

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра системного аналізу та інформаційних технологій

**Магістерська кваліфікаційна робота на тему:**

**«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЦНИ НА  
ПРИРОДНІЙ ГАЗ»**

Виконав: студент 2 курсу, групи ЗІСТ-22м  
спеціальності 126 «Інформаційні системи  
та технології»


 Олександр ЛАДУБ

Керівник: к.т.н., доц. каф. САІТ

 Олексій КОЗАЧКО

« 01 » 12 2023 р.

Опонент: к.т.н., доц. каф. КН

 Володимир ОЗЕРАНСЬКИЙ

« 05 » 12 2023 р.

**Допущено до захисту**

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН


« 01 » 12 2023 р.

Вінниця ВНТУ – 2023 рік

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра системного аналізу та інформаційних технологій  
Рівень вищої освіти – II-й (магістерський)  
Галузь знань – 12 Інформаційні технології  
Спеціальність – 126 Інформаційні системи та технології  
Освітньо-професійна програма – Інформаційні технології аналізу даних та зображень

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

 д.т.н., проф. Віталій МОКІН

«01» 09 2023 року

**ЗАВДАННЯ**  
**НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**  
Ладубу Олександрю Васильовичу

1. Тема роботи: «Інформаційна технологія аналізу та прогнозування ціни на природній газ»

керівник роботи: Олексій КОЗАЧКО, к.т.н., доц. каф. САІТ

затверджені наказом ВНТУ від «18» 09 2023 року № 247

2. Термін подання студентом роботи 30.11.2023 р.

3. Вихідні дані до роботи:

- 1) Kaggle Dataset „Natural Gas Price Actually 2023”. URL: <https://www.kaggle.com/datasets/laduboleksandr/natural-gas-price-actually-2023>;
- 2) Мінфін “Ціни на газ на Нью-Йоркській товарній біржі (NYMEX)”. URL: <https://index.minfin.com.ua/ua/markets/gas/nymex/>.

4. Зміст текстової частини:

- 1) Загальна характеристика об'єкту досліджень;
- 2) Вибір оптимальних налаштувань для розв'язання поставленої задачі;
- 3) Розроблення інформаційної технології аналізу та прогнозування ціни на природній газ;
- 4) Економічна частина.

5. Перелік ілюстративного матеріалу:

- 1) Графік динаміки ціни;
- 2) Лінійна діаграма з аналізом ціни;
- 3) Результат перевірки часового ряду на стаціонарність;
- 4) Графік відображення навчальної та тестової вибірки;
- 5) Графік порівнянь прогнозованих та фактичних значень на тестових даних;
- 6) Графік прогнозу;
- 7) Графік порівняння прогнозованих даних з реальними;
- 8) Таблиця порівняння прогнозованих даних з реальними.

## 6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта         | Підпис, дата   |                  |
|--------|---|----------------|------------------|
|        |   | завдання видав | завдання прийняв |
| 4      | Наталія БУРЄННІКОВА,<br>д. е. н., проф. каф. ЕПВМ | 20.10.23       | 05.11.23         |

7. Дата видачі завдання « 01 » 09 2023 року

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва та зміст етапу  | Термін виконання |            | Примітка |
|-------|---|------------------|------------|----------|
|       |   | початок          | закінчення |          |
| 1     | Аналіз предметної області   | 01.09            | 15.09      | визн     |
| 2     | Вибір оптимальних налаштувань для розв'язання поставленої задачі                    | 15.09            | 01.10      | визн     |
| 3     | Розроблення інформаційної технології аналізу та прогнозування ціни на природній газ | 01.10            | 20.10      | визн     |
| 4     | Економічна частина  | 20.10            | 05.11      | визн     |
| 5     | Оформлення матеріалів до захисту МКР  | 05.11            | 30.11      | визн     |

Студент



Олександр ЛАДУБ

Керівник роботи



Олексій КОЗАЧКО

## АНОТАЦІЯ

УДК 004.09

Ладуб О.В. Інформаційна технологія аналізу та прогнозування ціни на природній газ. Магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2023. 114 с.

На укр. мові. Бібліогр.: 33 назв; рис.: 88; табл.: 13.

В магістерській кваліфікаційній роботі проведено загальну характеристику об'єкту дослідження, здійснено огляд та аналіз відомих аналогів, обґрунтовано доцільність даного дослідження, Проведено опис моделі, яка буде використовуватись для прогнозування даних та здійснено вибір мови програмування та середовища розробки для подальшої роботи. Наступний крок включив у себе моделювання даних та оцінку оптимальної моделі. На основі цього було здійснено прогнозування ціни на природній газ.

Ілюстративна частина складається з 8 плакатів.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження інформаційної технології аналізу та прогнозування ціни на природній газ.

Ключові слова: інформаційні технології, машинне навчання, природній газ, нейронна мережа, прогнозування, python.

## **ABSTRACT**

Ladub O.V. Information technology for analysis and forecasting of natural gas prices. Master's thesis in specialty 126 – information systems and technologies, educational and professional program – information technology data and image analysis. Vinnytsia: VNTU, 2023. 114 p.

In Ukrainian language. Bibliography: 33 titles; fig.: 88; table: 13.

The master's qualification work provides a general characterization of the research object, conducts a review and analysis of known analogs, justifies the relevance of the study. The description of the model, which will be used for data forecasting, is presented, and the choice of programming language and development environment for further work is made. The next step included data modeling and evaluation of the optimal model. Based on this, forecasting of natural gas prices was carried out.

The illustrative part consists of 8 posters.

In the section of the economic part the question of expediency of development and introduction of information technology for the analysis and forecasting of natural gas prices are considered.

**Keywords:** information technology, machine learning, natural gas, neural network, forecasting, python.

## ЗМІСТ

|   |    |
|---|----|
| ВСТУП.....  | 4  |
| 1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ’ЄКТУ ДОСЛІДЖЕНЬ .....  | 6  |
| 1.1 Опис та аналіз об’єкта досліджень .....   | 6  |
| 1.2 Розгляд технологій прогнозування даних .....  | 8  |
| 1.3 Огляд та аналіз відомих аналогів .....  | 11 |
| 1.4 Висновки .....  | 18 |
| 2 ВИБІР ОПТИМАЛЬНИХ НАЛАШТУВАНЬ ДЛЯ РОЗВ’ЯЗАННЯ<br>ПОСТАВЛЕНОЇ ЗАДАЧІ.....                    | 20 |
| 2.1 Опис моделі, яка буде використовуватись для прогнозування даних.....                      | 20 |
| 2.2 Вибір мови програмування та середовища розробки.....                                      | 22 |
| 2.3 Визначення виду машинного навчання .....  | 27 |
| 2.4 Вибір та опис моделей машинного навчання.....   | 29 |
| 2.5 Вибір та опис бібліотек Python .....  | 31 |
| 2.6 Висновки .....  | 33 |
| 3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА<br>ПРОГНОЗУВАННЯ ЦІНИ НА ПРИРОДНІЙ ГАЗ..... | 34 |
| 3.1 Опис ознак та зчитування вхідного набору даних.....                                       | 34 |
| 3.2 Очищення даних.....   | 37 |
| 3.3 Розвідувальний аналіз даних.....  | 40 |
| 3.4 Моделювання даних.....  | 48 |
| 3.5 Оцінка моделі.....  | 64 |
| 3.6 Прогнозування.....  | 72 |
| 3.7 Висновки .....  | 77 |
| 4 ЕКОНОМІЧНА ЧАСТИНА.....   | 79 |
| 4.1 Проведення комерційного та технологічного аудиту науково-технічної<br>розробки .....      | 79 |
| 4.2 Визначення рівня конкурентоспроможності розробки .....                                    | 83 |
| 4.3 Розрахунок витрат на проведення науково-дослідної роботи.....                             | 86 |

|  |     |
|--|-----|
| 4.3.1 Витрати на оплату праці.....   | 87  |
| 4.3.2 Відрахування на соціальні заходи .....   | 91  |
| 4.3.3 Сировина та матеріали.....   | 91  |
| 4.3.4 Розрахунок витрат на комплектуючі.....   | 93  |
| 4.3.5 Спецустаткування для наукових (експериментальних) робіт .....  | 93  |
| 4.3.6 Програмне забезпечення для наукових (експериментальних) робіт .....  | 94  |
| 4.3.7 Амортизація обладнання, програмних засобі та приміщень .....   | 96  |
| 4.3.8 Паливо та енергія для науково-виробничих цілей .....   | 97  |
| 4.3.9 Службові відрядження.....  | 98  |
| 4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....                                     | 99  |
| 4.3.11 Інші витрати.....   | 100 |
| 4.3.12 Накладні (загальновиробничі) витрати.....   | 100 |
| 4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором..... | 102 |
| 4.5 Висновки .....   | 106 |
| ВИСНОВКИ.....  | 108 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....   | 110 |
| Додаток А (обов'язковий). Технічне завдання.....   | 114 |
| Додаток Б (обов'язковий). Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень.....       | 116 |
| Додаток В (довідковий). Лістинг програми .....   | 117 |
| Додаток Г (обов'язковий). Ілюстративна частина .....   | 132 |

