ, АВТОРЕГРЕСІЙНІ МЕТОДИ, ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ.

## ABSTRACT

The master's thesis was written by Vyacheslav Kulyk, a student of the group KI-61 of the FCN of V. N. Karazin Kharkiv National University.

The work consists of an introduction, three chapters, conclusions, and three appendices. The total volume of the work is 87 pages, 51 of which are the main parts with 19 figures, 7 tables, and 30 references.

**The topic of the qualification work** is the development of a model for predicting loads in computer networks using autoregressive methods.

**The object of study** is the process of analyzing loads in computer networks using time series modeling methods.

**The subject of the study** is methods of time series analysis and forecasting, in particular SARIMA-based models, which allow taking into account the trend and seasonal component for load forecasting.

**The purpose of the study** is to improve the performance of computer networks by predicting overloads using autoregressive methods.

The novelty of this work is the use of autoregressive methods to predict loads based on data on the number of website users in a given period of time, which was obtained from the Google Analytics service integrated into the website. The model was developed in Python using the Google Colab online service.

The work is relevant because computer networks face the problem of insufficient bandwidth in certain periods of time. The ability to predict peak and overtime loads will contribute to a more stable operation of computer networks since the decision-maker will be able to allocate additional resources in advance or restrict access to KM to prevent system outages.

**Keywords:** COMPUTER NETWORK, MONITORING, LOAD, AUTOREGRESSIVE METHODS, TIME SERIES FORECASTING.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ .....	7
ВСТУП .....	8
<b>РОЗДІЛ 1. МЕТОДИ МОНІТОРИНГУ НАВАНТАЖЕНЬ В КОМП'ЮТЕРНИХ МЕРЕЖАХ.....</b>	<b>10</b>
1.1    Визначення навантаження в комп'ютерних мережах.....	10
1.2    Цілі та завдання моніторингу навантажень .....	14
1.3    Аналіз методів та засобів моніторингу обчислювальних мереж.....	15
1.4    Огляд методів аналізу часових рядів.....	19
1.4.1    Прогнозна екстраполяція.....	20
1.4.2    Інтуїтивні (експертні) методи прогнозування.....	25
1.4.3    Кореляційний та регресійний аналізи .....	26
1.4.4    Моделі нестационарних часових рядів .....	27
1.4.5    Нейронні мережі.....	28
Висновки до 1 розділу .....	30
<b>РОЗДІЛ 2. АНАЛІЗ АВТОРЕГРЕСІЙНИХ МЕТОДІВ .....</b>	<b>32</b>
2.1    Огляд авторегресійних моделей.....	32
2.2    Вибір моделі авторегресії для аналізу часових рядів .....	38
2.3    Алгоритми оцінювання моделі за методологією Бокса-Дженкінса ....	40
2.3.1    Ідентифікація моделі.....	40
2.3.2    Оцінювання параметрів моделі.....	42
2.3.3    Діагностика моделі.....	45
2.3.4    Використання моделі для прогнозування майбутніх значень часового ряду .....	48
Висновки до 2 розділу .....	48

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОЇ МОДЕЛІ ПРОГНОЗУВАННЯ НАВАНТАЖЕНЬ В КОМП'ЮТЕРНИХ МЕРЕЖАХ.....	50
3.1 Програмно-апаратний комплекс .....	50
3.2 Вибір та підготовка набору даних .....	51
3.3 Розробка моделі .....	56
3.4 Покращення моделі .....	60
Висновки до 3 розділу .....	62
ВИСНОВКИ.....	63
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТКИ.....	68

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

КМ – комп'ютерна мережа

МА – мережева архітектура

ISDN – Цифрова мережа з інтегрованими послугами (Integrated Services Digital Network)

ICMP – Міжмережевий протокол керуючих повідомлень (Internet Control Message Protocol)

SNMP – простий протокол керування мережею (Internet Control Message Protocol)

IPv4 – четверта версія мережевого протоколу IP (Internet Protocol version 4)

IPv6 –шоста версія мережевого протоколу IP (Internet Protocol version 6)

MTU –термін, що в комп'ютерних мережах використовується для позначення максимального розміру блоку корисного навантаження пакету (Maximum Transmission Unit)

TCP/IP – набір протоколів мережі Інтернет. Назва походить від назви стрижневих протоколів мережі Інтернет — IP (англ. Internet Protocol – «міжмережевий протокол») і TCP (англ. Transmission Control Protocol – «протокол керування передаванням»)

UDP – протокол передачі датаграм (User Datagram Protocol)

ARIMA – модель авторегресійної інтегрованої ковзної середньої (Autoregressive Integrated Moving Average)

ARMA – модель авторегресії ковзного середнього (Autoregressive Moving-Average)

АКФ – автокореляційна функція

ЧАКФ – частково автокореляційна функція

API – інтерфейс програмування застосунку (Application Programming Interface)

## ВСТУП

У сучасному інформаційному суспільстві, де комп'ютерні мережі відіграють визначальну роль у функціонуванні практично всіх галузей, важливість стабільності цих мереж стає надзвичайно великою. Зростання обсягу даних, постійна зміна робочих завдань та вимог користувачів створюють унікальні виклики для адміністраторів мереж та інженерів з управління ресурсами.

Однією з ключових складових для забезпечення ефективності комп'ютерних мереж є прогнозування навантажень. Критично важливо мати можливість передбачити збільшення або зниження навантажень для запобігання виключним ситуаціям, таким як перевантаження мережі, зниження швидкості передачі даних чи навіть системні збої.

Прогнозування навантажень дозволяє оперативно адаптувати ресурси мережі до умов, що змінюються, і забезпечує безперебійну роботу інформаційно-комунікаційних систем. Відсутність адекватного прогнозування може призвести до недоцільного використання ресурсів, затримок у передачі даних та загрози безпеці інформації.

Дана кваліфікаційна робота присвячена розробці та вдосконаленню моделі прогнозування навантажень в комп'ютерних мережах за допомогою авторегресійних методів, з метою підвищення стійкості та продуктивності інформаційних технологій. Вивчення цього аспекту стає невід'ємною частиною стратегій управління мережами, спрямованих на запобігання перевантажень.

**Мета даної роботи** полягає в поліпшенні роботи комп'ютерних мереж шляхом прогнозування перенавантажень за допомогою авторегресійних методів.

Реалізація поставленої мети вимагає вирішення наступних **завдань**:

– ознайомлення з існуючими типами моніторингу навантажень в КМ;

- проведення детального аналізу методів прогнозування часових рядів;
- виокремлення однієї авторегресійної моделі для прогнозування навантажень;
- обрання набору даних, який стане основою для створення комп'ютерної моделі;
- розробка комп'ютерної моделі прогнозування навантажень в КМ за допомогою авторегресійних методів;
- тестування та покращення існуючої моделі з метою зменшення похибки.

**Об'єктом дослідження** є процес аналізу навантажень в комп'ютерних мережах за допомогою методів моделювання часових рядів.

**Предметом дослідження** є методи аналізу та прогнозування часових рядів, зокрема SARIMA-базованих моделей, які надають можливість врахування тренду та сезонної компоненти для прогнозування навантажень.

**Практична значимість** результатів полягає у використанні розробленої моделі для прогнозування навантажень КМ, зокрема веб-застосунків та веб-сайтів.

**Актуальність** даної роботи полягає у необхідності прогнозування наднормових навантажень, які можуть стати причиною перебоїв у КМ, з метою передчасної адаптації ресурсів.

## РОЗДІЛ 1. МЕТОДИ МОНІТОРИНГУ НАВАНТАЖЕНЬ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

### 1.1 Визначення навантаження в комп'ютерних мережах

Повсюдна комп'ютеризація викликає збільшення кількості провайдерів, які пропонують послуги з підключення високошвидкісного Інтернету. Через високий попит на дані послуги щодня збільшується кількість підключених користувачів. Це створює підвищення навантаження на сервери.

Провайдери високошвидкісного інтернету займаються прогнозуванням завантаженості мереж через те, що це впливає на багато факторів. Своєчасне виявлення потенційної проблеми може дозволити уникнути збоїв у роботі. Фахівці використовують багато різних алгоритмів прогнозування. На результат впливають вхідні дані вибірки, що формуються з часових рядів. При виборі способу для прогнозу навантаження слід враховувати тип часового ряду та його властивості.

Електронний ресурс «Вікіпедія» визначає поняття перевантаження в комп'ютерних мережах і теорії черг як «зниження якості обслуговування, яке виникає, коли мережевий вузол або лінія зв'язку переносить більше даних, ніж може обробляти. Типові ефекти включають затримку в черзі, втрату пакетів або блокування нових з'єднань» [1]. А Жан-Клод Франчітті зазначає, що перевантаження в комп'ютерній мережі «виникає коли збільшення кількості пакетів переданих через мережу наближається до пропускної здатності пакетів мережі» [2]. Можна зробити висновок, що перевантаження з'являється тоді, коли по мережі передається така кількість даних, яка досягає або дорівнює граничному показнику пропускної здатності, а наслідком перевантаження може стати зниження пропускної здатності мережі.

Далі слід визначати поняття комп'ютерної мережі. Хоменко В. Г. в навчальному посібнику сформував визначення комп'ютерної мережі як «програмно-апаратний комплекс, що включає в себе комп'ютери, принтери, комунікаційне устаткування, кабельну систему і мережеві операційні системи,

призначений для передачі інформації, ефективного використання обчислювальних і комунікаційних ресурсів устаткування і програмного забезпечення» [3]. Мінухін С.В, в свою чергу, визначив поняття комп'ютерної мережі як «сукупність комп'ютерів, пристроїв печатки, мережних пристроїв і комп'ютерних програм, зв'язаних між собою кабелями або радіохвилями» [4]. Отже, можна зробити висновок, що будь-яка комп'ютерна мережа складається з деяких вузлів, кожен з яких виконує свої функції і пов'язаний з іншими комунікаційними каналами.

Розглянувши поняття КМ, слід більш детально розібрати сутність мережевої архітектури. Ільчук А.В. виділив наступну трактовку: «це комбінація стандартів, топологій і протоколів, які утворюють працездатну мережу» [5]. Також він зазначив, що «МА характеризує загальну структуру мережі, тобто всі компоненти завдяки яким мережа функціонує як апаратні засоби так і системи ПЗ. Кожна МА має свої характеристики, параметри продуктивності, апаратні та програмні засоби» [5].

За принципом організації обміну даними між абонентами розрізняють види мереж, що перераховані на рис. 1.1.

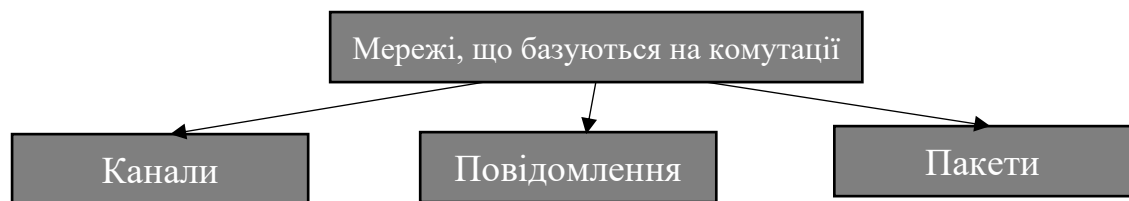


Рис. 1.1 – Види мереж, які базуються на комутації

Комутація каналів (circuit switching) – це спосіб, який може бути використаний для налагодження зв'язку для передачі голосових або звичайних даних між двома віддаленими місцями. Перед початком фактичної передачі інформації необхідно встановити зв'язок шляхом включення комутаторів. Це може бути виконано телефонною службою через набір номера в стандартних голосових лініях або в цифрових каналах ISDN [6].

Гордєєв О.О. виділяє наступну особливість мереж з комутацією каналів: «якщо канал використовується для передачі комп'ютерних даних, то використання таких фіксованих частин пропускної здатності може виявитися неефективним. Наприклад, якщо канал використовується для доступу до Internet, то при передачі веб-сторінки відбувається сплеск активності, після якого настає період бездіяльності каналу поки користувач читає сторінку, а потім новий сплеск при одержанні нової. Такі коливання інтенсивності між нульовою і максимальною типові для потоків даних у комп'ютерних мережах. Оскільки користувач має виключне право на використання такої фіксованої пропускної здатності, то комутовані канали є дорогим способом передачі даних» [6].

Далі слід розглянути наступний тип мереж – мережі із комутацією повідомлень. Спосіб комутації повідомлень передбачає передачу інформації у вигляді логічно завершених порцій даних (таких як телеграми, листи або звіти) між абонентами. У цьому випадку мережа з комутацією повідомлень працює подібно до мережі з комутацією каналів, але фізичні канали зв'язку використовуються лише під час передачі конкретного повідомлення, а не протягом всього сеансу зв'язку. Мережі з комутацією повідомлень були передвісниками мереж з комутацією пакетів, і в даний момент практично не використовуються [6].

Також слід зазначити поняття комутації пакетів, яке Гордєєв О.О. визначив наступним чином: «це технологія доставки повідомлень, при якій дані, розбиті на окремі блоки малих розмірів, що називаються пакетами, можуть пересилатися з вихідного пункту в пункт призначення різними маршрутами. Пакети у пункті призначення потім збираються у початкові дані» [6]. Також, в мережах із комутацією пакетів дані мають проміжне збереження на вузлах комутації. Беручи до уваги той факт, що самі дані мають порівняно малий розмір, «фізичні канали зв'язку виявляються зайнятими протягом мінімальних інтервалів часу, що дозволяє практично завжди забезпечити передачу даних між будь-якими вузлами мережі без тривалих затримок» [6].

Для розкриття поняття моніторингу слід визначити, що таке мережевий трафік. Гордєєв О.О. так охарактеризував дане визначення як інформаційні потоки, що передаються мережею [6].

Одним із найважливіших понять можна визначити моніторинг. Гузій М. М. зазначав дві трактовки твердження моніторинг: «це спеціально організоване, систематичне спостереження за станом об'єктів, явищ, процесів з метою їх оцінки, контролю або прогнозу» та «це систематичний збір і обробка інформації, яка може бути використана для поліпшення процесу ухвалення рішення, а також побічно для інформування громадськості або прямо як інструмент зворотного зв'язку в цілях здійснення проектів, оцінки програм або вироблення політики» [7]. Друге твердження є більш конкретним, адже в ньому Гузій М. М. виділяє деякі функції процесу моніторингу. Даний процес може нести одну або більше функцій, які зазначені на рис. 1.2 [8].

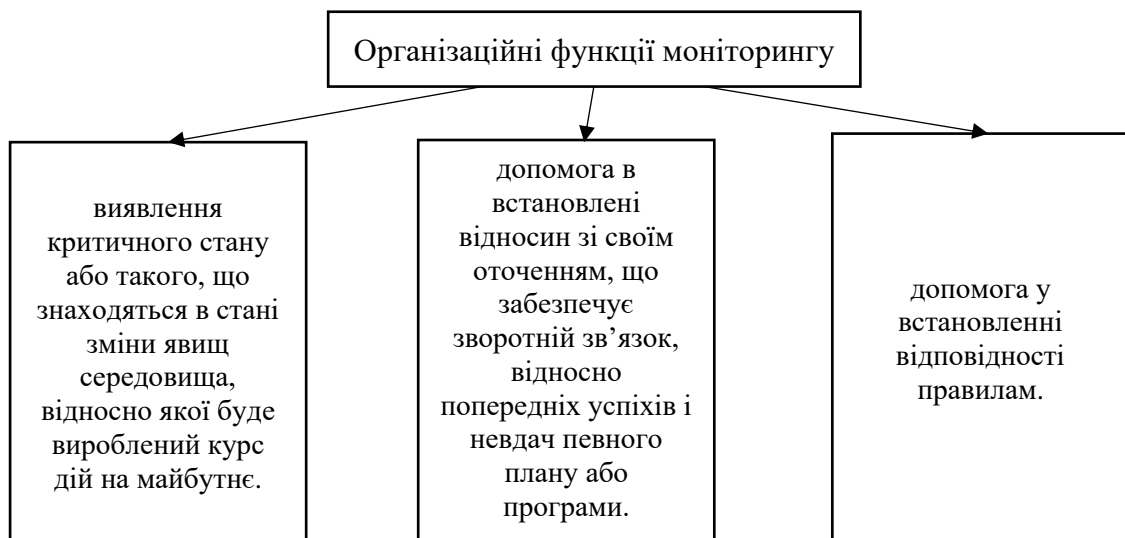


Рис. 1.2 – Організаційні функції моніторингу

Поняття моніторингу буде використовуватися в контексті КМ, тому слід конкретизувати його. Vokogokos L. визначив термін моніторингу КМ таким чином: «це робота системи, яка виконує постійне спостереження за комп'ютерною мережею у пошуках повільних або несправних систем і яка, при виявленні збоїв, повідомляє про них мережевого адміністратора» [9].

Отже, даний процес є одним із найважливіших елементів ефективного та раціонального управління КМ, адже дозволяє оперативно виявляти та реагувати на нетипову поведінку системи.

Vokorokos L. виділив наступну мету моніторингу КМ: «це отримання необхідної інформації про стан КМ, щоб прийняти ефективні рішення в управлінні. При цьому моніторинг у основному полягає у вимірі певних показників КМ, а також виведені агрегованої функції цих вимірювань. Ці показники описують стан та продуктивність мережі з точки зору використання ресурсів, перевантаження, втрати пакетів та допомагають адміністраторам виявити потенційні проблеми» [9].

## **1.2 Цілі та завдання моніторингу навантажень**

Моніторинг у КМ являє собою процес спостереження за роботою системи задля виявлення відхилень від попередньо визначених нормальних показників функціонування задля попередження уповноваженої особи, яка буде приймати рішення по оптимізації процесу роботи КМ та усунення несправностей. Можна визначити дві основні цілі моніторингу КМ, які полягають у виявленні:

- перевантажень;
- аномалій.

Визначення цілей моніторингу КМ важливо для створення майбутньої стратегії дій, тому необхідно спершу виділити для чого потрібен моніторинг, для виявлення перенавантажень чи аномалій.

Вказані нижче завдання моніторингу КМ включають:

- моніторинг завантаження каналів зв'язку;
- виявлення випадків перевантаження;
- виявлення виходу з експлуатації обладнання або програмних сервісів;

- виявлення активності користувачів та програм, а також виявлення можливих атак;
- виявлення слабких місць у мережі та можливих неоптимальностей у конфігурації мережного обладнання;
- визначення параметрів мережевого трафіку;
- прогнозування стану мережі.

Для контролю і моніторингу перенавантажень в КМ адміністратору слід ураховувати всі вищезазначені завдання, адже кожен із аспектів може впливати на роботу системи, її стан в конкретний проміжок часу та прогнозовану поведінку у майбутньому.

### **1.3 Аналіз методів та засобів моніторингу обчислювальних мереж**

Методи моніторингу перевантажень в КМ можна розділити на 2 групи: пасивні та активні. Для їх реалізації можуть бути використані два протоколи ICMP та SNMP.

#### **1.3.1 Огляд протоколів ICMP та SNMP**

Протокол ICMPv4 та ICMPv6 використовується для формування повідомлень про помилки, які виникли під час обробки пакету IPv4 та IPv6 відповідно. Жураковський Б. Ю. визначає завдання протоколу наступним чином: «ICMP дозволяє маршрутизатору або іншому комунікаційному вузлу повідомляти станцію-відправника даних про помилки або нештатні ситуації, що виникли при передаванні IP-пакета, виконувати діагностичні функції (наприклад, утиліта ping), передавати значення MTU, здійснювати пошук сусідів тощо» [10]. Отже, даний протокол дозволяє знаходити виключні ситуації, які можуть статися під час передачі пакетів в КМ. Також, слід перерахувати головні функції ICMP протоколу, які Жураковський Б. Ю. визначив наступним чином:

- «контроль доступності IP-адреси (echo request, echo respond);
  - контроль часу життя пакета в мережі;
  - переадресація пакета;
  - видача повідомлень про недосяжність адресата або про некоректність параметрів;
  - формування та пересилання часових міток;
  - видача запитів та відгуків для адресних масок та іншої інформації»
- [10].

Слід зазначити, що протокол має декілька недоліків, пов'язаних із принципом його роботи. Пакети з ICMP повідомленнями оброблюються як пакети клієнтських сервісних додатків. Тому, пакет з даним повідомленням, як і будь-який інший цього типу, може бути втрачений під час передачі [10].

Протокол SNMP є однією з трьох обов'язкових складових для керування TCP/IP мережами. В даній структурі він виступає протоколом який функціонує безпосередньо між менеджером і елементами, де менеджер – це сама мережа, а елемент – будь-який структурний елемент мережі, що використовує сімейство протоколів TCP/IP. В якості транспортного протоколу, SNMP частіше за все використовує UDP. Через це, взаємодія менеджера і елемента є швидкою, але без гарантії доставки повідомлень [10].

Протокол SNMP визначає п'ять типів повідомлень для обміну між менеджером і клієнтом:

- «get-request (отримати значення однієї або кількох змінних);
- get-next-request (отримати наступну змінну після даної або кілька вказаних змінних);
- set-request (встановити значення однієї або кількох змінних);
- get-response (видати значення однієї або кількох змінних. Це повідомлення повертається агентом менеджеру у відповідь на оператори get-request, get-next-request і set-request);

– trap (повідомити менеджера, якщо щось сталося з агентом).

В табл. 1.1 наведено порівняння протоколів ICMP та SNMP.

Таблиця 1.1

## Порівняльна характеристика протоколів ICMP та SNMP

	ICMP	SNMP
Тип	Протокол мережевого рівня моделі TCP/IP	Протокол прикладного рівня моделі TCP/IP
Призначення	Передачі повідомлень про помилки та інших інформаційних повідомлень, пов'язаних з мережевими операціями	Управління мережевими пристроями та збору інформації про стан мережі
Спосіб взаємодії	Відправка повідомлень про помилки або запитів до вузлів мережі	Відправка запитів та відповідей між менеджером і агентом мережевого пристрою
Безпека	Обмеження інформації в повідомленнях	Доступний механізм автентифікації та авторизації
Застосування	Застосовується в інструментах типу ping та tracerout і обробки повідомлень про помилки мережевого стеку	Застосовується для моніторингу та керування мережевими пристроями з центрального менеджера

### 1.3.2 Характеристика видів моніторингу

При пасивному моніторингу, мережне обладнання чи програми періодично автоматично надсилають дані про свій стан монітору. Збережені дані можуть використовуватися в подальшому, прикладом може бути протокол SNMP. У випадку активного моніторингу, обладнання чи програми реагують на запити, які надходять від адміністратора, прикладом може бути протокол ICMP [11]. Отже, при пасивному моніторингу системний адміністратор зберігає дані, отримані від мережного обладнання, а при активному йому потрібно попередньо надіслати запити, після яких він отримає відповідь і зможе зберегти дані для подальшого аналізу.

У табл. 1.2 [8] зазначені переваги та недоліки моніторингу КМ за допомогою пасивних та активних методів.

Таблиця 1.2

Переваги та недоліки пасивних та активних методів моніторингу трафіку і КМ

Переваги		Недоліки	
Пасивний	Активний	Пасивний	Активний
більша кількість показників створює велику інформативність моніторингу	точний підхід, який спрямований на різні об'єкти	несумісність обладнання зі стандартами SNMP	може створювати велику завантаженість на мережу

Кінець таблиці 1.2

Переваги		Переваги	
Пасивний	Активний	Пасивний	Активний
створює менше навантаження на мережу	можливість виконувати моніторинг без доступу до обладнання	необхідність наявності адміністративного доступу до програм мережевого обладнання для конфігурування параметрів моніторингу	через меншу кількість показників менша інформативність моніторингу
	малі інтервали семплювання	великі інтервали семплювання	брандмауери можуть впливати на процес моніторингу

#### 1.4 Огляд методів аналізу часових рядів

Далі розглядаються методи та поняття аналізу часових рядів, формули для розрахунків. Серед них виділені наступні групи:

- прогнозна екстарполяція;
- інтуїтивні (експертні) методи прогнозування;
- кореляційний та регресивний аналізи;
- моделі нестационарних часових рядів;
- нейронні мережі.

### 1.4.1 Прогнозна екстраполяція

Одним із основних інструментів прогнозування є схема екстраполяції. Є формальна та прогнозна екстраполяція [12].

Коркуна Д.М. визначив, що «формальна екстраполяція ґрунтується на припущенні про збереження в майбутньому реальних тенденцій розвитку об'єкта прогнозу. Під час прогнозних екстраполяції фактичний розвиток пов'язується з гіпотезами про динаміку досліджуваного процесу з урахуванням його фізичної і логічної суті» [12].

В основі екстраполяційних методів прогнозування знаходиться вивчення часових рядів. Часові ряди – це впорядковані в часі набори вимірювань тих чи інших характеристик досліджуваного об'єкта чи процесу [12].

Часовий ряд  $y_t$  може бути представлений у такому вигляді:

$$y_t = x_t + S + C + \varepsilon_t \quad (1.1)$$

Найбільш поширеними методами оцінки параметрів залежностей є метод найменших квадратів та його модифікації, метод експоненціального згладжування, метод імовірнісного моделювання та метод адаптивного згладжування.

#### — Метод найменших квадратів

Метод найменших квадратів (МНК) відзначається винятковим значенням серед методів математичної статистики завдяки широкому спектру його застосувань. Задачею МНК є оцінка закономірностей, які виявляються в умовах випадкових коливань, та його використання для подальших розрахунків, включаючи прогнозування [13], тобто:

$$S = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \rightarrow \min \quad (1.2)$$

де  $\hat{y}_i$  – це розрахункові значення вихідного ряду;

$y_i$  – фактичні значення вихідного ряду;

$n$  – кількість спостережень.

Модель тренду можна подати у вигляді:

$$\hat{y} = f(x_i; a_1, a_2, \dots, a_k, t), \quad (1.3)$$

де  $a_1, a_2, \dots, a_k$  – параметри моделі;

$t$  – час;

$x_i$  – незалежні змінні.

Отже, для знаходження параметрів моделі, задовольняючих умову (1.2), перші похідні величини  $S$  по кожному з коефіцієнтів  $a_j$  необхідно прирівняти до нуля.

МНК використовується для одержання конкретних прогнозів. Він застосовується через свою простоту та легкість реалізації на ЕОМ. Тому даний метод відносять до короткострокового [13].

#### — **Метод експоненціального згладжування**

Експоненціальне згладжування було запропоноване в 1950-х роках Brown R.G., Holt C.C., Winters P.R. Прогнози які створюються за допомогою експоненціальних методів є середньозваженими значеннями минулих спостережень. При цьому, ваги зменшуються експоненціально при переході до старших спостережень Ця структура швидко генерує надійні прогнози для

широкого діапазону часових рядів, що є великою перевагою та має велике значення для застосувань у промисловості [14].

Метод експоненціального згладжування є високоефективним та надійним. Його основні переваги це:

- можливість обліку ваги вихідної інформації,
- простота обчислювальних операцій,
- гнучкість опису різних динаміків процесів.

Євтушенко О.В. зазначає, що метод експоненціального згладжування «дає можливість отримати оцінку параметрів тренду, які позначають не середній рівень процесу, а тенденцію, яка склалася на час останнього спостереження» [15]. Даний метод є одним із найбільш надійних та широко використовується в середньостроковому прогнозуванні. Найскладніший етап експоненціального згладжування це вибір початкових умов, параметра згладжування  $\alpha$  та ступеня прогнозуючого полінома [15].

$$s_t = \begin{cases} c_t & : t = 1 \\ s_{t-1} + \alpha \cdot (c_t - s_{t-1}) & : t > 1 \end{cases} \quad (1.4)$$

де  $S_t$  – згладжений ряд;

$c_t$  – вихідний ряд;

$\alpha$  – коефіцієнт згладжування, який вибирається апріорі, ( $0 < \alpha < 1$ ) [14].

Серед експоненціальних методів згладжування окремої уваги заслуговують метод лінійного тренду Холта та сезонний метод Холта-Вінтерса.

В 1957 році Холт розширив просте експоненціальне згладжування, щоб дозволити прогнозувати дані з тенденцією. Цей метод включає рівняння

прогнозу та два рівняння згладжування (одне для рівня та одне для тенденції), де (1.6) це рівняння прогнозу, (1.7) рівняння рівня, а (1.8) – тренду [14]:

$$\hat{y}_{t+h|t} = \ell_t + hb_t \quad (1.6)$$

$$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \quad (1.7)$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \quad (1.8)$$

Далі, в Холт та Вінтерс розширили метод Холта та додали можливість визначення сезонності. Сезонний метод Холта-Уінтерса складається з рівняння прогнозу та трьох рівнянь згладжування — одне для рівня  $\ell_t$ , одне для тренду  $b_t$ , і одне для сезонної складової  $s_t$  з відповідними параметрами згладжування  $\alpha$ ,  $\beta^*$  і  $\gamma$  [14].

Є два варіанти даного методу, які відрізняються характером сезонної складової: адитивний та мультиплікативний методи.

Адитивний метод є кращим, коли сезонні коливання є приблизно постійними протягом ряду, тоді як мультиплікативний метод є кращим, коли сезонні коливання змінюються пропорційно рівню ряду.

При адитивному методі сезонна складова виражається в абсолютних величинах у масштабі спостережуваного ряду, а в рівнянні рівня ряд сезонно скоригований шляхом віднімання сезонної складової. Протягом кожного року сезонна складова складатиме приблизно нуль.

При мультиплікативному методі сезонна складова виражається у відносних величинах (відсотках), а ряд сезонно скоригований діленням на сезонну складову. У межах кожного року сезонна складова буде складати приблизно  $m$  [14].

Форми компоненту для адитивного (1.9) і мультиплікативного (1.10) методів зазначені нижче:

$$\begin{aligned}
 \hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\
 \ell_t &= \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \\
 b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \\
 s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m},
 \end{aligned} \tag{1.9}$$

$$\begin{aligned}
 \hat{y}_{t+h|t} &= (\ell_t + hb_t)s_{t+h-m(k+1)} \\
 \ell_t &= \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \\
 b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \\
 s_t &= \gamma \frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1 - \gamma)s_{t-m}
 \end{aligned} \tag{1.10}$$

### — Метод ймовірнісного моделювання

Відмінність імовірнісних моделей від екстраполяційних полягає у тому, що основою другого типу є опис зміни у часі процесу. У ймовірнісних моделях оцінюються ймовірності, а не коефіцієнти:

$$y_p = \bar{y} + k_p \sigma_y = \frac{N}{2} + k_p \sqrt{\frac{N}{12}}, \tag{1.11}$$

де  $k_p$  – деякий множник, що враховує число ступенів свободи розподілу.

Дане співвідношення може бути основою оцінок для імовірнісної моделі і при достатній кількості вхідної інформації може дати цілком надійний прогноз.

Цю модель можна назвати простою та наочною тому, що оцінки, отримані з використанням імовірнісного моделювання, мають конкретний сенс. Недоліком можна вважати незнання початкового розподілу та вимогу великої кількості спостережень. Проте, за допомогою визначення процедури початкового розподілу чи з допомогою байесовського методу, коригуючи його, можна розглядати цю модель як ефективний метод прогнозу.