## ВСТУП

**Актуальність теми.** У сучасному світі прогнозування ціни на природній газ стало невід'ємною частиною економічного вирішення завдань. Цінові прогнози охоплюють різноманітні аспекти, такі як геополітична обстановка, технологічні тенденції у видобутку та транспортуванні газу, а також погодні умови, які впливають на попит і подання. Прогнози також охоплюють розвиток альтернативних джерел енергії та їх вплив на ринок природного газу.

Дані прогнози є важливим елементом прийняття рішень в енергетичній сфері. Вони допомагають планувати видобуток, розробляти стратегії транспортування та визначати оптимальні напрямки інвестицій. Геополітичні фактори, такі як ситуація на світовому ринку, політичні рішення та міжнародні відносини, впливають на прогнозування цін на природній газ.

Важливо відзначити, що прогнозування цін на природній газ не лише визначає економічні аспекти, але і сприяє сталому розвитку та ефективному використанню енергетичних ресурсів. В контексті сучасних викликів енергетичної ефективності та збереження навколишнього середовища, прогнозування цін на природній газ грає ключову роль у формуванні стратегій енергетичної політики.

**Мета і завдання роботи.** Метою даної роботи є поліпшення точності прогнозування ціни на природній газ. Використавши спеціальне програмне середовище та методи машинного навчання, можна провести аналіз даних та здійснити прогноз ціни на природній газ.

Для досягнення мети дослідження нам потрібно розв'язати такі задачі:

- провести загальну характеристику об'єкту досліджень;
- здійснити вибір оптимальних налаштувань для розв'язання поставленої задачі;
- розробити інформаційну технологію аналізу та прогнозування ціни на природній газ.

**Об'єктом дослідження** магістерської кваліфікаційної роботи є процес аналізу та прогнозування ціни на природній газ.



**Предметом дослідження** магістерської кваліфікаційної роботи є методи машинного навчання, які ґрунтуються на сучасних інформаційних технологіях щодо аналізу та прогнозування ціни на природній газ.

**Новизна одержаних результатів.** Дістала подальший розвиток інформаційна технологія прогнозування ціни на природній газ, яка дозволяє, за допомогою методів машинного навчання, підвищити точність даного прогнозування. Цей прогрес досягнуто завдяки використанню потужних бібліотек для аналізу даних та складних моделей, які використовують часові ряди.

**Практичне значення.** Практичне значення цього дослідження полягає в поліпшенні точності прогнозування ціни на природній газ. Отримані результати є значущими для розробки ефективних стратегій в економічній та фінансовій сферах.

**Апробація результатів магістерської кваліфікаційної роботи.** Результати роботи доповідались на ЛІІ Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (м. Вінниця, 2023-2024 рр.).

**Публікації результатів магістерської кваліфікаційної роботи.** Опубліковано тези на ЛІІ Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації (м. Вінниця, 2023-2024 рр.) [1].

# 1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕНЬ

## 1.1 Опис та аналіз об'єкта досліджень

Природний газ (Gas) — суміш газів, що утворилася в надрах землі при анаеробному розкладанні органічних речовин та/або при конденсації у пастках еманацій з верхньої мантії. Як правило, це суміш газоподібних вуглеводнів (метану, етану, пропану, бутану тощо), що утворюється в земній корі та широко використовується як високоекономічне паливо на електростанціях, у чорній та кольоровій металургії, цементній та скляній промисловості, у процесі виробництва будматеріалів та для комунально-побутових потреб, а також як сировина для отримання багатьох органічних сполук.

Природний газ є корисною копалиною. Часто є побічним газом при видобутку нафти. Природний газ у пластових умовах (умовах залягання в земних надрах) перебуває в газоподібному стані у вигляді окремих скупчень (газові поклади) або у вигляді газової шапки нафтогазових родовищ — це вільний газ, або в розчиненому стані в нафті або воді (у пластових умовах), а в стандартних умовах (0,101325 МПа і 20 °С) — тільки в газоподібному стані. Також природний газ може перебувати у вигляді газогідратів [2].

Розміщення природного газу в земній корі виглядає наступним чином: найближче до поверхні ми спостерігаємо метан у вугільних родовищах. Нижче — конвенційні родовища природного газу та попутного нафтового газу. Ще нижче — газ, «закупорений» в щільних породах. Під ним — шар насичених газом сланців.

Запаси природного газу: 21,4 % світових запасів природного газу належать Росії. На другому місці — Іран (15,9 % світових запасів).

З 2001 року діє Форум країн-експортерів газу (GECF), до складу якого входять: Алжир, Болівія, Бруней, Венесуела, Єгипет, Індонезія, Іран, Катар, Лівія, Малайзія, Нігерія, Об'єднані Арабські Емірати, Оман, Росія, Тринідад і Тобаго. GECF називають «газовим ОПЕК».

Також великі запаси газу мають Норвегія, США, Канада.

У 2018 році за даними американських інституцій доведені запаси природного газу по країнах зафіксовані на такому рівні: 21 % — Росія ; 17,1 % — Іран; 12,2 % — Катар; 4,4 % — США; 4,4 % — Саудівська Аравія; 3,8 % — Туркменістан; 3,1 % — ОАЕ; 2,9 % — Венесуела; 2,8 % — Нігерія; 2,8 % — Китай; 22,4 % — інші країни. За інформацією від British Petroleum на 2013 рік, ситуація трошки відрізняється: загальні світові ресурси становлять 187 300 кубічних кілометрів, Іран має 33 600, Росія – 32 900, Катар – 25 100, Туркменістан – 17 500, а у США – 8500. Варто відзначити, що за оцінками, обсяг "нетрадиційного" газу, такого як сланцевий газ, оцінюється приблизно в 900 000 км<sup>3</sup> [3].

Прогнозування ціни на природній газ має важливе значення і для енергетичного сектору, і для економіки загалом. Ось деякі ключові аспекти щодо важливості та перспектив прогнозування ціни на природній газ:

- Економічна стабільність: Ціна природного газу має прямий вплив на витрати підприємств та домогосподарств. Точні прогнози дозволяють компаніям та споживачам планувати свої бюджети і вирішувати, як ефективно використовувати ресурси;

- Енергетична стратегія: Прогнози цін на природній газ впливають на рішення щодо розвитку енергетичної інфраструктури. Це може стосуватися будівництва газопроводів, електростанцій, а також розвитку альтернативних джерел енергії;

- Постачання енергії: Важливість прогнозів цін на природній газ стосується забезпечення надійного постачання енергії. Якщо ціни на газ надто високі або нестабільні, це може призвести до енергетичних криз та перебоїв у постачанні енергії;

- Геополітична стабільність: Перспективи цін на газ можуть впливати на геополітичні відносини між країнами, особливо тими, які є великими експортерами або імпортерами газу. Розуміння тенденцій у цінах допомагає країнам розвивати свою енергетичну стратегію та уникати конфліктів;

- Екологічний вплив: Прогнозування цін на природній газ також важливе з екологічної точки зору. Зниження цін на газ може сприяти його більш широкому

використанню як більш чистого джерела енергії порівняно з іншими видами палива, такими як вугілля або нафта.