#### **1.4.2 Інтуїтивні (експертні) методи прогнозування**

##### **— Метод інтерв'ю**

Метод інтерв'ю – це співбесіда прогнозіста з експертом, під час якої перший задає питання експерту, слідуючи попередньо розробленому плану. Успіх оцінки залежить від професійних якостей експерта та його вміння швидко знаходити відповіді і робити висновки з різних фундаментальних питань [16]. Буденко С.Ф. зазначає, що «аналітичні експертні оцінки припускають тривалу і ретельну самостійну роботу експерта над аналізом тенденцій, оцінкою стану та шляхів розвитку об'єкта прогнозу. Цей метод дає можливість експертові використовувати всю необхідну йому інформацію про об'єкт прогнозу. Свої міркування експерт оформляє у вигляді доповідної записки» [16].

##### **— Метод Дельфі**

Вважається, що метод Дельфі використовує замість колективного обговорення тієї чи іншої проблеми індивідуальне опитування експертів. Воно проводиться у формі анкет і дозволяє з'ясувати відносну важливість та терміни здійснення гіпотетичних подій. Далі анкети статистично оброблюються, із результатів формується колективна думка групи, знаходяться та узагальнюються аргументи, які були приведені на користь того чи іншого судження. Уся сформована та зібрана інформація доноситься до експертів. Далі учасники переглядають оцінки і пояснюють чому вони незгодні з колективним судженням. Це повторюється 3-4 рази і призводить до звуження

діапазону оцінок. Єдиний недолік полягає в тому, що неможливо оцінити вплив, який здійснювався на експертів під час опитування, тому цей метод може бути необ'єктивним [17].

### 1.4.3 Кореляційний та регресійний аналізи

Дослідники часто стикаються з необхідністю встановлення певної гіпотези про характер зв'язку у вигляді функціональної залежності, яка може бути апроксимована деяким відносно простим математичним виразом, таким як лінійне рівняння чи многочлен. Для виявлення таких математичних функціональних або структурних залежностей між двома або більше змінними (на основі накопичених експериментальних даних) дуже корисні методи кореляційного та регресійного аналізу [18]. Для лінійного випадку була отримана модель множинної регресії, яка в наступному вигляді:

$$y_j = \sum_{i=1}^n \alpha_i x_{ij} + \varepsilon_j \quad (1.12)$$

де  $\alpha$  – коефіцієнти моделі;

$y_j$ ,  $x_{ij}$ , відповідно значення  $j$ -ї функції (залежної змінної) та  $i$ -ї незалежної змінної;

$$i = \overline{0, n};$$

$$j = 1, \bar{N}, \varepsilon_j - \text{випадкова помилка};$$

$n$  – число незалежних змінних у моделі (у ряді випадків вважається, що  $\alpha_i$  вільний член  $x_{0j} = 1$ ).

#### 1.4.4 Моделі нестационарних часових рядів

##### — Модель авторегресії-проінтегрованого ковзного середнього (ARIMA(p, k, q)-модель)

Дана модель була запропонована Дж. Боксом та Г. Дженкінсом. Ця методика призначена для аналізу нестационарних часових рядів  $x_t$ , що мають такі характеристики:

- аналізований тимчасовий ряд включає адитивну складову  $f(t)$ , яка представляє собою алгебраїчний поліном (залежний від часового параметра часу  $t$ ) з деяким коефіцієнтом  $k > 1$ ; коефіцієнти цього полінома можуть бути як стохастичної, так і нестохастичної природи.

- ряд  $x_t^k, t = 1, \dots, T - k$ , отриманий з  $x_t$  після застосування до нього  $k$ -кратної процедури методу послідовних різниць, може бути описаний моделлю ARMA(p, q).

На основі вищезазначеної інформації можна записати ARIMA(p, k, q)-модель аналізованого процесу  $x_t$  у наступному вигляді:

$$x_t^k = \alpha_1 x_{t-1}^k + \alpha_2 x_{t-2}^k + \dots + \alpha_p x_{t-p}^k + \delta - \theta_1 \delta_{t-1} - \dots - \theta_q \delta_{t-q}, \quad (1.13)$$

$$x_t^k = \Delta^k x_t = x_t - C_k^1 x_{t-1} + C_k^2 x_{t-2} - \dots + (-1)^k x_{t-k}. \quad (1.14)$$

Можна зауважити, що класу моделей ARIMA належить і найпростіша модель стохастичного тренду – процес випадкового блукання (або просто випадкове блукання). Випадкове блукання визначається аналогічно до процесу авторегресії першого порядку, проте тільки у випадкового блукання  $\alpha = 1$ :

$$\varepsilon_t = \varepsilon_{t-1} + \delta_t. \quad (1.15)$$

— **Моделі рядів, що містять сезонну компоненту**

Юрченко М. Є. визначив, що «тренд-сезонні часові ряди окрім тренду містять чітко виражені сезонні коливання, які, своєю чергою, спричинюють нестационарність. Якщо процес включає періодичні (сезонні) коливання навколо середнього значення з періодом  $m$ , тобто:

$$y_{t+m} \approx y_t \quad (2.3)$$

із точністю до випадкової складової, то в цьому разі різниці через  $m$  часових інтервалів представляють стаціонарний процес:

$$\Delta_m y_t = y_{t+m} - y_t, \quad t = 0, 1, 2, \dots \text{ де } m \text{ — const,} \quad (2.4)$$

середнє значення якого збігається із середнім значенням початкового ряду» [19].

Слід зазначити, що амплітуда сезонних коливань може змінюватися не лінійно в часі. Подібні моделі часових рядів мають тренд в дисперсії та середньому значенні [19].

#### **1.4.5 Нейронні мережі**

Штучні нейронні мережі – це методи прогнозування, які базуються на простих математичних моделях мозку. Вони дозволяють створювати складні нелінійні зв'язки між змінною відповіді та її предикторами [14].

Нейронну мережу можна розглядати як мережу «нейронів», які організовані в шари. Прогнози (або входи) утворюють нижній рівень, а

прогнози (або виходи) утворюють верхній рівень. Також можуть бути проміжні шари, що містять «приховані нейрони» [14].

Найпростіша нейронна мережа з чотирма предикатами, яка не містить прихованих шарів, є еквівалентом лінійній регресії зображена на рис. 1.3:

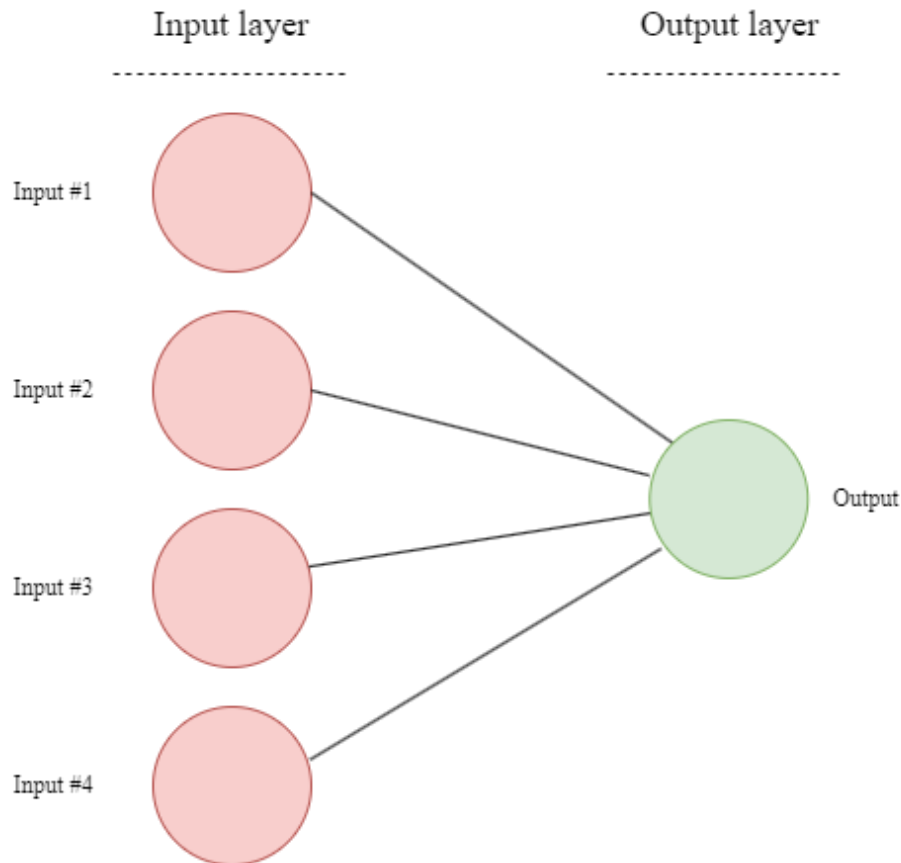


Рис. 1.3 – Нейронна мережа без прихованого шару

Коефіцієнти, додані до цих предикторів, називаються «вагами». Прогнози отримують шляхом лінійної комбінації вхідних даних. Вагові коефіцієнти вибираються в структурі нейронної мережі за допомогою «алгоритму навчання», який мінімізує «функцію вартості», таку як середньоквадратична похибка [14].

Як тільки ми додаємо проміжний шар із прихованими нейронами, нейронна мережа стає нелінійною, як зображену на рис. 1.4:

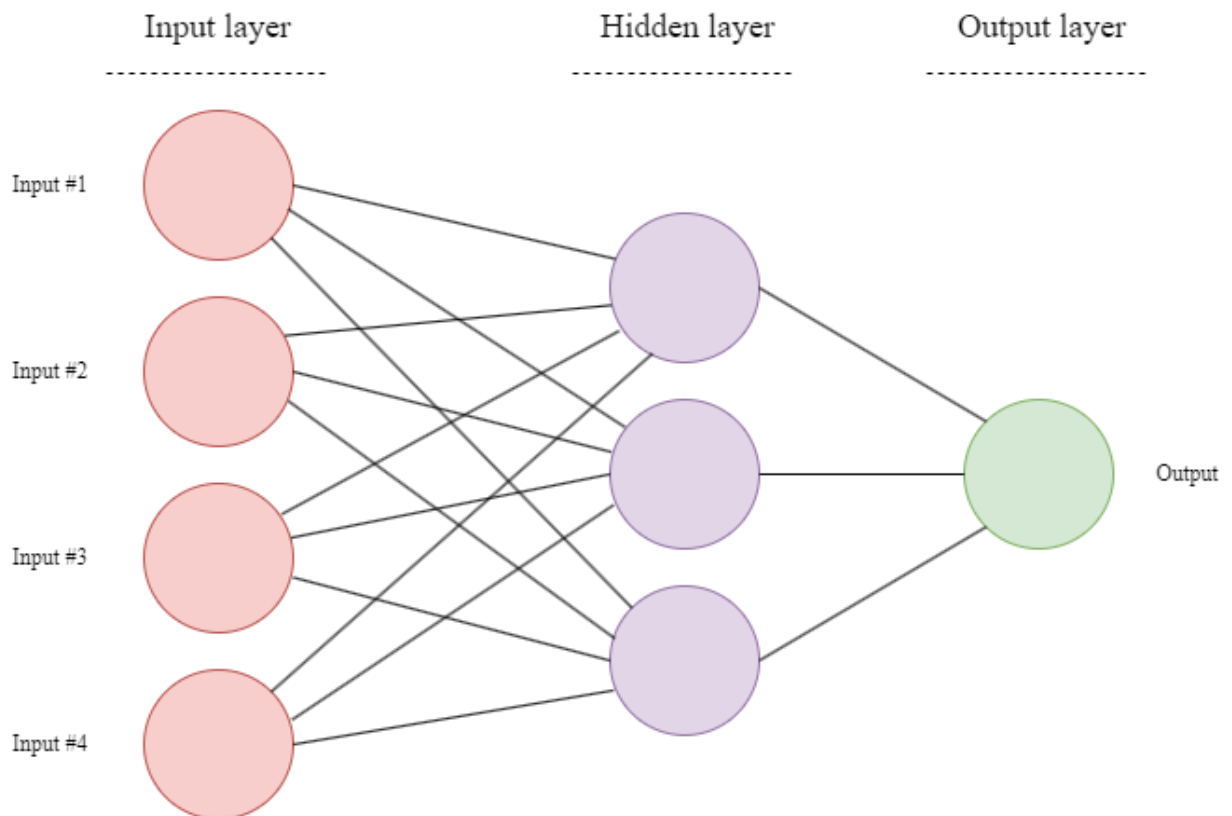


Рис 1.4 – Нейронна мережа із прихованим шаром

Використання штучних нейронних мереж для аналізу часових рядів спирається виключно на дані, які спостерігалися. Оскільки багатошарові нейронні мережі прямого поширення з принаймні одним прихованим шаром і достатньою кількістю прихованих одиниць здатні апроксимувати будь-яку вимірювану функцію, штучна нейронна мережа достатньо потужна, щоб представити будь-яку форму часових рядів [20].

Здатність до узагальнення дозволяє штучним нейронним мережам навчатися навіть у разі шуму та/або відсутності даних. Ще однією перевагою перед лінійними моделями, такими як техніка ARIMA, є здатність мережі представляти нелінійні часові ряди [20].

### Висновки до 1 розділу

У розділі 1 було проаналізовано наявні методи моніторингу комп'ютерних мереж, а саме активні та пасивні. При пасивному моніторингу,

мережне обладнання чи програми періодично автоматично надсилають дані про свій стан монітору. У випадку активного моніторингу, обладнання чи програми реагують на запити, які надходять від адміністратора.

Також, було обґрунтовано застосування методів аналізу часових рядів для прогнозування навантажень. Розглянуті недоліки та переваги різних методів аналізу часових рядів, серед яких були зазначені наступні:

- прогнозна екстарполяція;
- інтуїтивні (експертні) методи прогнозування;
- кореляційний та регресивний аналізи;
- моделі нестационарних часових рядів;
- нейронні мережі.

Для розв'язку задачі прогнозування навантажень в комп'ютерних мережах запропоновано обрати моделі нестационарних часових рядів, оскільки вони спроможні працювати з нестационарними рядами.

## РОЗДІЛ 2. АНАЛІЗ АВТОРЕГРЕСІЙНИХ МЕТОДІВ

### 2.1 Огляд авторегресійних моделей

#### 2.1.1 Прогнозування на базі ARIMA-моделей

В 1976 році Дженкінс і Бокс створили систематичний підхід для побудови моделі ARIMA. Дана методологія включає перераховані нижче етапи для побудови моделі ARIMA для тимчасового ряду, які виділила Березька К.М.:

- «ідентифікація моделі;
- оцінювання моделі і перевірка її адекватності;
- прогнозування» [18].

Chatfield С. зазначив наступний характерний запис моделі ARIMA( $p, d, q$ ):

$$(\Delta^d X_t) = \sum_{i=1}^p \varphi_i (\Delta^d X_{t-i}) + \varepsilon_t + \sum_{j=1}^q \theta_j (\Delta^d \varepsilon_{t-j}), \varepsilon_t \sim N(0, \sigma_t^2) \quad (2.1)$$

або більш короткий запис:

$$\varphi(B)(1 - B)^d X_t = \theta(B)Z_t \quad (2.2)$$

де  $\varphi(\bullet), \theta(\bullet)$  є поліномами степеню  $p$  та  $q$ ;

$B$  – лаговий оператор;

$$(B^j X_t = X_{t-j}, B^j \varepsilon_{t-j}, j = 0, \pm 1, \dots),$$

$d$  – порядок взяття послідовної різниці ( $\Delta X_t = X_{t-1} - X_t = (1 - B)X_t, \Delta^2 X_t = \Delta^2 X_{t+1} - \Delta X_t = (1 - B)^2 X_t, \dots$ ) [16].

### 2.1.2 Моделі ковзного середнього порядку $q$ (MA( $q$ )-моделі)

Спочатку треба розглянути окремий випадок загального лінійного процесу, при якому тільки перші  $q$  з вагових коефіцієнтів  $\beta_j$  ненульові. В такій ситуації процес може бути записаний в наступному вигляді:

$$\varepsilon_t = \delta_t - \theta_1 \delta_{t-1} - \theta_2 \delta_{t-2} - \dots - \theta_q \delta_{t-q}, \quad (2.3)$$

де символи  $\theta_1, \dots, \theta_q$  використовуються для позначення кінцевого набору параметрів  $\beta$ .

Процес (2.3) називається моделлю ковзного середнього порядку  $q$  (MA( $q$ )).

Також дане співвідношення (2.3) можна записати у наступному вигляді:

$$\delta_t = \varepsilon_t + \theta_1 \delta_{t-1} + \theta_2 \delta_{t-2} + \dots + \theta_q \delta_{t-q}. \quad (2.4)$$

Звідси:

$$\delta_t = \varepsilon_t - \pi_1 \varepsilon_{t-1} - \pi_2 \varepsilon_{t-2} - \dots, \quad (2.5)$$

де коефіцієнти  $\pi_j$  ( $j = 1, 2, \dots$ ) певним чином виражаються через параметри  $\theta_1, \dots, \theta_q$ .

Співвідношення (2.5) можна переписати як модель авторегресії нескінченного порядку:

$$\varepsilon_t = \sum_{j=1}^{\infty} \pi_j \varepsilon_{t-j} + \delta_t. \quad (2.6)$$

Умова оборотності MA(q)-моделі (тобто умова збіжності ряду  $\sum_{j=1}^{\infty} \pi_j$ ) формулюється в термінах характеристичного рівняння моделі (2.6) в наступний спосіб. Усі корені характеристичного рівняння  $1 - \theta_1 z - \theta_2 z^2 - \dots - \theta_q z^q = 0$  повинні лежати поза одиничним колом, тобто  $|z_j| > 1$  для всіх  $j = 1, 2, \dots, q$  [21].

Далі треба встановити взаємозв'язок між процесами AR(q) та MA(q). Для цього можна створити ряд зауважень про взаємозв'язки між процесами авторегресії та ковзного середнього.

Відповідно до визначення для кінцевого процесу авторегресії порядку  $p$   $\delta_t$  може бути виражено як кінцеву зважену суму попередніх  $\varepsilon$ , або  $\varepsilon_t$  може бути представлено як нескінченну суму попередніх  $\delta$ . У той же час, в кінцевому процесі ковзного середнього порядку  $q$ ,  $\varepsilon_t$  може бути виражено як кінцеву зважену суму попередніх  $\delta$  або  $\delta_t$  – як нескінченну зважену суму попередніх  $\varepsilon$ .

Отже, параметри процесу авторегресії кінцевого порядку не повинні задовольняти якісь умови, аби процес був стаціонарним. Для того, щоб процес MA був оборотним, коріння його характеристичного рівняння повинні лежати поза одиничним колом.

### 2.1.3 Моделі авторегресії порядку $p$ (AR(p)-моделі)

Для більш комплексного представлення для початку слід розглянути найпростіші окремі випадки. Наприклад, модель авторегресії 1-го порядку – AR(1) або як її ще називають – марківський процес. Дана модель є найпростішим варіантом авторегресійного процесу коли всі коефіцієнти крім першого дорівнюють нулю. Запишемо наступним виразом:

$$\varepsilon_t = \alpha\varepsilon_{t-1} + \delta_t \quad (2.7)$$

де  $\alpha$  – деякий числовий коефіцієнт, що не перевищує абсолютну величину одиниці ( $|\alpha| < 1$ );

При цьому  $\varepsilon_t$  залежить від  $\delta_t$  і всіх попередніх  $\delta$ , але не залежить від майбутніх значень  $\delta$ ,  $\delta_t$ , не залежить від  $\varepsilon_{t-1}$  та більш ранніх значень  $\varepsilon$ . Через вищезазначені факти,  $\delta_t$  називають інновацією (оновленням).

Далі буду проаналізовані моделі авторегресії 2-го порядку – AR(2), які також іменуються «процесами Юла». Дана модель відноситься до окремих випадків авторегресійного процесу, в якому всі коефіцієнти  $\pi_j$ , окрім перших двох, у правій частині дорівнюють нулю. Даний процес можна записати в наступному вигляді [17]:

$$\varepsilon_t = \alpha_1\varepsilon_{t-1} + \alpha_2\varepsilon_{t-2} + \delta_t, \quad (2.8)$$

де послідовність  $\delta_1, \delta_2, \dots$  утворює білий шум.

Необхідні та достатні умови стаціонарності ряду визначаються як [13]:

$$\begin{cases} |\alpha_1| < 2, \\ |\alpha_2| < 1 - |\alpha_1|. \end{cases} \quad (2.9)$$

У межах загальної теорії моделей, ті ж самі умови стаціонарності впливають з вимоги, щоб всі корені відповідного характеристичного рівняння знаходились поза одиничним колом.

Характеристичне рівняння моделі авторегресії 2-го порядку:

$$1 - \alpha_1 z - \alpha_2 z^2 = 0 \quad (2.10)$$

Для того, щоб підрахувати автокореляційну функцію процесу Юла треба визначити два перші значення  $r(1)$  та  $r(2)$  за допомогою наступних співвідношень:

$$r(1) = \frac{\alpha_1}{1 - \alpha_2}, \quad (2.11)$$

$$r(2) = \alpha_2 + \frac{\alpha_1^2}{1 - \alpha_2}, \quad (2.12)$$

Значення для  $r(\tau)$ ,  $\tau = 3, 4, \dots$  обчислюються за допомогою наступного рекурентного співвідношення:

$$r(\tau) = \alpha_1 r(\tau - 1) + \alpha_2 r(\tau - 2). \quad (2.13)$$

Моделі авторегресії  $p$ -го порядку – AR( $p$ ) ( $p \geq 3$ ) утворюють підмножини у класі загальних лінійних моделей і включають в себе досить велику кількість різних моделей. Якщо в загальній лінійній моделі вважати всі параметри  $\pi_j$ , крім перших  $p$  коефіцієнтів, рівними нулю, ми приходимо до визначення AR( $p$ )-моделі:

$$\varepsilon_t = \sum_{j=1}^p \alpha_j \varepsilon_{t-j} + \delta_t, \quad (2.14)$$

де послідовність випадкових величин  $\delta_1, \delta_2, \dots$  утворює білий шум.

Умови стаціонарності процесу, що генерується моделлю (2.3), також сформульовані в термінах коріння його характеристичного рівняння:

$$1 - \alpha_1 z - \alpha_2 z^2 - \dots - \alpha_p z^p = 0. \quad (2.15)$$

Для стаціонарності процесу необхідно і достатньо, щоб усі корені рівняння лежали поза одиничним колом, тобто перевершували б за модулем одиницю [17].

#### **2.1.4 Авторегресійні моделі зі ковзними середніми в залишках (ARMA(p, q)-моделі)**

Моделі типу ARMA(p, q) можна охарактеризувати як уявлення процесу типу MA як процесу авторегресії неекономічне з погляду його параметризації. Аналогічно, процес AR може бути представлений з використанням моделі ковзного середнього. Тому, для отримання параметризації, іноді доцільно включати в модель як члени, що описують авторегресію, так і члени, які моделюють залишок у вигляді ковзного середнього. Такі лінійні процеси записуються наступним чином:

$$\varepsilon_t = \alpha_1 \varepsilon_{t-1} + \dots + \alpha_p \varepsilon_{t-p} + \delta_t - \theta_1 \delta_{t-1} - \dots - \theta_q \delta_{t-q} \quad (2.16)$$

та називаються процесами авторегресії – ковзного середнього порядку (p, q)(ARMA(p, q)). Далі буде розглянута стаціонарність та оборотність ARMA(p, q)-процесів [12]. Процес (1.29) можна також записати як:

$$\varepsilon_t = \sum_{j=1}^p \alpha_j \varepsilon_{t-j} + \bar{\delta}_{qt}, \quad (2.17)$$

де  $\bar{\delta}_{qt} = \delta_t - \theta_1 \delta_{t-1} - \dots - \theta_q \delta_{q-1}$ .

Аналіз стаціонарності можна провести за тією ж схемою, що й для AR(p)-процесів. При цьому відмінність "залишків"  $\bar{\delta}_{qt}$  та  $\delta_e$  ніяк не вплине на висновки, що визначають умови стаціонарності процесу. Таким чином (2.5) є стаціонарним процесом лише тоді, коли всі корені характеристичного рівняння AR(p)-процесу лежить поза одиничним колом.

## 2.2 Вибір моделі авторегресії для аналізу часових рядів

В попередньому розділі була наведена загальна характеристика методів аналізу часових рядів. Попередній вибір пав на авторегресійні моделі, через низку їх переваг описаних нижче.

- В основі AR-моделей лежить ідея залежності поточного значення від попередніх значень часового ряду.
- AR-моделі можуть бути адаптовані до різних порядків, тобто кількості попередніх значень, які використовуються для прогнозу. Це дозволяє оптимально підібрати модель до конкретного часового ряду.
- AR-моделі можуть виявляти та використовувати локальні шаблони у часових рядах.
- AR-моделі можуть бути розширені до, наприклад, ARIMA-моделей при необхідності врахування інших факторів, як тренд, сезонність, тощо.

Останній фактор може бути вирішальним, оскільки в залежності, від продукту чи послуги, для комп'ютерної мережі якого застосовується модель, сезонність може вносити додаткові складності під час прогнозування навантажень. Тому, вирішальний вибір пав на SARIMA-базовані моделі.

SARIMA (Seasonal Autoregressive Integrated Moving Average) є розширенням моделі ARIMA для прогнозування часових рядів з сезонним компонентом. Вона враховує не тільки залежності в часовому ряді, але й сезонні залежності, що спостерігаються на певному періоді часу (наприклад, щоденні, щотижневі, щомісячні сезони).

Основна відмінність між ARIMA та SARIMA полягає в тому, що SARIMA включає додаткові сезонні компоненти для моделювання сезонних залежностей в часових рядах. В SARIMA моделі враховуються не тільки попередні значення в ряді, але й попередні значення на певному сезонному інтервалі.

SARIMA створює модель сезонного часового ряду шляхом об'єднання моделі ARIMA та стохастичної сезонної моделі, яка скорочено називається SARIMA (p, d, q) (P, D, Q)S, де:

- p і q є порядком авторегресії та ковзного середнього,
- P і Q – порядок сезонної авторегресії та ковзного середнього,
- d – різниця часів,
- D – сезонна різниця часів,
- S – сезонний період і тривалість циклу.

Структура моделі SARIMA показана далі:

$$\begin{cases} \Phi(L)A_p(L^S)(\nabla^d\nabla_S^D x_t) = \Theta_q(L)B_q(L^S)\varepsilon_t, \\ E(\varepsilon_t) = 0, \text{Var}(\varepsilon_t) = \sigma_\varepsilon^2, E(\varepsilon_t|\varepsilon_s) = 0, s \neq t, \\ E(x_s\varepsilon_t) = 0, s < t. \end{cases} \quad (2.18)$$

де  $L$  – оператор затримки;

$A_p(L^S)$  – авторегресійний оператор p -порядку;

$A_q(L^S)$  – сезонний ковзний середній оператор q -порядку;

$\nabla^d = (1 - L)^d$  – різницева операція;

$\nabla_S^D = (1 - L^S)^d$  – сезонна різниця [22].

## 2.3 Алгоритми оцінювання моделі за методологією Бокса-Дженкінса

Початок використання ARMA-моделей пов'язують із створенням методики побудови моделей Г. Боксом та Г. Дженкінсом. Методика передбачає чотири послідовні процедури описані на рис. 2.1.

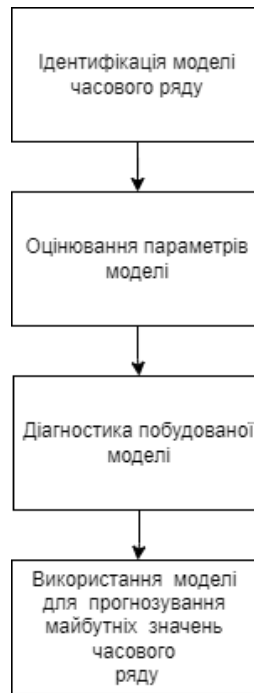


Рис. 2.1 – Методика побудови ARMA-моделі

### 2.3.1 Ідентифікація моделі

Ідентифікація моделі – це процес визначення моделі із мінімальною кількістю параметрів. На цьому етапі необхідно визначити дві незалежні проблеми:

- стаціонарність та порядок  $d$  оператора переходу до послідовних одиниць:

$$z_t = (1 - L)^d y_t \quad (2.19)$$

- вибір параметрів  $p$  і  $q$  (описують стаціонарний ряд як процес ковзної середньої та авторегресії).

«У разі нестационарності ряду для визначення порядку різницевого оператора можна скористатися емпіричним критерієм, сутність якого полягає у тому, що знаходять такі значення  $d$ , за якими вираз:

$$\sqrt{\frac{1}{n-d} \sum_{t=1}^{n-d} (z_t - \bar{z}_t)^2}, \quad (2.20)$$

де  $\bar{z}_t$  – середнє значення стаціонарного процесу  $z_t = \Delta^d y_t, t = 1, 2, \dots, n - d$  буде мінімальним» [19].

Слід зазначити, що величина критерію (2.20) буде зменшуватися при збільшенні  $d$  до тих пір, доки оператор різниці не стане стаціонарним.

Систематичну компоненту можна виключити з аналізу, використовуючи метод найменших квадратів або будь-який інший метод згладжування часового ряду. Після досягнення стаціонарності у ряду, визначається параметр ARMA-моделі. Для цього етапу корисними є графічні методи, а також порівняння автокореляційної та часткової автокореляційної функцій із відповідними характеристиками відомих ARMA-процесів, наведених у табл. 2.1.

Таблиця 2.1

## Характеристики ARMA-базованих моделей

Модель	Автокореляційна функція	Частково автокореляційної функції
Білий шум	Нулі	нулі
МА(1)	0 після $\rho_1$	після $\phi_{11}$ – геометрично спадна

Кінець таблиці 2.1

Модель	Автокореляційна функція	Частково автокореляційної функції
MA(2)	0 після $\rho_2$	після $\phi_{22}$ – геометрично спадна
MA(q)	0 після $\rho_q$	після $\phi_{qq}$ – геометрично спадна
AR(1)	після $\rho_1$ геометрично спадна	після $\phi_{11}$ – нулі
AR(2)	після $\rho_2$ геометрично спадна	після $\phi_{22}$ – нулі
AR(p)	після $\rho_p$ геометрично спадна	після $\phi_{pp}$ – нулі
ARMA(1, 1)	після $\rho_1$ геометрично спадна	після $\phi_{11}$ – спадна
ARMA(p, q)	після $\rho_p$ геометрично спадна	після $\phi_{qq}$ – спадна

Узагальнюючи, при використанні вибірки спостережень, розмір якої зазвичай є обмеженим, слід мати на увазі, що точна відповідність між даними та теоретичною моделлю може бути недосяжною. Це може призвести до вибору двох або трьох пробних ARMA (p, q) моделей, які мають різні значення параметрів часових лагів p у авторегресійному процесі та лагових змінних q у моделі ковзної середньої. Вибір найбільш доцільної моделі для подальшого аналізу та прогнозування здійснюється за допомогою методів діагностичної перевірки, які будуть розглянуті надалі [19].