Перспективи прогнозування ціни на природній газ:

– Розширення ринку: Прогнози цін на природній газ є важливим інструментом для розвитку та розширення міжнародних ринків газу. Ринок природного газу стає все більш глобальним, і точні прогнози допомагають створити стабільні умови для торгівлі газом;

– Розвиток альтернативних джерел енергії: Прогнози допомагають спрямувати інвестиції в розвиток альтернативних джерел енергії, таких як відновлювана енергія, оскільки вони враховують конкурентоспроможність цих джерел в порівнянні з природнім газом;

– Ефективність регулювання: Для забезпечення справедливих цін на газ та захисту прав споживачів важливо мати точні прогнози, які допомагають урядам встановлювати ефективне регулювання цін на енергоресурси;

– Зменшення викидів: Прогнозування цін на природній газ може вплинути на рівень викидів парникових газів, оскільки це може вплинути на вибір палива для енергогенераторів. Зменшення цін на газ може сприяти переходу до більш чистих джерел енергії [4].

Узагальнюючи, прогнозування ціни на природній газ є важливим інструментом для стабільності, розвитку та сталого використання цього важливого джерела енергії.

## **1.2 Розгляд технологій прогнозування даних**

Штучний інтелект (СІ) – це галузь інформатики, яка займається створенням програм та систем, що можуть виконувати завдання, які зазвичай потребують людського інтелекту. Штучний інтелект використовує комп'ютерні алгоритми та моделі для навчання та прийняття рішень на основі великих обсягів даних. Основні аспекти і області штучного інтелекту включають машинне навчання, обробку

природної мови, комп'ютерне бачення, робототехніку, автоматизоване планування та інші [5].

Нейронні мережі – це математичні моделі, які імітують роботу нейронів у мозку та використовуються для вирішення різноманітних завдань штучного інтелекту та машинного навчання. Вони складаються з великої кількості обчислювальних одиниць, які називаються штучними нейронами або просто нейронами, з'єднаних між собою зв'язками. Нейрони в мережі оброблюють вхідні дані, виконуючи різні обчислення та передаючи сигнали через зв'язки.

Нейронні мережі здатні до навчання та адаптації до різних завдань, і вони застосовуються в широкому спектрі дисциплін, включаючи комп'ютерне бачення, обробку природної мови, рекомендаційні системи, автономний транспорт, медицину, фінанси та інші галузі [6].

Машинне навчання (Machine Learning, ML) – це галузь штучного інтелекту, яка зосереджена на розробці алгоритмів та моделей, які дозволяють комп'ютерам вчитися і робити прогнози або приймати рішення на основі даних, без явного програмування.

Машинне навчання є потужним інструментом для аналізу та розуміння даних, а також для автоматизації прийняття рішень у різних галузях, що робить його важливою галуззю в сучасному світі штучного інтелекту.

Мета машинного навчання полягає в навчанні комп'ютерних систем або моделей виконувати завдання без явного програмування, а замість цього базуватися на аналізі та вивченні даних. Основні цілі машинного навчання включають: прогнозування і класифікація, кластеризація, оптимізація рішень, витягнення знань з даних, автоматизація завдань і підсилення інтелекту [7].

Отже, мета машинного навчання полягає в створенні моделей та систем, які можуть навчатися, адаптуватися та вирішувати різноманітні завдання, покращуючи при цьому ефективність та точність рішень.

Машинне навчання може вирішувати широкий спектр завдань і задач у різних галузях. Ось деякі з найпоширеніших задач, які вирішує машинне навчання:

– Класифікація (Classification):

- 1) Розпізнавання об'єктів на зображеннях (наприклад, розпізнавання лиць);
- 2) Визначення категорії або класу для об'єктів (наприклад, класифікація електронних листів на "спам" та "не спам");
- 3) Визначення типу хвороби на основі медичних зображень чи симптомів.

– Регресія (Regression):

- 1) Прогнозування числових значень (наприклад, прогнозування цін на нерухомість);
- 2) Прогнозування часових рядів (наприклад, прогнозування погоди).

– Кластеризація (Clustering):

- 1) Групування подібних об'єктів (наприклад, сегментація аудиторії для маркетингу);
- 2) Виявлення аномалій та виокремлення несподіваних патернів у даних.

– Пошук асоціацій (Association):

- 1) Виявлення зв'язків між різними елементами в наборі даних (наприклад, визначення товарів, які часто купують разом).

– Рекомендації (Recommendation):

- 1) Рекомендація продуктів або контенту користувачам на основі їхніх попередніх дій та вподобань;
- 2) Персоналізовані рекомендації для стрімінгових сервісів (музика, фільми тощо).

– Обробка природної мови (Natural Language Processing, NLP):

- 1) Автоматичний переклад між мовами;
- 2) Аналіз текстів та настрою (визначення настрою тексту);
- 3) Створення голосових асистентів та ботів.

- Автономне керування (Autonomous Control):
  - 1) Управління роботами, автомобілями та дронами без участі людини;
  - 2) Автоматичне виявлення та уникнення перешкод.
- Графічні та відео ігри (Gaming):
  - 1) Реалістична поведінка ворогів та персонажів;
  - 2) Автоматичне генерування ігрового контенту.
- Фінанси та фінансовий аналіз:
  - 1) Прогнозування ринкових цін та трендів;
  - 2) Аналіз ризику та оцінка портфеля.
- Медицина:
  - 1) Діагностика захворювань на основі медичних даних;
  - 2) Персоналізована терапія та лікування.

Машинне навчання широко використовується у багатьох галузях і продовжує розвиватися, вирішуючи нові і більш складні завдання завдяки розвитку технологій та збільшенню обчислювальної потужності [8].

Отже, для задачі прогнозування ціни на природній газ необхідно використовувати інформаційні технології, а саме методи машинного навчання.

### **1.3 Огляд та аналіз відомих аналогів**

Існують вже готові системи пошуку та оцінювання вартості природнього газу, ці системи та сайти також відображають, як змінювалася ціна на природній газ протягом обраного періоду часу, таким чином можна спостерігати за тим, як змінювалася ціна та виявити чи ціна є завищена чи ні.

Bloomberg – це визнаний світовий фінансовий портал, який надає широкий спектр інформації з економіки, фінансів та бізнесу. Заснований в 1981 році, Bloomberg став одним із провідних джерел новин та аналітики для інвесторів та фінансових професіоналів [9].

Платформа пропонує широкий вибір інструментів для відстеження ринків, валют, акцій та товарів. Bloomberg також надає доступ до ексклюзивних інтерв'ю, коментарів експертів та аналітичних звітів, що дозволяє користувачам отримувати найактуальнішу інформацію для прийняття виважених фінансових рішень.

Основною послугою Bloomberg є їхній термінал Bloomberg Professional, який є необхідним інструментом для багатьох фінансових професіоналів по всьому світу. Він надає доступ до ринкових даних, новин, аналітики та інших ресурсів, що допомагають трейдерам та аналітикам приймати управлінські рішення в реальному часі.

Bloomberg, крім того, надає вичерпну інформацію про ціни на природній газ. За допомогою свого фінансового порталу та терміналу Bloomberg Professional, користувачі можуть відстежувати актуальні дані щодо цін на природній газ, аналізувати тренди на ринку енергоресурсів та отримувати спеціалізовану інформацію, необхідну для прийняття рішень в енергетичній сфері. Такий широкий спектр послуг робить Bloomberg важливим ресурсом для професіоналів, які мають інтерес до фінансового ринку та енергетики (рис. 1.1).

Завдяки своєму великому впливу на фінансовий світ, Bloomberg продовжує залишатися ключовим гравцем у галузі інформаційних технологій для фінансів.



The screenshot shows the Bloomberg website interface. At the top, there's a navigation bar with the Bloomberg logo and links for 'Live Now', 'Markets', 'Economics', 'Industries', 'Tech', 'AI', 'Politics', 'Wealth', 'Pursuits', 'Opinion', 'Businessweek', 'Equality', 'Green', 'CityLab', 'Crypto', and 'More'. A search bar on the right contains 'gas prices'. Below the search bar, there are filters for 'Filter By Author', 'Type' (All, Stories, Video, Audio, Graphics, Businessweek), 'Sort' (By Newest, By Relevance, By Oldest), and 'Time' (Any Time, Last 24 Hours, Past Week, Past Month). A 'RESET FILTERS' button is also present.