### 2.3.2 Оцінювання параметрів моделі

Оцінювання параметрів моделі. Після того, як процес ідентифікації визначив початковий варіант стаціонарної ARMA-моделі, цю модель пристосовують до даних спостережень шляхом знаходження оцінок параметрів  $a = (a_1, \dots, a_p)$  та  $b = (b_1, \dots, b_q)$ .

Раніше було показано, що ARMA-модель порядку (p, d, q), що враховує нестационарні процеси, зводиться за допомогою перших різниць до

стаціонарної моделі порядку  $(p, 0, q)$ . Тому процедура обчислення коефіцієнтів розглядається тільки для стаціонарної моделі.

Параметри AR-моделі можуть бути оцінені за допомогою звичайного методу найменших квадратів (виходять зсунуті, але консистентні оцінки), проте його не можна застосувати до MA або ARMA моделей. Наприклад, для MA(1)-моделі (2.21):

$$y_t = \mu + \varepsilon_t - b_1 \varepsilon_{t-1} \quad (2.21)$$

неможливо оцінити параметри, користуючись лише спостереженнями  $y_t$ , оскільки невідомі значення параметрів для розрахунку.

Метод Бокса-Дженкінса пропонує використовувати процедуру нелінійної оптимізації: процедуру пошуку на мережі (grid-search procedure). Це ітеративна процедура, в якій оцінки параметрів мінімізують суму квадратів залишків. Запишемо MA(1)-модель як:

$$\varepsilon_t = y_t - \mu + b \cdot \varepsilon_{t-1} \quad (2.22)$$

Аналізуючи оцінки АКФ та ЧАКФ, можна зробити попередні припущення відносно значень параметрів. Можна використовувати вибіркове середнє (для  $\mu$ ) та першу автокореляцію (для  $b$ ). Припустімо, що вони дорівнюють 100 та 0,2. Тоді модель має вигляд:

$$\varepsilon_t = y_t - 100 + 0,2\varepsilon_{t-1} \quad (2.23)$$

Якщо припустити, що  $\varepsilon_0 = 0$ , то можемо отримати оцінки  $\varepsilon_t$ , де  $t \in [1; n]$  та розрахувати суми квадратів залишків  $s_1$ . Відповідно, вибираючи нові значення  $\mu$  та  $b$  буде отримане нове значення  $s_2$ . Після перевірки початкових даних, що залишилися, можна констатувати, що остаточними оцінками моделі залишаються коефіцієнти моделі, серед яких  $s$  є мінімальними [19].

У минулі часи, у зв'язку із обмеженими можливостями комп'ютерів, дослідники розробляли спеціальні методи для оцінювання коефіцієнтів кожної моделі в аналізі часових рядів. В сучасній науці, вчені створили універсальний метод максимальної правдоподібності, який дозволяє отримувати надійні та ефективні оцінки коефіцієнтів для будь-якої моделі [21].

Основна ідея полягає в тому, що вважається, що дані мають певний ймовірнісний розподіл, і обчислюється ймовірність необхідної події. Ця ймовірність, зазвичай, залежить від невідомих параметрів. З використанням цих даних можна максимізувати ймовірність відбуття цієї події. Коефіцієнти, при яких досягається найвища ймовірність, є оцінками параметрів. Іноді отримати ці оцінки в аналітичному вигляді буває складно, тому застосовують числові методи оптимізації для функції правдоподібності [19].

Для запису моделі ARMA( $p, q$ ) (2.24) потрібно провести спостереження, де  $z = (z_1, z_2, \dots, z_n)$ , де кількість необхідних спостережень за початковим рядом  $y_t$  ( $y_{-d+1}, \dots, y_0, y_1, \dots, y_n$ ) буде дорівнювати  $N = n + d$ :

$$z_t = \sum_{i=1}^p a_i y_{t-i} + \varepsilon_t + \sum_{j=1}^q b_j \varepsilon_{t-j} \quad (2.24)$$

Логарифм функції правдоподібності можна записати у наступному вигляді:

$$\ln L(\sigma^2, A, B) = -\frac{1}{2}(n \ln \sigma^2 + \ln V(A, B) + \frac{Z^T (V(A, B))^{-1} Z}{\sigma^2}) \quad (2.25)$$

Слід зазначити що це не єдиний метод обчислення функції правдоподібності, є більш ефективні [19].

### 2.3.3 Діагностика моделі

Наступним етапом після знаходження оцінок параметрів є перевірка адекватності моделі. Оскільки далі буде використана ARIMA-базована модель, слід навести специфічні додаткові можливості перевірки саме для таких типів моделей.

Будь-яка теоретична модель містить випадкову компоненту. Якщо ця модель є коректною, то залишки є «білим шумом». В табл. 2.2 зазначені правила перевірки залишків в ARIMA моделях.

Таблиця 2.2

Перевірка залишків в ARIMA-базованих моделях

Модель	Правило
AR(p)	$e_t = y_t - \sum_{j=1}^p \hat{a}_j y_{t-j}, t = p + 1, \dots, n$ <p><math>e_t</math> не визначені при <math>t &lt; p</math></p>
MA(q)	<p><math>e_j = 0, j \leq 0, e_1 = y_1</math></p> <p><math>e_2 = y_2 - \hat{b}_1 e_1</math>, тощо</p> <p>для <math>t &gt; q</math>:</p> $e_t = y_t - \sum_{j=1}^q \hat{b}_j e_{t-j}$

Кінець таблиці 2.2

Модель	Правило
ARMA(p, q)	$e_t = 0, t \leq p$ $e_t = y_t - \sum_{j=1}^p \hat{a}_j y_{t-j} - \sum_{j=1}^q \hat{b}_j e_{t-j}, t \leq \max(p, q)$

Після розрахунку залишків необхідно визначити їх відповідність «білому шуму» за допомогою обчислення АКФ та ЧАКФ залишків. Також, необхідно перевірити їх статистичну значимість за критеріями Льюнга-Бокса, Бокса-Пірса.

Не виключені ситуації при яких після попередньої операції можуть залишитись декілька задовільних моделей. В таких ситуаціях слід обирати ту, яка має найменше параметрів. «Для використання цього принципу треба формалізувати правило компромісу між точністю пристосування моделі та кількістю її параметрів» [19].

Найпростішою перевіркою є порівняння моделей. Припустимо, що  $L_0$  є максимальним значенням функції правдоподібності задовільняючої ARMA(p, q) моделі. Далі, потрібно розрахувати максимальне значення функції для моделей ARMA(p, q + 1) та ARMA(p + 1, q). Максимальними будуть значення  $L_1$  та  $L_2$ . Теорія функції правдоподібності стверджує, що модель ARMA(p, q) є коректною тоді, коли  $2(L_1 - L_0)$  та  $2(L_2 - L_0)$  розподілені як  $\chi^2$ . Проте, дана перевірка може давати неправильні результати при сильній кореляції даних.

Іншими інструментами тестування є числові критерії, за значеннями яких можна судити про адекватність моделі. В таблиці 2.3 наведені числові критерії та їх характеристика.

Таблиця 2.3

## Характеристика числових критеріїв

Модель	Бажаний екстремум	Формула
Прогнозовий критерій (FC)	Min	$FC = \frac{\sum_{t=1}^n e_t^2}{n - (p + q)} \left(1 + \frac{p + q}{n}\right)$
Критерій Ханнана-Квіна (HQ)	Min	$HQ = \ln \left( \frac{\sum_{t=1}^n e_t^2}{n - (p + q)} \right) + \frac{c(p + q)}{n} \ln(\ln n), c \geq 2$
Інформаційний критерій Шварца-Ріссанена (SIC)	Min	$SIC = \ln \left( \frac{\sum_{t=1}^n e_t^2}{n - (p + q)} \right) + \frac{p + q}{n} \ln n$
Інформаційний критерій Акаїке (AIC)	Min	$AIC = \ln \left( \frac{\sum_{t=1}^n e_t^2}{n - (p + q)} \right)$
Скоригований коефіцієнт детермінації	1	$\bar{R}^2 = 1 - (1 - R^2) \frac{n - 1}{n - (p + q)}$
Коефіцієнт детермінації	1	$R^2 = 1 - \frac{\sum_{t=1}^n e_t^2}{\sum_{t=1}^n (y_t - \bar{y})^2}$

Вибір між зазначеними критеріями є арбітрарним, оскільки всі статистичні показники вказаного типу виявляють тенденцію до зміни в одному напрямі при збільшенні кількості оцінюваних параметрів. У практичному застосуванні рекомендується вибрати один з цих критеріїв для подальшого дослідження.

### 2.3.4 Використання моделі для прогнозування майбутніх значень часового ряду

У моделях ARIMA, при прогнозуванні майбутніх значень змінної, лагові значення цієї змінної, виступаючи як регресори, можна вважати фіксованими чи випадковими. Це може призвести до умовного чи безумовного прогнозу відповідно. Таким чином, при аналізі ARIMA прогнозів враховують умовні та безумовні аспекти, враховуючи, що умовна дисперсія випадкової величини завжди не перевищує безумовної, що забезпечує більшу точність умовних прогнозів.

При правильно специфікованій моделі можуть виникати дві помилки прогнозу: невизначеність майбутніх значень випадкової величини та відсутність точних значень коефіцієнтів моделі (наявні лише їхні оцінки).

При прогнозуванні за моделлю ARIMA значення регресорів та оцінки коефіцієнтів моделі залежать від наявної вибірки, що робить складним аналітичний вираз для умовної дисперсії помилки прогнозу через спостереження часового ряду. Зазвичай роблять припущення, що коефіцієнти відомі точно, хоча це зменшує дисперсію помилки прогнозу, збільшуючи точність умовних та безумовних прогнозів [19].

Щоб досягти MSE – мінімуму середньоквадратичної помилки – слід використати математичне сподівання:  $M\{y_{t+\tau}|y_1, \dots, y_t\}$ .

Умовний точковий прогноз моделей MA(q), AR(p), ARMA(p, q) асимптотично наближається до математичного сподівання ряду. Дисперсія помилки в свою чергу наближається до дисперсії ряду. Звідси, вплив інформації на точність прогнозу і сам прогноз асимптотично спадає до нуля для стаціонарного процесу [19].

### Висновки до 2 розділу

У другому розділі кваліфікаційної роботи були проаналізовані авторегресійні методи для прогнозування часових рядів. Відміною рисою

ARIMA-базованих моделей з поміж інших є можливість працювати із нестационарними рядами. Також, була обрана SARIMA-базована модель для подальшої побудови моделі прогнозування навантажень в комп'ютерних мережах через наявність сезонної компоненти, яка може стати вирішальною при прогнозі часових рядів пов'язаних із, наприклад, веб-застосунками.

Окрім зазначеного вище, були сформульовані кроки алгоритму оцінювання моделі за методологією Бокса-Дженкінса:

- ідентифікація моделі часового ряду;
- оцінювання параметрів моделі;
- діагностика побудованої моделі;
- використання моделі для прогнозування майбутніх значень часового ряду.

## **РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОЇ МОДЕЛІ ПРОГНОЗУВАННЯ НАВАНТАЖЕНЬ В КОМП'ЮТЕРНИХ МЕРЕЖАХ**

Розробку програмної моделі прогнозування навантажень в комп'ютерних мережах можна розділити на 3 етапи:

- вибір та підготовка набору даних;
- безпосередньо розробка моделі;
- тестування отриманої моделі.

### **3.1 Програмно-апаратний комплекс**

В якості мови програмування було обрано Python через широкий вибір бібліотек для створення авторегресійних моделей. В якості основної бібліотеки було обрано skforecast, яка, згідно із офіційною документацією, є «бібліотекою Python, яка спрощує використання регресорів scikit-learn як, наприклад, багатокрокових прогнозистів. Він також працює з будь-яким регресором, сумісним з API scikit-learn (конвеєри, CatBoost, LightGBM, XGBoost, Ranger...)» [23]. Для візуалізації даних були використані бібліотеки matplotlib та seaborn.

Розробка програмної моделі була проведена із використанням наступного програмного та апаратного забезпечення:

- Google Chrome для використання Google Colab, який дозволяє писати та відтворювати код в браузері;
- Python 3;
- ноутбук Lenovo IdeaPad Gaming 3 з наступними характеристиками:
  - процесор: AMD Ryzen 5600H, 3.3–4.1 GHz;
  - RAM: 16 Gb, DDR4

- SSD: 512 Gb, M2.

## 3.2 Вибір та підготовка набору даних

### 3.2.1 Вибір набору даних

В якості набору даних було обрано часовий ряд «visitas\_por\_dia\_web\_cienciadedatos.csv». Дані були отримані з інтегрованої на веб-сайт служби аналітики Google, і їх можна завантажити із GitHub репозиторію [24]. Включені поля (див. рис. 3.1):

- дата: день/місяць/рік;
- користувачі: загальна кількість користувачів, які відвідують веб-сайт.

	date	users
1		
2	1/7/20	2324
3	2/7/20	2201
4	3/7/20	2146
5	4/7/20	1666
6	5/7/20	1433
7	6/7/20	2195
8	7/7/20	2240
9	8/7/20	2295
10	9/7/20	2279

Рис. 3.1 – Перші десять рядків датасету

Цей набір даних може бути використаний для створення SARIMA-базованої моделі через ряд причин, описаних далі.

- Кількість користувачів на сайті утворює часовий ряд, який можна вивчати.
- Кількість користувачів сайту зазвичай залежить від часу. Поширені сценарії включають пікові години, робочі дні проти вихідних, сезонні зміни та інші часові особливості.

- Авторегресійні моделі можуть ураховувати дану часову залежність, оскільки прогноз будується на попередніх значеннях ряду.
- Кількість користувачів сайту може слугувати індикатором навантаження на сервер та мережу. Їх збільшення може означати більшу кількість запитів до серверу та використання мережі.
- Прогнозування навантаження на сервер та мережу дозволяє оптимізувати ресурси, призначені для обслуговування користувачів. Це може включати планування потужності сервера, резервування мережевої пропускної здатності та інші аспекти інфраструктури.

Отже, пропонований набір даних є підходящою кандидатурою для створення моделі прогнозування навантажень за допомогою SARIMA-базованої моделі.

### **3.2.2 Підготовка набору даних**

Основна ідея авторегресії полягає в тому, що поточне значення часового ряду залежить від попередніх значень цього ж ряду, тому потрібно мати часовий ряд з правильною та сталою частотою, а саме, ряд даних з регулярно розташованими датами [25]. Отже, потрібно провести ряд маніпуляцій із вибіркою даних аби перетворити її на готовий до створення моделі набір. Нижче перераховані кроки, необхідні для підготовки.

Для початку, імпортуємо всі необхідні бібліотеки (рис. 3.2) та компоненти і зчитуємо дані із .csv файлу (рис 3.3).

```
[1] import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
!pip install skforecast[full]
!pip install hvplot
from skforecast.ForecasterAutoreg import ForecasterAutoreg
from skforecast.model_selection import grid_search_forecaster
from skforecast.model_selection import backtesting_forecaster
from skforecast.ForecasterSarimax import ForecasterSarimax
from skforecast.model_selection_sarimax import backtesting_sarimax
from skforecast.model_selection_sarimax import grid_search_sarimax

from sklearn.linear_model import Ridge
from sklearn.preprocessing import StandardScaler
from pmdarima import ARIMA

import hvplot.pandas
```

Рис. 3.2 – Бібліотеки необхідні для розробки комп'ютерної моделі

```
[18] web_traffic = pd.read_csv("visitas_por_dia_web_cienciadatos.csv", sep=',')
web_traffic.head(10)
```

	date	users
0	1/7/20	2324
1	2/7/20	2201
2	3/7/20	2146
3	4/7/20	1666
4	5/7/20	1433
5	6/7/20	2195
6	7/7/20	2240
7	8/7/20	2295
8	9/7/20	2279
9	10/7/20	2155

```
web_traffic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 421 entries, 0 to 420
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0   date    421 non-null    object
1   users   421 non-null    int64
dtypes: int64(1), object(1)
memory usage: 6.7+ KB
```

Рис. 3.3 – Відображення десяти перших записів та опису датасету

Далі, нам необхідно перетворити дату в об'єкт дати та часу і додати його до індексу таблиці (див. рис. 3.4):

- Рядок `web_traffic['date'] = pd.to_datetime(web_traffic['date'], format='%d/%m/%y')` перетворює стовпець 'date' в об'єкт дати та часу, використовуючи формат дати '%d/%m/%y'. П

- припускається, що стовпець 'date' містить рядки у форматі дня/місяць/рік, наприклад, "01/01/22".