The main content area displays search results for "gas prices". The first article is titled "Stocks Rise as Wall Street Dials Back Fed Wagers: Markets Wrap" by Rita Nazareth, dated October 10, 2023. The second article is "NATO Member Finland on Alert as Gas Pipe Sabotage Suspected" by Alberto Nardelli, Ewa Krukowska, Ott Tammik, and Kati Pohjanpalo, also dated October 10, 2023. The third article is "Starmer Sparkles in Pitch for Power: The Readout with Julian Harris" by Julian Harris, dated October 10, 2023.

Рисунок 1.1 – Новини з сайту Bloomberg про природній газ

Oil Price – це онлайн-ресурс, що спеціалізується на забезпеченні користувачів інформацією про ціни на нафту та інші енергетичні ресурси. Заснований з метою стеження за динамікою ринку, він надає широкий спектр сервісів для інвесторів, аналітиків та всіх, хто цікавиться енергетикою.

Також, цей ресурс, пропонує користувачам доступ до актуальних цін на нафту, графіків змін цін, аналітики ринку та новин, пов'язаних з енергетичним сектором. Сервіс дозволяє трекувати цінові тенденції, аналізувати вплив подій на глобальний ринок енергії та приймати обґрунтовані фінансові рішення.

Oil Price також може служити цінним ресурсом для тих, хто має інтерес до геополітичних аспектів енергетики, оскільки він надає інсайти та аналіз подій, які впливають на світовий ринок нафти та газу. Усе це робить Oil Price цінним інструментом для отримання об'єктивної та зрозумілої інформації у сфері енергетики.

Oil Price є важливим джерелом для гравців енергетичного ринку, де користувачі можуть відстежувати не лише ціни на нафту, але й інші ключові енергетичні показники, такі як ціни на природній газ та інші види палива. Споживачі отримують можливість глибше розуміти динаміку ринку та аналізувати вплив факторів, що впливають на енергетичний сектор [10].

Крім того, Oil Price часто надає спеціалізовані статті, дослідження та коментарі експертів, що дозволяє користувачам отримувати глибше розуміння подій у світі енергетики. Такий підхід робить платформу не лише інформаційним ресурсом, але і інтелектуальним інструментом для тих, хто бажає залишатися в курсі подій у сфері енергетики та приймати обґрунтовані рішення (рис. 1.2).

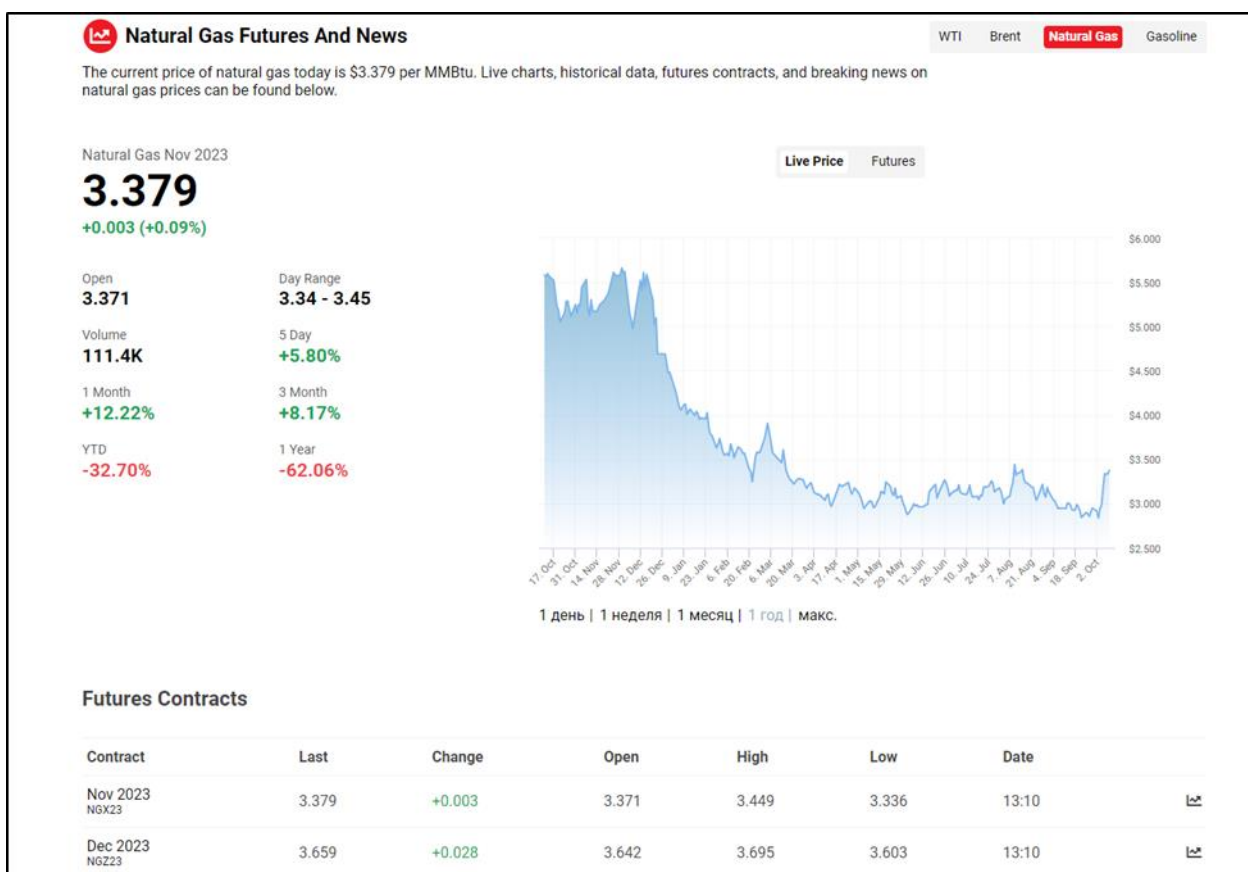


Рисунок 1.2 – Дані та графік ціни на природній газ за останній рік з ресурсу Oil Price

NYMEX (New York Mercantile Exchange) – це одна з провідних товарних бірж у світі, розташована в Нью-Йорку. Заснована в 1882 році, NYMEX є ключовим

центром торгівлі енергетичними товарами, металами та сільськогосподарськими продуктами.

Біржа відома своєю роллю у визначенні цін на нафту, природній газ, золото та інші сировини. На NYMEX відбувається торгівля ф'ючерсами та опціонами, що дозволяє учасникам ринку захищати від цінових ризиків та здійснювати інвестиції.

Ця біржа має великий вплив на глобальні ринки та відіграє ключову роль у формуванні цінових трендів для багатьох базових товарів. NYMEX продовжує залишатися важливим центром для торгівлі та ціноутворення в сфері сировинних ринків [11].

NYMEX (New York Mercantile Exchange) відіграє ключову роль у торгівлі енергетичними товарами та формуванні цін на природній газ на ф'ючерсних ринках. Учасники ринку використовують NYMEX для проведення операцій, здійснення спекуляцій та управління ризиками у секторі енергетики. Динаміка цін на природній газ на Нью-Йоркській товарній біржі може бути відслідкована протягом обраного періоду (рис. 1.3). Ціни на газ на NYMEX вимірюються в доларах США за один мільйон британських теплових одиниць (MMBtu).

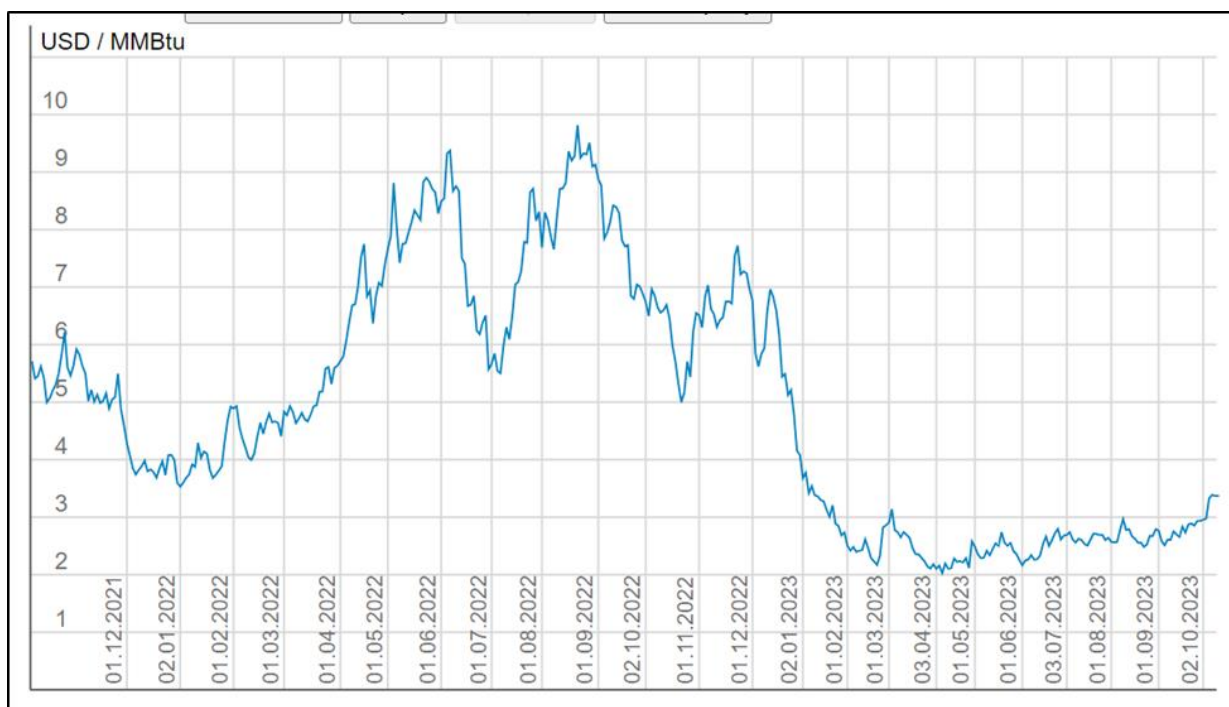


Рисунок 1.3 – Графік динаміки ціни на природній газ за останні два роки взяті з сайту NYMEX

Використовуючи дані з сайту Мінфін та NYMEX про ціну на природній газ було створено датасет “Natural Gas Price Actually 2023”, який в подальшому використовувався у ноутбучі у системі Kaggle під назвою «Natural Gas Price Prediction using Time Series», автором якого є Siddharth Kulkarni [12].

Датасет «Natural Gas Price Actually 2023» містить 2 ознаки та 6897 значень (рис. 1.4). Цей набір даних містить ціни на природній газ за період з 07.01.1997 по 01.10.2023 [13, 14].

```

RangeIndex: 6897 entries, 0 to 6896
Data columns (total 2 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Day                                         6897 non-null   object
1   Price in Dollars per Million Btu          6896 non-null   float64
dtypes: float64(1), object(1)
memory usage: 107.9+ KB

```

Рисунок 1.4 – Кількість ознак та значень в датасеті

У даній розробці для прогнозування використовуються різні моделі регресії та їх налаштування, як прості так і складні, в результаті прогнозування відбувається прогнозування ціни на природній газ на 10 днів.

У даному ноутбучі використано наступні бібліотеки мови Python: NumPy, Pandas, Matplotlib, TensorFlow, Os, Statsmodels, Plotly (рис 1.5).

```

import warnings
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")

import tensorflow as tf
import os
from tensorflow.keras import layers, models
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
import plotly.express as px
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

import numpy as np
from sklearn.model_selection import RandomizedSearchCV
from tensorflow.keras.wrappers.scikit_learn import KerasRegressor
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

import tensorflow as tf

import plotly.graph_objects as go

```

Рисунок 1.5 – завантаженні бібліотеки мови Python

Після підключення бібліотек та завантаження датасету йде підготовка даних. Під час підготовки даних можна побачити вибірку з 5 найсвіжіших дат з цінами (рис. 1.6), вибірку з 5 найдавніших дат (рис. 1.7), період з якого року по який рік містяться дані в датасеті та їх кількість (рис. 1.8) та підготовленні дані під подальшу роботу (рис. 1.9).

|   | Day        | Price in Dollars per Million Btu |
|---|------------|----------------------------------|
| 0 | 2023-10-01 | 2.84                             |
| 1 | 2023-09-30 | 2.93                             |
| 2 | 2023-09-29 | 2.93                             |
| 3 | 2023-09-28 | 2.93                             |
| 4 | 2023-09-27 | 2.89                             |

Рисунок 1.6 – 5 найсвіжіших дат з цінами на природній газ

|      | Day        | Price in Dollars per Million Btu |
|------|------------|----------------------------------|
| 6892 | 1997-01-13 | 4.00                             |
| 6893 | 1997-01-10 | 3.92                             |
| 6894 | 1997-01-09 | 3.61                             |
| 6895 | 1997-01-08 | 3.80                             |
| 6896 | 1997-01-07 | 3.82                             |

Рисунок 1.7 – 5 найстаріших дат з цінами на природній газ

Data is from 7 Jan 1997 to 1 October 2023 with 6897 records.

Рисунок 1.8 – Період з якого по який рік містять дані в датасеті та їхня кількість

|            | gas_price |
|------------|-----------|
| date       |           |
| 1997-01-13 | 4.00      |
| 1997-01-10 | 3.92      |
| 1997-01-09 | 3.61      |
| 1997-01-08 | 3.80      |
| 1997-01-07 | 3.82      |

Рисунок 1.9 – Підготовленні дані під подальшу роботу

## 1.4 Висновки

У даному розділі виконано загальний огляд об'єкта дослідження, детально розглянуто сутність прогнозування ціни на природній газ і визначено, що для досягнення цієї мети найбільш відповідними є технології машинного навчання. Також було проведено огляд відомих аналогів, серед яких виділилися платформи, такі як Bloomberg, Oil Price, NYMEX, а також ноутбуки у системі Kaggle, призначені для прогнозування ціни на природній газ.

Описано та підготовлено для подальшої обробки необхідний датасет, що є важливим етапом для подальших досліджень та аналізу. Відомості, представлені в цьому розділі, стануть основою для подальшого розгортання методології дослідження і розробки моделей прогнозування ціни на природній газ у подальших частинах магістерської кваліфікаційної роботи.

## **2 ВИБІР ОПТИМАЛЬНИХ НАЛАШТУВАНЬ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ**

### **2.1 Опис моделі, яка буде використовуватись для прогнозування даних**

Інформаційні системи та технології на сьогоднішній день стали неодмінною складовою технічної бази усіх організацій.

Інформаційна технологія – це комплекс методів, процесів та інструментів, які використовують для створення, збирання, передачі, пошуку, оброблення та розповсюдження інформації, використовуючи обчислювальну техніку і системи зв'язку. Головна її мета – забезпечення ефективної організації діяльності людей.

Інформаційні системи та технології відіграють ключову роль в сучасному прогнозуванні. Створення прогнозів – це процес прогнозування подій або стану речей у майбутньому на основі наявних даних та аналізу. Використання інформаційних систем і технологій розширює можливості прогнозування і робить його більш точним і ефективним.

Ось деякі способи, які інформаційні системи та технології впливають на процес прогнозування: збір та обробка даних, аналіз даних, розробка моделей, візуалізація даних та реальний час.

Загалом, інформаційні системи та технології в значній мірі полегшують процес прогнозування, допомагаючи організаціям та індивідуалам приймати більш обґрунтовані рішення та планувати майбутнє. Вони використовуються в різних галузях, включаючи економіку, метеорологію, маркетинг, наукові дослідження та інші, сприяючи підвищенню ефективності та точності прогнозування.

Згідно з розділом 1, для вирішення завдання прогнозування ціни на природній газ доцільно використовувати технологію машинного навчання.

Машинне навчання – це галузь штучного інтелекту, яка вивчає розвиток систем, здатних навчатися та покращувати свою продуктивність без явного програмування. В інших словах, це процес, за якого комп'ютерні системи



навчаються на основі даних і досвіду, а потім використовують ці знання для вирішення завдань та прийняття рішень.