- Рядок `web_traffic = web_traffic.set_index('date')` встановлює об'єкт дати та часу 'date' як індекс датафрейму 'web\_traffic'. Це означає, що 'date' стає основним стовпцем для ідентифікації рядків, а не звичайним числовим індексом.

- Рядок `web_traffic = web_traffic.asfreq('1D')` перетворює датафрейм 'web\_traffic' на щоденну частоту, забезпечуючи наявність рядків для кожного дня. Якщо деякі дні відсутні, вони заповнюються значеннями NaN.

- Рядок `web_traffic = web_traffic.sort_index()` сортує датафрейм 'web\_traffic' за зростанням значень індексу, тобто за датою. Результатом є датафрейм, в якому рядки впорядковані за зростанням дат.

```
web_traffic['date'] = pd.to_datetime(web_traffic['date'], format='%d/%m/%y')
web_traffic = web_traffic.set_index('date')
web_traffic = web_traffic.asfreq('1D')
web_traffic = web_traffic.sort_index()
web_traffic.head(5)
```

users	
date	
2020-07-01	2324
2020-07-02	2201
2020-07-03	2146
2020-07-04	1666
2020-07-05	1433

Рис. 3.4 – Перетворення дати та додання до індексу

Вираз `(web_traffic.index == pd.date_range(start=web_traffic.index.min(), end=web_traffic.index.max(), freq=web_traffic.index.freq)).all()` перевіряє, чи всі значення індексу датасету 'web\_traffic' співпадають з діапазоном дат, створеним за допомогою `pd.date_range()` з використанням поточної частоти набору даних. Якщо всі значення співпадають, це означає, що індекс набору даних є регулярним та стабільним, що необхідно для моделі AR.

Виконання цієї перевірки (див. рис. 3.5) допомагає впевнитися, що часовий ряд має правильний індекс і може бути використаний для побудови моделі AR без пропусків або неправильного розташування даних. Це важливий крок перед подальшим аналізом та прогнозуванням з використанням авторегресії.

```
(web_traffic.index == pd.date_range(
    start = web_traffic.index.min(),
    end   = web_traffic.index.max(),
    freq  = web_traffic.index.freq)).all()
True
```

Рис. 3.5 – Перевірка часових значень

Далі, перевіримо чи є пропущені значення (див. рис. 3.6):

```
print(f"Missing values: {web_traffic.isnull().any(axis=1).sum()}")
Missing values: 0
```

Рис. 3.6 – Перевірка наявності пропущених значень

### 3.3 Розробка моделі

Після проведення необхідних операцій із даними, потрібно розділити набір даних. В даному випадку було обрано розділення на три частини (див. рис. 3.7): тренувальний набір даних (`data_train`), валідаційний набір даних (`data_val`) та тестовий набір даних (`data_test`). Також, ці частини було візуалізовано на рис. 3.8.

```

end_train = '2021-03-30 23:59:00'
end_validation = '2021-06-30 23:59:00'

data_train = web_traffic.loc[: end_train, :]
data_val = web_traffic.loc[end_train:end_validation, :]
data_test = web_traffic.loc[end_validation: , :]

print(f"Training dates : {data_train.index.min()} --- {data_train.index.max()} (n={len(data_train)})")
print(f"Validation dates : {data_val.index.min()} --- {data_val.index.max()} (n={len(data_val)})")
print(f"Test dates : {data_test.index.min()} --- {data_test.index.max()} (n={len(data_test)})")

Training dates : 2020-07-01 00:00:00 --- 2021-03-30 00:00:00 (n=273)
Validation dates : 2021-03-31 00:00:00 --- 2021-06-30 00:00:00 (n=92)
Test dates : 2021-07-01 00:00:00 --- 2021-08-25 00:00:00 (n=56)

```

Рис. 3.7 – Розділення датасету на три частини

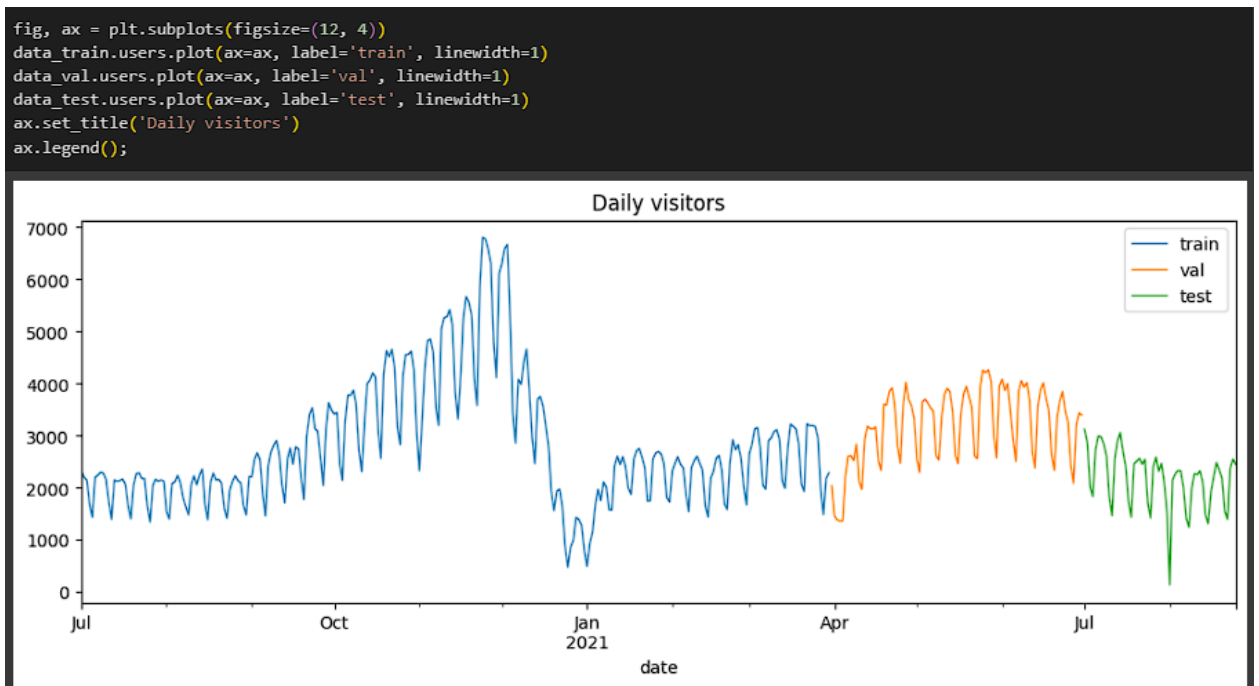


Рис. 3.8 – Візуалізація трьох частин датасету

Наступними кроками є побудова графіків автокореляції (ACF) та часткової автокореляції (PACF):

- Графік ACF (див. рис. 3.9) показує кореляцію між кожним значенням ряду та його попередніми значеннями (лагами).

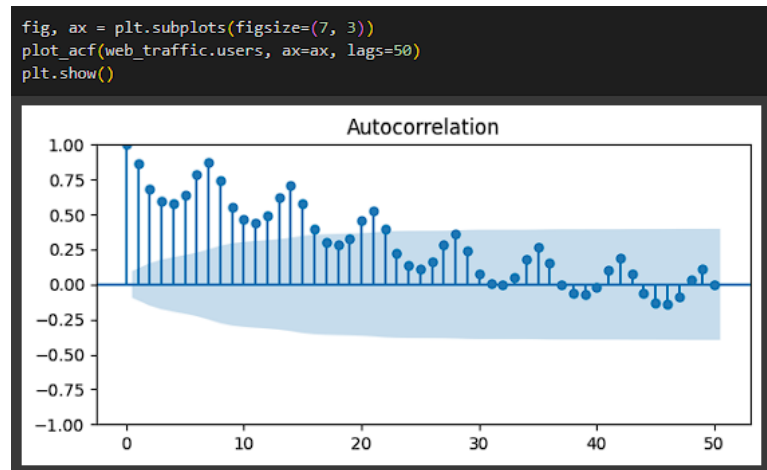


Рис. 3.9 – Графік ACF

- Графік PACF (див. рис. 3.10) показує кореляцію між кожним значенням ряду та його попередніми значеннями, використовуючи тільки проміжні лаги.

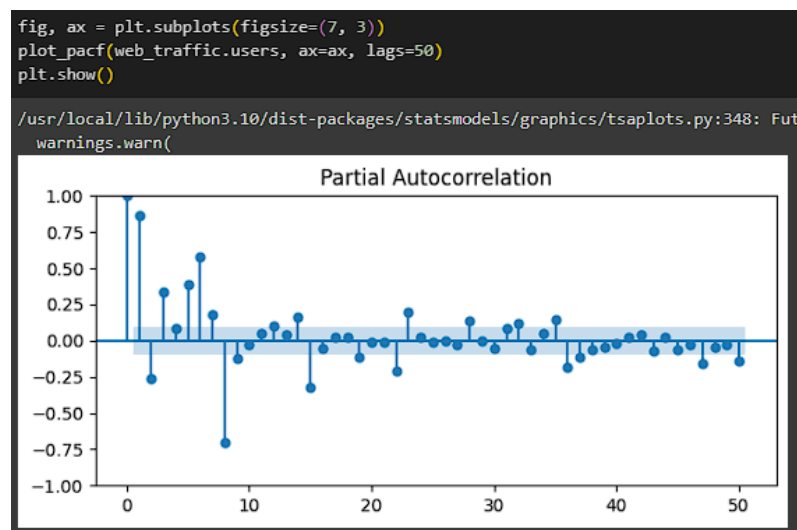


Рис. 3.10 – Графік PACF

Далі можна перейти до побудови авторегресійної моделі. Проте, попередньо, необхідно визначитися із трьома параметрами (p, d, q) для моделі SARIMA. Основний процес підбору параметрів (p, d, q) може бути виконаний за допомогою кроків перерахованих нижче.

- Ступінь різницювання (d). Спочатку важливо визначити ступінь різницювання необхідного для того, щоб зробити часовий ряд стаціонарним. Стаціонарність означає, що статистичні властивості ряду не змінюються з часом. Якщо часовий ряд має тренд або сезонність, необхідно виконати різницювання для видалення цих ефектів. Зазвичай достатньо одного різницювання (d=1) для більшості часових рядів.
- АСФ. Значущі показники АСФ на графіку, що перевищують межу значущості, вказують на значення параметра q.
- РАСФ. Значущі показники РАСФ на графіку, що перевищують межу значущості, вказують на значення параметра p.

Для даної моделі були обрані наступні параметри: 14; 0; 0 (див. рис. 3.11).

```
[11] forecaster_sarimax = ForecasterSarimax(
      regressor      = ARIMA(order=(14, 0, 0), maxiter=250),
      fit_kwargs     = {'disp': 0},
      forecaster_id  = 'web_traffic_sarimax')
```

Рис. 3.11 – Прогноз моделі

Зворотнє тестування моделі необхідне для отримання надійної оцінки продуктивності моделі після розгортання. На даний момент, модель має показник помилки зворотнього тестування, який дорівнює 221.

Метрикою помилки було обрано середню абсолютну помилку. Вона використовує абсолютне значення, щоб зробити всі щоденні помилки

позитивними, перш ніж знайти середнє. Це вирішує проблему врівноваження позитивних і негативних помилок і дозволяє легко пояснити, наскільки хороший чи поганий прогноз [26].

На даний момент, модель дає прогноз із похибкою в 221 користувача на день (див. рис. 3.12).

```
metric, predictions = backtesting_sarimax(
    forecaster      = forecaster_sarimax,
    y               = web_traffic.users,
    initial_train_size = len(web_traffic.loc[:end_validation]),
    steps           = 7,
    metric          = 'mean_absolute_error',
    refit           = False,
    fixed_train_size = False,
    verbose         = True,
    show_progress   = False
)

print(f'Backtest error: {metric}')
predictions.head(5)
```

```
Information of backtesting process
-----
Number of observations used for initial training: 365
Number of observations used for backtesting: 56
  Number of folds: 8
  Number of steps per fold: 7

Data partition in fold: 0
  Training: 2020-07-01 00:00:00 -- 2021-06-30 00:00:00 (n=365)
  Validation: 2021-07-01 00:00:00 -- 2021-07-07 00:00:00 (n=7)
Data partition in fold: 1
  Training: 2020-07-01 00:00:00 -- 2021-06-30 00:00:00 (n=365)
  Validation: 2021-07-08 00:00:00 -- 2021-07-14 00:00:00 (n=7)
Data partition in fold: 2
  Training: 2020-07-01 00:00:00 -- 2021-06-30 00:00:00 (n=365)
  Validation: 2021-07-15 00:00:00 -- 2021-07-21 00:00:00 (n=7)
Data partition in fold: 3
  Training: 2020-07-01 00:00:00 -- 2021-06-30 00:00:00 (n=365)
  Validation: 2021-07-22 00:00:00 -- 2021-07-28 00:00:00 (n=7)
Data partition in fold: 4
  Training: 2020-07-01 00:00:00 -- 2021-06-30 00:00:00 (n=365)
  Validation: 2021-07-29 00:00:00 -- 2021-08-04 00:00:00 (n=7)
Data partition in fold: 5
  Training: 2020-07-01 00:00:00 -- 2021-06-30 00:00:00 (n=365)
  Validation: 2021-08-05 00:00:00 -- 2021-08-11 00:00:00 (n=7)
Data partition in fold: 6
  Training: 2020-07-01 00:00:00 -- 2021-06-30 00:00:00 (n=365)
  Validation: 2021-08-12 00:00:00 -- 2021-08-18 00:00:00 (n=7)
Data partition in fold: 7
  Training: 2020-07-01 00:00:00 -- 2021-06-30 00:00:00 (n=365)
  Validation: 2021-08-19 00:00:00 -- 2021-08-25 00:00:00 (n=7)

Backtest error: 221.16175748400454
```

	pred
2021-07-01	3168.884377
2021-07-02	2982.103716
2021-07-03	2241.585855
2021-07-04	2007.877601
2021-07-05	2986.983944

Рис. 3.12 – Зворотнє тестування моделі

### 3.4 Покращення моделі

Оскільки модель має середню абсолютну похибку в 221 користувача на день, із максимальними значеннями похибки до 1,500 користувачів, була висунута гіпотеза, що розроблена модель може бути покращена. Для цього потрібно змінити параметри ( $p, d, q$ ).

Для початку, слід зазначити, що параметри  $d$  та  $q$  дорівнюють нулю. Ступінь різниціювання можна збільшити до 1, оскільки, зазвичай, такого значення параметру достатньо. Також, спробуємо підвищити значення параметру  $q$  до 1 та 2, оскільки при повторному аналізі ACF було виявлено від 1 до 2 значень, що перевищують межу значущості.