Розв'язання будь-якої задачі аналізу та прогнозування даних з використанням інтелектуальних методів машинного навчання, зазвичай включає наступні етапи:

- Збір та підготовка даних: Перший крок – це збір необхідних даних для аналізу. Це можуть бути структуровані або неструктуровані дані. Після збору даних, їх слід очистити, видалити аномалії, вирішити проблеми з відсутніми даними та провести попередню підготовку для подальшого аналізу;

- Вибір моделі: На цьому етапі ви вибираєте тип моделі машинного навчання, який найкраще підходить для вашої задачі. Це може бути, наприклад, лінійна регресія, дерева рішень, нейронні мережі або інші алгоритми;

- Розділення даних: Дані зазвичай розділяються на навчальний набір (для тренування моделі) і тестовий набір (для оцінки продуктивності моделі). Це допомагає перевірити, наскільки добре модель буде працювати на нових даних;

- Тренування моделі: На цьому етапі модель навчається на навчальних даних. Вона адаптується до зразків у даних і старається підібрати параметри для здійснення прогнозів;

- Оцінка моделі: Після тренування моделі її продуктивність оцінюється на тестовому наборі даних. Це допомагає визначити, наскільки добре модель справляється зі своєю задачею;

- Налаштування моделі: Якщо модель не дає задовільних результатів, може бути потрібно внести корективи, включаючи зміну параметрів моделі або вибір іншого типу моделі;

- Використання моделі: Після успішного тренування і оцінки моделі її можна використовувати для прогнозування або аналізу нових даних;

- Підтримка та оновлення: Моделі машинного навчання потребують постійної підтримки та оновлення, оскільки дані та вимоги можуть змінюватися з часом.

Ці етапи представляють загальний підхід до розв'язання задач з використанням машинного навчання і можуть бути адаптовані відповідно до конкретної задачі та потреб користувача [15].

## **2.2 Вибір мови програмування та середовища розробки**

На сьогоднішній день найбільш поширеною мовою програмування для інтелектуального аналізу та прогнозування даних є мова програмування Python.

Python – це мова програмування, яка стала найпопулярнішою та найбільш визнаною серед фахівців у галузі інтелектуального аналізу та прогнозування даних. Її успіх пояснюється рядом переваг, які роблять її ідеальним вибором для цих завдань і відкривають безліч перспектив для подальшого використання.

Однією з ключових переваг Python є його простота та легкість вивчення. Він має зрозумілу синтаксис і багатий набір бібліотек, що робить його доступним навіть для початківців. Python також відомий своєю великою активною спільнотою, яка регулярно розробляє нові бібліотеки та інструменти для аналізу даних.

Іншою перевагою Python є його універсальність. Ця мова програмування може бути використана для різних завдань, включаючи обробку даних, статистику, машинне навчання, роботу з великими наборами даних, візуалізацію і багато інших. Велика кількість спеціалізованих бібліотек, таких як NumPy, pandas, sklearn та TensorFlow, роблять Python ідеальним інструментом для інтелектуального аналізу даних [16].

Python також славиться своєю портативністю, що дозволяє використовувати код на різних операційних системах без змін. Це робить його універсальним і зручним для розробників, які працюють в різних оточеннях.

Ще однією перевагою Python є його активна участь у розвитку технологій штучного інтелекту та машинного навчання. Мова надає розширені інструменти для створення та навчання моделей, а також велику кількість ресурсів та спільноту, яка підтримує і розвиває цей напрям.

Машинне навчання є однією з найбільш актуальних областей в сучасному світі, і Python грає ключову роль в її розвитку. За допомогою Python можна створювати, навчати та вдосконалювати моделі для прогнозування на основі великих обсягів даних. Ця мова програмування також ідеально підходить для роботи з нейронними мережами та глибоким навчанням, що є актуальними напрямками у галузі машинного навчання.

Завдяки своїм перевагам і постійному розвитку, Python має світле майбутнє як основний інструмент для інтелектуального аналізу та прогнозування даних. Його простота і потужність дозволять вченим, аналітикам та розробникам ефективно працювати з даними та робити нові відкриття в цій захоплюючій області.

Зазначені фактори віддзеркалюють популярність мови програмування Python у сфері машинного навчання. Його простота і гнучкість допомагають ефективно працювати над складними алгоритмами з мінімальними зусиллями для їх реалізації.

Після вибору мови програмування стає важливим обрати середовище розробки програмного забезпечення (IDE), де буде відбуватися написання коду.

Середовище розробки програмного забезпечення – це набір інструментів і функцій, які спрощують процес програмування та допомагають в управлінні проектами. Воно автоматизує та полегшує багато аспектів розробки програм, включаючи управління командою та проектами, а також великі завдання програмування, такі як управління конфігурацією.

Середовища розробки програмного забезпечення можуть бути загального призначення або спеціалізовані для конкретних мов програмування чи платформ. Вони надають інструменти для написання, відлагодження та тестування коду, а також можуть включати системи керування версіями, редактори коду, документаційні засоби та інші корисні функції.

Обираючи середовище розробки програмного забезпечення, розробники зазвичай враховують власні потреби, особисті вподобання та специфіку проекту.

Середовища розробки програмного забезпечення (СРПЗ) відіграють ключову роль у спрощенні та оптимізації процесу програмування. Вони дозволяють розробникам ефективно взаємодіяти з кодом, управляти проектами та вдосконалювати якість розробки.

У сучасному світі існує розмаїття СРПЗ, як загального призначення, так і спеціалізованих для конкретних мов програмування чи типів проектів. Вони надають інструменти для розробки, відлагодження та тестування коду. Багато СРПЗ включають системи керування версіями, які полегшують спільну роботу команди, а також редактори коду, що підтримують різноманітні мови програмування.

Обираючи СРПЗ, розробники зважають на власні потреби, тип проекту та особисті вподобання. Деякі середовища, такі як Visual Studio Code, PyCharm або Eclipse, спеціалізуються на конкретних мовах програмування, тоді як інші, наприклад, IntelliJ IDEA, надають розширену функціональність для універсального використання. Вибір правильного СРПЗ може визначити комфорт та ефективність у процесі розробки програмного забезпечення.

Вірний вибір середовища може суттєво покращити продуктивність та якість розробки програмного забезпечення [17].

На сьогоднішній день існує велика кількість середовищ для роботи з мовою програмування Python, включаючи, але не обмежуючись:

- PyCharm – це потужне інтегроване середовище розробки (IDE) від компанії JetBrains, яке спеціалізується на інструментах розробки. PyCharm має багато корисних функцій для розробників Python і надає підтримку для відлагодження, аналізу коду, систем керування версіями та багато іншого;
- Jupyter Notebook – це інтерактивне середовище для розробки, яке дозволяє створювати та обмінюватися документами, що містять живий код, візуалізацію та пояснювальні текстові блоки. Воно дуже популярне серед дослідників даних та аналітиків;

- Visual Studio Code (VS Code) – це безкоштовний та легкий редактор коду, який може бути розширений різними розширеннями для роботи з Python. VS Code популярний завдяки своїй широкій спільноті та активному розвитку;
- Spyder – це середовище розробки, розроблене спеціально для наукових обчислень та аналізу даних. Воно включає в себе багато корисних інструментів для роботи з Python і його бібліотеками;
- Anaconda – це дистрибутив Python, який включає в себе не тільки сам Python, але й багато наукових бібліотек та середовищ для їх розробки;
- Google Colab – це безкоштовне інтерактивне середовище від Google, яке дозволяє виконувати код Python у хмарі та надає доступ до GPU для виконання обчислень у глибокому навчанні;
- Thonny – це легке та інтуїтивно зрозуміле середовище розробки для мови програмування Python. Спеціально створене для початківців, Thonny має інтегрований редактор коду, консоль для виконання команд, а також простий інтерфейс для управління пакетами. Однією з його особливостей є простота налаштування віртуальних середовищ та легкість використання для навчання основам програмування;
- IDLE – це вбудоване середовище розробки, яке постачається разом з Python. Воно просте і добре підходить для початківців;
- Atom – це відкритий та розширюваний редактор коду від GitHub. Завдяки великій кількості розширень, Atom може бути легко адаптований для роботи з Python. Редактор надає розширені можливості, такі як автозавершення коду, відлагодження та розширені можливості налаштувань. Atom також підтримує спільну роботу команди та інтеграцію з системами керування версіями для зручності роботи над проектами;
- Sublime Text – це інший легкий редактор коду, який може бути розширений розширеннями для підтримки Python.

Також існують онлайн середовища для написання коду мовою програмування Python, які дозволяють вам розробляти програми без потреби

встановлення та налаштування середовища на вашому комп'ютері. Деякі з популярних онлайн середовищ включають:

- Google Colab – це вже згадане середовище від Google, яке надає доступ до обчислювальних ресурсів у хмарі для виконання коду Python. Воно особливо корисне для великих обчислень та проектів із штучного інтелекту;

- Repl.it – це онлайн середовище для програмування, яке підтримує не тільки Python, але і багато інших мов програмування. Ви можете створювати та запускати проекти в браузері, а також спільно працювати з іншими користувачами;

- PythonAnywhere – це сервіс, спеціалізований на розробці та запуску програм Python у хмарі. Він надає онлайн IDE та можливість розгортання Python-програм в Інтернеті;

- PyFiddle – це онлайн середовище для написання та виконання коду Python. Воно має простий інтерфейс та дозволяє вам ділитися своїм кодом з іншими користувачами;

- CoderPad – це інтерактивне середовище для спільного написання та виконання коду в реальному часі. Дозволяє працювати не лише з Python, але й з іншими мовами програмування.

- Kaggle Notebook – це онлайн-середовище для розробки та виконання коду Python, спеціалізоване на аналізі даних та машинному навчанні. Kaggle – це платформа для змагань з аналізу даних, де дослідники, аналітики та розробники можуть спробувати свої навички в рішенні різноманітних завдань та конкурсів.

Для виконання даної роботи було вибрано платформу Kaggle. Обираючи Kaggle Notebook для виконання роботи, було враховано ряд переваг, які ця платформа пропонує, особливо для задач аналізу даних та машинного навчання. Ось кілька ключових переваг, які спонукали до цього вибору:

- Обчислювальні ресурси: Kaggle надає велику кількість обчислювальних ресурсів, включаючи доступ до GPU, що є важливим для швидкої розробки та навчання моделей машинного навчання;

- Легка робота з даними: Kaggle має велику кількість наборів даних, і ви можете легко завантажувати, аналізувати та використовувати їх у ваших проектах без необхідності вручну завантажувати дані;
- Спільність та ресурси: Kaggle має активну спільноту користувачів, яка сприяє обміну знаннями, кодом та рішеннями. Ви можете швидко знайти підтримку та поради від інших учасників;
- Зручна інтеграція: Kaggle підтримує Python та всі основні бібліотеки для машинного навчання, що робить його ідеальним інструментом для виконання завдань у сфері аналізу даних та машинного навчання;
- Конкурси та визнання: Ви також можете приєднатися до конкурсів та змагань на Kaggle, що дозволить вам вимірювати свої навички та отримувати визнання [17, 18].

Отже, для розв'язання завдання з прогнозування ціни на природній газ була обрана мова програмування Python, оскільки вона найбільше підходить для використання методів машинного навчання. Як середовище розробки програмного коду на Python було вибрано систему Kaggle IDE Notebooks. Хмарні ресурси для обчислень у безкоштовному режимі виявилися досить достатніми для виконання цієї роботи.

### **2.3 Визначення виду машинного навчання**

Класичне машинне навчання (Classical Machine Learning) дійсно базується на класичних статистичних алгоритмах та методах і призначене для розв'язання завдань, пов'язаних з прийняттям рішень на основі даних.

Розрізняють декілька основних видів класичного машинного навчання:

- Контрольоване навчання (навчанням з учителем);
- Неконтрольоване навчання (навчання без учителя);
- Навчання з підкріпленням;
- Нейронні мережі і глибоке навчання;

– Інші види машинного навчання.

Ці види машинного навчання використовуються в залежності від конкретного завдання та характеристик даних, які необхідно аналізувати та моделювати.

Нейронні мережі і глибоке навчання (Neural Networks and Deep Learning) – це підгрупа машинного навчання, яка базується на інсульті структури та функціонування людського мозку. Головною характеристикою глибокого навчання є використання глибоких нейронних мереж, які складаються з багатьох шарів (глибших структур) [19].

Глибоке навчання стало надзвичайно популярним і зробило значні досягнення в багатьох галузях завдяки здатності вивчати складні залежності у даних. Воно є основою для багатьох інноваційних додатків і технологій в сучасному світі.

Отже, було визначено, що задача прогнозування ціни на природній газ відноситься до такого виду машинного навчання як, глибоке навчання. Завдання прогнозування ціни на природній газ дійсно може бути вирішене за допомогою глибокого навчання, включаючи використання рекурентних нейронних мереж (наприклад, LSTM) для аналізу часових рядів.

Часові ряди є важливим інструментом для моделювання та прогнозування цін на природній газ, оскільки вони враховують залежність цін від часу та попередніх значень. Використання глибокого навчання і LSTM може допомогти автоматично виявляти складні залежності в часових рядах і забезпечити точні прогнози цін.

Однак важливо враховувати, що успішність моделі залежить від якості даних, правильного підбору гіперпараметрів та методів оцінки моделі. Також враховуйте, що ціни на природній газ можуть бути впливати багатьма факторами, і прогнози можуть бути важким завданням через їх варіабельність.

Попереднє навчання моделі на історичних даних і постійний моніторинг та оновлення моделі є важливими кроками для досягнення найкращих результатів в задачі прогнозування цін на природній газ.



## 2.4 Вибір та опис моделей машинного навчання

Існує багато моделей машинного навчання, але для реалізації даного завдання буде наведено та використано, такі моделі як ARIMA та LSTM. Обидві ці моделі є добрими виборами для аналізу часових рядів і прогнозування цін.

ARIMA (AutoRegressive Integrated Moving Average) – це статистична модель для аналізу та прогнозування часових рядів. Модель ARIMA враховує авторегресію (AutoRegressive), рухоме середнє (Moving Average) та інтегровану складову (Integrated) часового ряду, що дозволяє моделювати тренди та сезонні зміни.

Основні компоненти ARIMA включають:

- Авторегресія (AR): Ця складова моделює залежність поточного значення від попередніх значень у часовому ряді. Параметр AR показує, скільки попередніх значень потрібно враховувати;
- Рухоме середнє (MA): Ця складова моделює залежність поточного значення від попередніх значень помилок моделі (шуму). Параметр MA показує, скільки попередніх помилок потрібно враховувати;
- Інтегрована складова (I): Ця складова відображає рівень диференціації (різниці) часового ряду, необхідний для зроблення ряду стаціонарним. Це дозволяє враховувати зміни тренду та сезонності у ряді.

Процес побудови моделі ARIMA включає в себе вибір параметрів AR, MA та I, а також оцінку моделі на основі історичних даних. ARIMA дозволяє прогнозувати майбутні значення в часовому ряді, враховуючи попередні значення та залежності.

ARIMA може бути корисним для прогнозування цін на природній газ, особливо якщо у ваших даних є сильні сезонні залежності та тренди. Для вирішення задачі ARIMA вимагає правильного вибору параметрів та аналізу даних для визначення оптимальної моделі.

Застосування ARIMA важливо не лише для прогнозування, але й для стратегічного управління ризиками та аналізу великого обсягу даних, що є важливими аспектами в сучасному енергетичному та фінансовому секторах.

Також, враховуючи зростаючу значимість сфери енергетики, особливо у контексті сталого розвитку, ефективне використання ARIMA може відігравати ключову роль у прогнозуванні та оптимізації виробництва енергії на основі аналізу ринкових тенденцій [20].

LSTM (Long Short-Term Memory) – це тип рекурентних нейронних мереж, який добре підходить для аналізу та прогнозування часових рядів та послідовних даних. LSTM є однією з розширених архітектур штучних нейронних мереж, яка була спеціально розроблена для вирішення проблеми "зниклого градієнту" в класичних рекурентних нейронних мережах.

Основні особливості LSTM включають:

- Збереження короткострокової та довгострокової пам'яті: LSTM може зберігати інформацію на короткостроковий та довгостроковий період, дозволяючи враховувати важливі залежності в часових рядах;
- Ворота: LSTM використовує ворота, які дозволяють регулювати потік інформації через нейрони. Це включає в себе ворота забування, ворота входу та ворота виходу, які керують, яка інформація повинна бути збережена або забута;
- Здатність до вивчення складних залежностей: Завдяки своїй архітектурі LSTM може вивчати складні залежності в часових рядах, включаючи сезонність, тренди та інші закономірності;
- Здатність до вирішення проблеми довгих залежностей: LSTM розроблена, щоб вирішити проблему "зниклого градієнту", яка виникає в класичних рекурентних нейронних мережах та обмежує їхню здатність враховувати довгострокові залежності в даних.