Також, проаналізуємо параметр графік PACF для корегування параметру  $p$ . Було виявлено, що його значення може дорівнювати 21. Тоді, побудуємо моделі із параметрами  $p, d, q$ , зазначеними в табл. 3.1:

Таблиця 3.1

Параметри 10 SARIMA-базованих моделей із різними значеннями вхідних параметрів

Параметри, ( $p, d, q$ )	[21, 1, 1]
	[21, 1, 0]
	[21, 2, 0]
	[21, 1, 2]
	[14, 1, 1]
	[14, 1, 0]
	[21, 0, 0]
	[14, 2, 0]
	[14, 1, 2]
	[14, 0, 0]

Після побудови 10 моделей, були отримані значення, зазначені в табл. 3.2 відсортовані за спаданням.

Таблиця 3.2

## Результати побудови 10 SARIMA-базованих моделей

Параметри, (p, d, q)	Середня абсолютна помилка
[21, 1, 1]	181.84
[21, 1, 0]	183.08
[21, 2, 0]	186.89
[21, 1, 2]	190.78
[14, 1, 1]	194.95
[14, 1, 0]	195.21
[21, 0, 0]	201.23
[14, 2, 0]	201.63
[14, 1, 2]	204.42
[14, 0, 0]	222.3

Проаналізувавши результати, можна засвідчити, що модель [21, 1, 1] має найменший показник середньої абсолютної помилки. Візуалізуємо результати для моделі із зазначеними параметрами як на рис. 3.13:

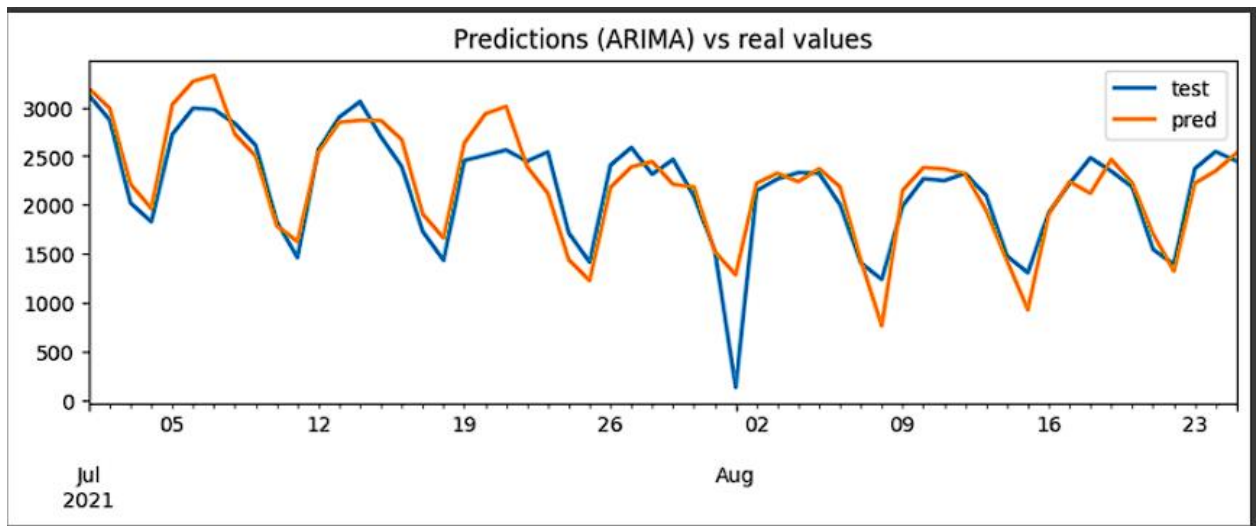


Рис. 3.13 – Візуалізація результатів

Можна побачити, що похибка в майже 1,500 користувачів, яка виникає 1 серпня не вдалося зменшити, проте інші значення стали більш наближеними до лінії тестових даних.

Задля більш наглядних значень, виразимо помилку у MAPE (Mean Absolute Percentage Error) – середня абсолютна відсоткова помилка. Вона вимірює відсоткову середню абсолютну помилку між фактичними та прогнозними значеннями. Тож, початкова модель із параметрами [14; 0; 0] мала помилку в 27%, а кінцева модель із параметрами [21; 1; 1] – 23%. Отже, відбулось покращення прогнозу на 4%.

### **Висновки до 3 розділу**

В даному розділі описана практична реалізація задачі прогнозування навантажень в комп'ютерній мережі у вигляді програмного додатка, створеного на мові Python. Описаний препроцесінг даних, налаштування параметрів моделі на підставі аналізу автокореляційної та часткової автокореляційної функцій. Як результат, була отримана похибка 27%. Було проведено дослідження щодо вибору моделі для забезпечення найкращої якості прогнозування. Після покращення, вдалося зменшити помилку на 4%.

## ВИСНОВКИ

В даній кваліфікаційній роботі було розв'язано **наступні задачі**:

- ознайомлення з існуючими типами моніторингу навантажень в КМ;
- проведення детального аналізу методів прогнозування часових рядів;
- виокремлення однієї авторегресійної моделі для прогнозування навантажень;
- обрання набору даних, який стане основою для створення комп'ютерної моделі;
- розробка комп'ютерної моделі прогнозування навантажень в КМ за допомогою авторегресійних методів;
- тестування та покращення існуючої моделі з метою зменшення похибки.

Далі наведена більш детальна характеристика розв'язаних задач.

По-перше, в роботі були розглянуті основні поняття пов'язані з моніторингом навантажень в КМ. Першим є навантаження, яке можна охарактеризувати як обставити, під час яких по мережі передається така кількість даних, яка досягає або дорівнює граничному показнику пропускної здатності, а наслідком перевантаження може стати зниження пропускної здатності мережі. Другим є визначення комп'ютерних мереж, які у свою чергу є сукупністю деяких вузлів, кожен з яких виконує свої функції і пов'язаний з іншими комунікаційними каналами. Третім є поняття моніторингу, яке можна охарактеризувати як процес, необхідний для ефективного та раціонального управління КМ, який дозволяє оперативно виявляти та реагувати на нетипову поведінку систем.

По-друге, в роботі були проаналізовані методи прогнозування часових рядів з акцентом на авторегресійні методи. З-поміж них, була обрана SARIMA-базована модель через низку переваг, які описані нижче:

- наявність залежності поточного значення від попередніх значень часового ряду;
- можливість адаптації до різних порядків, тобто кількості попередніх значень, які використовуються для прогнозу, що дозволяє оптимально підібрати модель;
- можливість виявлення та використання локальних шаблонів у часових рядах;
- можливість врахування таких факторів, як тренд і сезонність.

Останній фактор є вирішальним через можливу наявну сезонність, яка може вносити додаткові складності під час прогнозування навантажень. Прикладом може слугувати сайт електронної комерції з продажу одягу, який має пікові навантаження кожної пори року через необхідність оновлення гардеробу.

По-третє, в роботі була розроблена комп'ютерна модель прогнозування навантажень із використанням SARIMA-базованої моделі мовою Python. Дана модель навчалася та перевірялася на наборі даних, який був сформований вбудованим в веб-сайт сервісом Google Analytics. Після завершення навчання та перевірки моделі із параметрами  $[14, 0, 0]$ , було спрогнозовано часовий ряд, який має показник помилки зворотнього тестування, що дорівнює 221. Далі було висунуто гіпотезу, що цей показник може бути зменшений. Тому, було проведено повторний аналіз графіків АКФ та ЧАКФ, які дозволили виявити неточність в параметрах моделі і побудувати десять нових моделей з іншими параметрами. З-поміж них була обрана модель із параметрами які дорівнюють  $[21, 1, 1]$ , яка має найменше значення помилки в 182.



- 10 Б. Ю. Жураковський, І.О. Зенів. Комп'ютерні Мережі. Частина 1. Навчальний Посібник 2020. – 336 с.
- 11 Кордяк В., Дронюк І., Федевич О. Інформаційна технологія моніторингу та аналізу трафіку у комп'ютерних мережах. Вісник Національного університету Львівська політехніка. Комп'ютерні науки та інформаційні технології. – 2015. – №. 826. – С. 35-42.
- 12 Коркуна Д. М. Фінансове прогнозування, як основа фінансових планів підприємства. Вісник національного університету Львівська Політехніка. – 2008. – №. 628. – С. 539-545.
- 13 Жуков М. Н. Математична статистика та обробка геологічних даних/Жуков МН–К.: Вища школа, 2008.–481 с. Zhukov, MN (2008) //Matematychna statystyka i obrobka geologichnykh danykh.
- 14 Hyndman, R.J., & Athanasopoulos, G. (2018) Forecasting: principles and practice, 2nd edition, OTexts: Melbourne, Australia.
- 15 Євтушенко О. В. Прогнозування ризиків виробничого травматизму на підприємствах харчової промисловості //Наукові праці Національного університету харчових технологій. – 2013. – №. 51. – С. 46-56.
- 16 Експоненційне згладжування. – 2019. URL: [https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%86%D1%96%D0%B9%D0%BD%D0%B5\\_%D0%B7%D0%B3%D0%BB%D0%B0%D0%B4%D0%B6%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F](https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%86%D1%96%D0%B9%D0%BD%D0%B5_%D0%B7%D0%B3%D0%BB%D0%B0%D0%B4%D0%B6%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F). Дата звернення [10.10.2023].
- 17 Прогнозування Інтуїтивними Методами. – 2016. URL: <http://www.tsatu.edu.ua/ophv/wp-content/uploads/sites/13/prohnozuvannja-intuityvnymy-metodamy.pdf>. Дата звернення [10.10.2023].
- 18 Chatfield C., Xing H. The analysis of time series: an introduction with R. – CRC press, 2019.
- 19 Березька К. М., Маслій В. В. Побудова ARIMA моделей часових рядів для прогнозування інвестицій. – 2015.

- 20 Kolarik T., Rudorfer G. Time series forecasting using neural networks //ACM Sigapl Apl Quote Quad. – 1994. – Т. 25. – №. 1. – С. 86-94.
- 21 Box George E. P., Jenkins Gwilym M. Time series analysis, forecasting and control. Holden-day, Inc. 1976. P. 67.
- 22 Ren S., Cui H. B. Application of SARIMA time series analysis in tax forecast: take Guizhou Province as an example //Journal of Hubei University (Natural Science). – 2021. – Т. 43. – №. 1. – С. 80-85.
- 23 Skforecast Docs – 2022 <https://joaquinamatrodrigo.github.io/skforecast/0.4.3/index.html>. [Дата звернення 10.10.2023].
- 24 Visitas Por Dia Web Cienciadedatos – 2021. – URL: [https://github.com/JoaquinAmatRodrigo/Estadistica-machine-learning-python/blob/master/data/visitas\\_por\\_dia\\_web\\_cienciadedatos.csv](https://github.com/JoaquinAmatRodrigo/Estadistica-machine-learning-python/blob/master/data/visitas_por_dia_web_cienciadedatos.csv). [Дата звернення 10.10.2023].
- 25 Practical Time Series Analysis: Prediction with Statistics and Machine Learning, Aileen Nielsen, 2020.
- 26 Backtesting Forecasts to Estimate Future. – 2017. – URL: <https://www.forecastforge.com/learning/backtesting/>. Дата звернення [10.10.2023].
- 27 Tang B., Matteson D. S. Probabilistic transformer for time series analysis //Advances in Neural Information Processing Systems. – 2021. – Т. 34. – С. 23592-23608.
- 28 Zou Y. et al. Complex network approaches to nonlinear time series analysis //Physics Reports. – 2019. – Т. 787. – С. 1-97.
- 29 Bezabih G. et al. Forecasting cereal crops production using time series analysis in Ethiopia //Journal of the Saudi Society of Agricultural Sciences. – 2023.
- 30 Fernandez I. et al. NATSA: a near-data processing accelerator for time series analysis //2020 IEEE 38th International Conference on Computer Design (ICCD). – IEEE, 2020. – С. 120-129.

## ДОДАТКИ

### Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук  
Кафедра теоретичної та прикладної системотехніки  
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Магістр**  
Галузь знань: **12 – Інформаційні технології**  
Спеціальність: **123 «Комп'ютерна інженерія»**

**ЗАТВЕРДЖУЮ**

Завідувач кафедри теоретичної та  
прикладної системотехніки  
д.т.н., проф. Шматков С. І.



«08 » грудня 2022 року

### З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ

**Кулик Вячеслав Миколайович**

(прізвище, ім'я, по батькові студента)

1. Тема роботи «Модель прогнозування навантажень в комп'ютерних мережах за допомогою авторегресійних методів»

керівник роботи Бакуменко Ніна Станіславівна, кандидат технічних наук, доцент,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затвержені наказом по університету від «10» листопада 2023 року № 4101-5/3197

2. Строк подання студентом роботи 28 .11.2023

3. Перелік питань, які потрібно розробити.

1. Аналіз методів моніторингу навантажень у комп'ютерних мережах.
2. Аналіз методів прогнозування часових рядів.
3. Розробка математичної моделі прогнозування навантажень в комп'ютерних мережах за допомогою авторегресійних методів.
4. Тестування програмної реалізації та аналіз отриманих результатів.

## 4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Аналіз та підбір документації та науково-методичної літератури	08.12.2022 – 08.01.2023
2	Огляд методів моніторингу навантажень у комп'ютерних мережах та прогнозування часових рядів	09.01.2023 – 09.02.2023
3	Дослідження роботи комп'ютерної мережі за допомогою утиліти Ping Plotter	10.02.2023 – 11.03.2023
4	Розробка математичної моделі прогнозування навантажень в комп'ютерних мережах за допомогою авторегресійних методів	12.03.2023 – 13.04.2023
5	Ознайомлення з засобами мови Python для аналізу та прогнозування часових рядів	14.04.2023 – 15.06.2023
6	Програмна реалізація моделі вирішення задачі прогнозування навантажень в комп'ютерних мережах за допомогою авторегресійних методів	16.06.2023 – 17.08.2023
7	Аналіз та тестування отриманих результатів	18.08.2023 – 19.09.2023
8	Збір даних та документації для формування звіту	20.09.2023 – 20.10.2023
9	Оформлення пояснювальної записки	21.10.2023 – 21.11.2023
10	Представлення кваліфікаційної роботи керівнику кваліфікаційних робіт	22.11.2023 – 28.11.2023

5. Дата видачі завдання 08.12.2022

Студент

Кулик В. М.

ініціали, прізвище



підпис

Керівник роботи

Бакуменко Н. С.

ініціали, прізвище



підпис

**Технічне завдання**  
**на розробку програмного виробу**  
**«Модель прогнозування навантажень в комп'ютерних мережах за**  
**допомогою авторегресійних методів»**

Назва розділу	Назва і зміст підрозділу
1. Введення	<p>1.1 Назва: Модель прогнозування навантажень в комп'ютерних мережах за допомогою <u>авторегресійних методів</u>.</p> <p>1.2 Область застосування: інформаційні технології.</p>
2. Підстава для розробки	<p>2.1 Навчальний план ФКН за спеціальністю 123 – Комп'ютерна інженерія.</p> <p>2.2 Завдання на кваліфікаційну роботу, затверджене наказом № 4101-5/3197 привести в Додатку А.</p>
3. Призначення розробки	<p>3.1. Мета розробки програмного виробу полягає у розробці комп'ютерної прогнозування наднормових навантажень, які можуть стати причиною перебоїв у КМ, з метою передчасної адаптації ресурсів.</p> <p>3.2. Призначення програмного виробу: для використання в будь-якій сфері, яка використовує КМ.</p> <p>3.3. Вихідні дані для розробки: прогнозований часовий ряд, який відображає навантаження в КМ.</p>
4. Технічні вимоги до програмного виробу	<p>4.1. Вимоги до функціональних характеристик:</p> <p>Програма повинна:</p> <ol style="list-style-type: none"> <li>1) Представляти з себе комп'ютерну реалізацію прогнозування навантажень за допомогою <u>авторегресійних методів</u> в комп'ютерних мережах;</li> <li>2) Забезпечити можливість навчатися модель за допомогою <u>авторегресійних методів</u>;</li> <li>3) Дозволити користувачу використати інші дані відповідного формату для прогнозу навантажень;</li> </ol>

	<p>4) Забезпечувати можливість виведення результату на екран у вигляді прогнозованого часового ряду;</p> <p>4.2. Вимоги до надійності:</p> <p>1) Програма повинна видавати повідомлення про помилки.</p> <p>4.3. Вимоги до умов експлуатації:</p> <p>1) Умови використання відповідають умовам експлуатації персонального комп'ютера, на якому планується використання, а також інших персональних комп'ютерів, які сумісні з ним.</p> <p>2) Для використання програми потрібно пройти короткий ознайомчий курс.</p> <p>4.4. Вимоги до складу і параметрів технічних засобів:</p> <p>1) Персональний комп'ютер у повній комплектації або ноутбук, який має не менш ніж 8 гігабайт оперативної пам'яті.</p> <p>4.5. Вимоги до інформаційної та програмної сумісності:</p> <p>«Windows» та «Linux» операційних систем тих версій, на які можна встановити останню версію <u>Google Chrome</u> або будь-якого іншого веб-браузера;</p> <p>6) Вимоги до маркування та упаковки (не висуваються);</p> <p>7) Вимоги до транспортування і зберігання (не висуваються);</p> <p>8) Спеціальні вимоги (не пред'являються).</p>
<p>5. Вимоги до програмної документації.</p>	<p>Програмною документацією до виробу «Модель прогнозування навантажень в комп'ютерних мережах за допомогою <u>авторегресійних методів</u>» вважати:</p> <p>1) Дане Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Програму і методику випробувань розробленого програмного виробу (представити у вигляді Додатку В до пояснювальної записки до кваліфікаційної роботи).</p>

	3) Опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).	
	4) Лістинг програмного коду (представити у Додатку Г).	
6. Техніко-економічні показники	6.1 Визначення економічних переваг методу у порівнянні з вітчизняними та зарубіжними аналогами – непотрібна.	
	6.2 Оцінка економічної ефективності – непотрібна.	
7. Стадії і етапи розробки	Дата	Назва етапу
	08.12.2022-08.01.2023	Підбір наукової літератури.
	09.01.2023-09.02.2023	Огляд методів моніторингу навантажень у комп'ютерних мережах та прогнозування часових рядів.
	12.03.2023-13.04.2023	Розробка математичної моделі прогнозування навантажень в комп'ютерних мережах за допомогою <u>авторегресійних</u> методів.
	14.04.2023-15.06.2023	Ознайомлення з засобами мови <u>Python</u> для аналізу та прогнозування часових рядів.
	16.06.2023-17.08.2023	Програмна реалізація моделі вирішення задачі прогнозування навантажень в комп'ютерних мережах за допомогою <u>авторегресійних</u> методів.
	18.08.2023-19.09.2023	Аналіз та тестування отриманих результатів
	20.09.2023-20.10.2023	Збір даних та документації для формування звіту
8. Порядок контролю і приймання	8.1 Перевірку ходу розробки програмного виробу Керівнику робіт виконувати 1 раз в 3 тижні.	

	<p>8.2 Випробування програмного виробу відповідно до Програми і методики випробувань провести на базі комп'ютерного класу.</p> <p>8.3 Захист розробленого програмного виробу провести на засіданні атестаційної комісії.</p> <p>8.4 Пояснювальну записку представити на паперових носіях в одному примірнику, в електронному вигляді - на CD-диску в одному екземплярі.</p>
--	---

Виконавець

студент групи КІ-61

Кулик В.М.



Замовник

к. т. н., доцент, доцент кафедри ТПС

Бакуменко Н.С.



**Програма і методика випробувань  
програмного виробу**

«Модель інформаційної системи класифікації пацієнтів за допомогою  
ймовірнісних штучних мереж»

**1 Об'єкт випробувань**

1.1 Найменування випробуваного програмного виробу: «Модель прогнозування навантажень в комп'ютерних мережах за допомогою авторегресійних методів»

1.2 Область його застосування: інформаційні технології.

1.3 Умовне позначення розробки (при необхідності): відсутнє.

**2. Мета випробувань**

Перевірка правильності функціонування комп'ютерної моделі.

**3. Загальні положення**

**3.1 Підстави для проведення випробувань**

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

**3.2 Місце і тривалість випробувань**

Приймальні (приймально-здавальні) випробування проводяться дистанційно в період роботи атестаційної комісії.

**3.3 Обсяг випробувань**

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

### 3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання за участю Замовника, Виконавця та інших осіб, присутніх на засіданні в дистанційному режимі.

### 4. Вимоги до програми або програмного виробу

Модель повинна:

- 1) Представляти з себе комп'ютерну реалізацію прогнозування навантажень за допомогою авторегресійних методів в комп'ютерних мережах;
- 2) Забезпечити можливість навчання модель за допомогою авторегресійних методів;
- 3) Дозволити користувачу використати інші дані відповідного формату для прогнозу навантажень;
- 4) Забезпечувати можливість виведення результату на екран у вигляді прогнозованого часового ряду;

#### 4.2. Вимоги до надійності:

- 1) Програма повинна видавати повідомлення про помилки.

#### 4.3. Вимоги до умов експлуатації:

- 1) Умови використання відповідають умовам експлуатації персонального комп'ютера, на якому планується використання, а також інших персональних комп'ютерів, які сумісні з ним.

- 2) Для використання програми потрібно пройти короткий ознайомчий курс.

#### 4.4. Вимоги до складу і параметрів технічних засобів:

- 1) Персональний комп'ютер у повній комплектації або ноутбук, який має не менш ніж 8 гігабайт оперативної пам'яті.

#### 4.5. Вимоги до інформаційної та програмної сумісності:

«Windows» та «Linux» операційних систем тих версій, на які можна встановити останню версію Google Chrome або будь-якого іншого веб-браузера.

- 4.6 Вимоги до маркування та упаковки (не висуваються);
- 4.7 Вимоги до транспортування і зберігання (не висуваються);
- 4.8 Спеціальні вимоги (не пред'являються).

## **5. Вимоги до програмної документації**

- 5. Вимоги до програмної документації.

Програмною документацією до виробу «Модель прогнозування навантажень в комп'ютерних мережах за допомогою авторегресійних методів» вважати:

1) дане Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).

2) програму і методику випробувань розробленого програмного виробу (представити у вигляді Додатку В до пояснювальної записки до кваліфікаційної роботи).

3) опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).

- 4) лістинг програмного коду (представити у Додатку Г).

## **6. Засоби і порядок випробувань**

### **6.1 Засоби випробувань**

Випробування проводяться на технічних засобах, таких як персональний комп'ютер у повній комплектації або ноутбук, який має не менш ніж 8 гігабайт оперативної пам'яті.

Випробування проводяться з використанням програмних засобів, таких як «Windows» та «Linux» операційних систем тих версій, на які можна встановити останню версію Google Chrome або будь-якого іншого веб-браузеру.

## 6.2 Порядок проведення випробувань

### 1) Перевірка програмної документації

1.1. Перевірка складу програмної документації. Перевірку здійснювати за критерієм наявності, представленої в ТЗ документації.

1.2. Критерієм успішності тесту вважати відповідність наявної документації згідно зі списком в ТЗ та відповідність якості наявної документації згідно з вимогами ЕСПД.

1.3. Перевірка якості програмної документації. Перевірку здійснювати за критерієм відповідності вимогам ЕСПД.

1.5. Модель працює відповідно до умов експлуатації ОС «Windows» версії, яка підтримує останні оновлення сучасних веб-браузерів.

### 2) Для роботи необхідно встановлений веб браузер.

Тест 1. Перевірка сумісності моделі з програмним забезпеченням

Критерієм успішності вважати сумісність успішний запуск моделі в середовищі Google Colab.

Порядок проведення випробувань:

- Запуск програми здійснюється переходу на веб-сайт середовища Google Colab.

- Копіюється програмний код із Додатку Г.

- Завантажується набір даних в папку «Файли» вручну.

- Запускається код натиском на відповідну кнопку.

- Завантажуються бібліотеки і будується часовий ряд.

Висновки: після проведення тесту встановлено, що створена модель запускається та функціонує в встановленому браузері на відповідній ОС, бібліотеки завантажуються та будується часовий ряд.

Тест 2. Відповідність створеної моделі до поставленої мети

Для проведення випробувань будуть проведені заходи, описані у третьому розділі кваліфікаційної роботи. Критерієм успішності будуть слугувати коректні вихідні дані.

	params	mean_absolute_error	order
8	{'order': (21, 1, 1)}	181.844647	(21, 1, 1)
6	{'order': (21, 1, 0)}	183.088047	(21, 1, 0)
7	{'order': (21, 2, 0)}	186.894731	(21, 2, 0)
9	{'order': (21, 1, 2)}	190.786303	(21, 1, 2)
3	{'order': (14, 1, 1)}	194.957459	(14, 1, 1)
1	{'order': (14, 1, 0)}	195.215985	(14, 1, 0)
5	{'order': (21, 0, 0)}	201.234168	(21, 0, 0)
2	{'order': (14, 2, 0)}	201.630797	(14, 2, 0)
4	{'order': (14, 1, 2)}	204.423642	(14, 1, 2)
0	{'order': (14, 0, 0)}	221.161757	(14, 0, 0)

Рисунок В.1 — Результат виконання Тесту 2

Висновок: після виконання тесту 2 було підтверджено, що тестування відповідає поставленій меті, а саме: результати тесту є ідентичними результатам, описаним у третьому розділі.

Висновки: 1 та 2 тести пройшли успішно, поставлені цілі були досягнуті, модель будується та надає точний прогноз.

Виконавець Кулик В.М.

**Лістинг коду програмної реалізації**

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf

!pip install skforecast[full]

!pip install hvplot

from skforecast.ForecasterAutoreg import
ForecasterAutoreg

from skforecast.model_selection import
grid_search_forecaster

from skforecast.model_selection import
backtesting_forecaster

from skforecast.ForecasterSarimax import
ForecasterSarimax

from skforecast.model_selection_sarimax import
backtesting_sarimax

from skforecast.model_selection_sarimax import
grid_search_sarimax
```

```
from sklearn.linear_model import Ridge

from sklearn.preprocessing import StandardScaler

from pmdarima import ARIMA

import hvplot.pandas

web_traffic =
pd.read_csv("visitas_por_dia_web_cienciadedatos.csv",
sep=',')

web_traffic.head(10)

web_traffic.info()

web_traffic['date'] =
pd.to_datetime(web_traffic['date'], format='%d/%m/%y')

web_traffic = web_traffic.set_index('date')

web_traffic = web_traffic.asfreq('1D')

web_traffic = web_traffic.sort_index()

web_traffic.head(5)

(web_traffic.index == pd.date_range(
start =
web_traffic.index.min(),
```

```

end = web_traffic.index.max(),
freq =
web_traffic.index.freq).all()

print(f"Missing values:
{web_traffic.isnull().any(axis=1).sum()}")

end_train = '2021-03-30 23:59:00'
end_validation = '2021-06-30 23:59:00'

data_train = web_traffic.loc[: end_train, :]
data_val =
web_traffic.loc[end_train:end_validation, :]
data_test = web_traffic.loc[end_validation: , :]

print(f"Training dates : {data_train.index.min()}
--- {data_train.index.max()} (n={len(data_train)})")

print(f"Validation dates : {data_val.index.min()} -
-- {data_val.index.max()} (n={len(data_val)})")

print(f"Test dates : {data_test.index.min()} -
-- {data_test.index.max()} (n={len(data_test)})")

fig, ax = plt.subplots(figsize=(12, 4))

```

```

    data_train.users.plot(ax=ax,          label='train',
linewidth=1)

    data_val.users.plot(ax=ax,          label='val',
linewidth=1)

    data_test.users.plot(ax=ax,        label='test',
linewidth=1)

    ax.set_title('Daily visitors')

    ax.legend();

fig, ax = plt.subplots(figsize=(7, 3))
plot_acf(web_traffic.users, ax=ax, lags=50)
plt.show()

fig, ax = plt.subplots(figsize=(7, 3))
plot_pacf(web_traffic.users, ax=ax, lags=50)
plt.show()

forecaster_sarimax = ForecasterSarimax(
                                regressor          =
ARIMA(order=(14, 0, 0), maxiter=250),
                                fit_kwargs        = {'disp':
0},
                                forecaster_id     =
'web_traffic_sarimax')
```

```

metric, predictions = backtesting_sarimax(
                                forecaster           =
forecaster_sarimax,
                                y                   =
web_traffic.users,
                                initial_train_size  =
len(web_traffic.loc[:end_validation]),
                                steps               = 7,
                                metric              =
'mean_absolute_error',
                                refit              = False,
                                fixed_train_size    =
False,
                                verbose             = True,
                                show_progress       = False)

```

```
print(f'Backtest error: {metric}')
```

```
predictions.head(5)
```

```

metric, predictions = backtesting_sarimax(
                                forecaster           =
forecaster_sarimax,

```

```

y =
web_traffic.users,

        initial_train_size =
len(web_traffic.loc[:end_validation]),

        steps = 7,

        metric =
'mean_absolute_percentage_error',

        refit = False,

        fixed_train_size =
False,

        verbose = True,

        show_progress = False

    )

```

```
print(f'Backtest error: {metric}')
```

```
fig, ax = plt.subplots(figsize=(10, 3))
```

```
data_test.loc[predictions.index,
'users'].plot(linewidth=2, label='test', ax=ax)
```

```
predictions.plot(linewidth=2, label='prediction',
ax=ax)
```

```
ax.set_title('Predictions (ARIMA) vs real values')
```

```
ax.legend();
```

```
param_grid = {'order': [(14, 0, 0), (14, 1, 0), (14, 2, 0), (14, 1, 1), (14, 1, 2),
                        (21, 0, 0), (21, 1, 0), (21, 2, 0), (21, 1, 1), (21, 1, 2)]}
```

```
results = grid_search_sarimax(
    forecaster = forecaster_sarimax,
    y = web_traffic.users,
    param_grid = param_grid,
    initial_train_size = len(web_traffic.loc[:end_validation]),
    steps = 7,
    metric = 'mean_absolute_error',
    refit = False,
    fixed_train_size = False,
    return_best = True,
    verbose = False)
```

```
results
```

```
metric, predictions = backtesting_sarimax(
```

```

                                forecaster          =
forecaster_sarimax,
                                y                  =
web_traffic.users,
                                initial_train_size =
len(web_traffic.loc[:end_validation]),
                                steps              = 7,
                                metric             =
'mean_absolute_error',
                                refit             = False,
                                fixed_train_size   =
False,
                                verbose           = True,
                                show_progress      = False)

print(f'Backtest error: {metric}')

fig, ax = plt.subplots(figsize=(10, 3))

data_test.loc[predictions.index,
'users'].plot(linewidth=2, label='test', ax=ax)

predictions.plot(linewidth=2, label='prediction',
ax=ax)

ax.set_title('Predictions (ARIMA) vs real values')

ax.legend()
```

```
metric, predictions = backtesting_sarimax(
    forecaster =
forecaster_sarimax,
    y =
web_traffic.users,
    initial_train_size =
len(web_traffic.loc[:end_validation]),
    steps = 7,
    metric =
'mean_absolute_percentage_error',
    refit = False,
    fixed_train_size =
False,
    verbose = True,
    show_progress = False
)

print(f'Backtest error: {metric}')
```