Застосування LSTM важливо не лише для прогнозування, але й для стратегічного управління ризиками та аналізу великого обсягу даних, що є важливими аспектами в сучасному енергетичному та фінансовому секторах. Також, враховуючи зростаючу значимість сфери енергетики, особливо у контексті сталого розвитку, ефективне використання LSTM може відігравати ключову роль у прогнозуванні та оптимізації рішень в цьому секторі.

LSTM добре підходить для задач прогнозування цін на природній газ, оскільки вона може враховувати складність і довгострокові залежності у часових рядах цін. Використання LSTM вимагає навчання на історичних даних та підбору параметрів для досягнення найкращих результатів в прогнозуванні цін [21].

## 2.5 Вибір та опис бібліотек Python

У світі програмування мовою Python існує надзвичайно багато різноманітних бібліотек. У рамках даної магістерської кваліфікаційної роботи будуть використовуватися такі бібліотеки:

Pandas – це потужна бібліотека для маніпулювання та аналізу даних в мові програмування Python. Вона надає структури даних і функції для ефективної роботи з табличними даними, читання та запису даних з різних джерел, агрегації, фільтрації, обчислення та візуалізації даних. Pandas широко використовується в областях наукового дослідження, аналізу даних та розробки програм для обробки і аналізу даних [22].

NumPy (Numerical Python) – це основна бібліотека для наукових обчислень у мові програмування Python. Вона надає підтримку для обробки багатовимірних масивів даних, включаючи вектори та матриці, і набір функцій для виконання операцій з цими масивами. NumPy є основною складовою великої кількості інших бібліотек для наукових обчислень, таких як SciPy, pandas, Matplotlib, і інших.

Matplotlib – це бібліотека для створення візуалізацій та графіків у мові програмування Python. Вона дозволяє створювати різноманітні типи графіків, включаючи лінійні графіки, стовпчаті діаграми, коліщаті діаграми, розсіювальні графіки, графіки з контурними лініями, гістограми та інші. Matplotlib є дуже потужним і розширюваним інструментом для візуалізації даних, і вона широко використовується в областях наукового дослідження, аналізу даних, статистики, машинного навчання та інших.

TensorFlow є однією з найпопулярніших бібліотек для машинного навчання та глибокого навчання та використовується в багатьох галузях, включаючи

дослідження, розробку продуктів та наукові дослідження. Існує дві основні версії TensorFlow: TensorFlow 1.x і TensorFlow 2.x. TensorFlow 2.x включає в себе більше високорівневий API та більше спрощений спосіб визначення та навчання моделей за допомогою Keras.

Os – це модуль в мові програмування Python, який надає функції для взаємодії з операційною системою (OS), такі як створення, видалення та переміщення файлів і каталогів, отримання і зміна поточної робочої директорії, виконання команд системи та багато інших операцій.

Sklearn – це відкрите програмне забезпечення для машинного навчання та аналізу даних, розроблене для мови програмування Python. Ця бібліотека надає інструменти та алгоритми для вирішення різних завдань машинного навчання, включаючи класифікацію, регресію, кластеризацію, врядування, виявлення аномалій та інші [23].

Statsmodels – це бібліотека для мови програмування Python, яка надає інструменти для статистичного моделювання та аналізу даних. Вона спрямована на виконання різних статистичних аналізів, включаючи лінійну та нелінійну регресію, вибірковий аналіз, аналіз дисперсії, часові ряди та багато інших статистичних методів.

Plotly – це бібліотека для створення інтерактивних графіків та візуалізації даних в мові програмування Python. Вона дозволяє розробникам та аналітикам створювати різноманітні види графіків, включаючи лінійні графіки, діаграми розсіювання, гистограми, графіки барів, карти, тривимірні графіки, графіки пустих (ієрархічні) та багато інших. Головною особливістю Plotly є можливість взаємодії з графіками вживу.

Модуль `statsmodels.tsa.stattools` в бібліотеці `statsmodels` містить функції для статистичного аналізу часових рядів. Цей модуль надає інструменти для виконання різних статистичних тестів та аналізу часових рядів, щоб оцінити їхню стаціонарність, автокореляцію, сезонність та інші характеристики.

Модуль `statsmodels.graphics.tsaplots` в бібліотеці `statsmodels` містить функції для створення різних графіків та візуалізації часових рядів. Ці функції допомагають

аналізувати та візуалізувати різні характеристики часових рядів, такі як автокореляція, часткова автокореляція, густини спектру та інші.

## **2.6 Висновки**

У даному розділі було визначено оптимальні налаштування для вирішення задачі, де детально розглянуто опис обраної моделі для прогнозування даних. Також було вибрано мову програмування та середовище розробки, а також визначено вид машинного навчання для подальших досліджень. Окрім цього, у роботі детально проаналізовано моделі ARIMA та LSTM, які були обрані для прогнозування даних, з описом їхньої структури та основних принципів функціонування. У контексті даного проекту також було визначено, що мова програмування Python та середовище розробки Kaggle є оптимальними для реалізації вибраних моделей та вирішення визначеної задачі.

Цей етап є критичним для продовження роботи над подальшим розробленням інформаційної технології аналізу та прогнозування ціни на природній газ. Визначення оптимальних налаштувань та вибір відповідних моделей машинного навчання визначають успішність прогнозування та ефективність розробленої системи.

## 3 РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЦІНИ НА ПРИРОДНІЙ ГАЗ

### 3.1 Опис ознак та зчитування вхідного набору даних

Вхідний датасет «Natural Gas Price Actually 2023» містить наступні атрибути (ознаки):

- Day (дата);
- Price in Dollars per Million Btu (ціна на природній газ) (рис. 3.1).

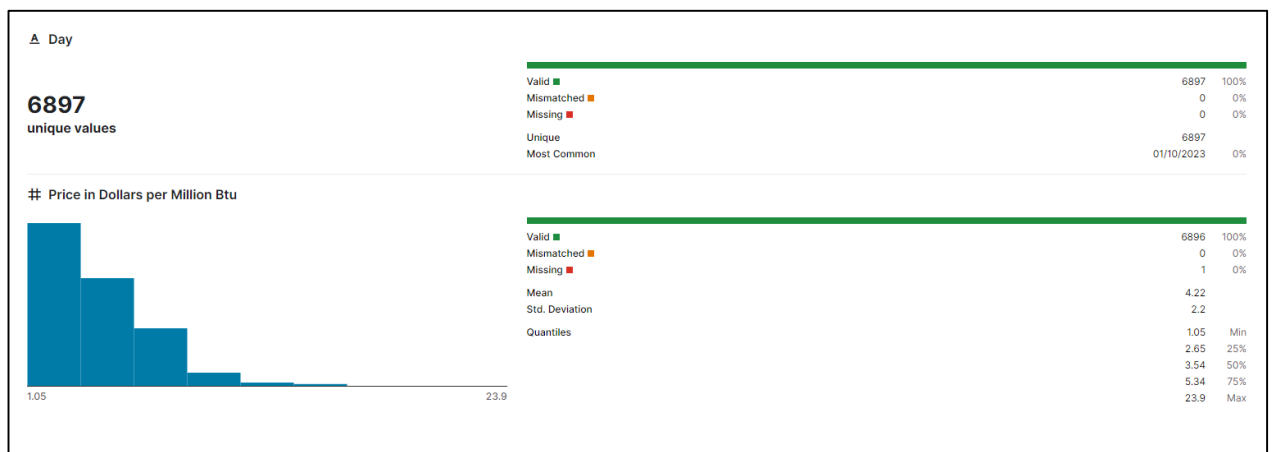


Рисунок 3.1 – Статистичні дані вхідного датасету

Першим етапом у розв'язанні будь-якої задачі машинного навчання є імпорт необхідних бібліотек і зчитування набору даних. Цей етап дуже важливий, оскільки правильний вибір бібліотек і коректне завантаження даних готує фундамент для подальших кроків у роботі з задачею машинного навчання.

Зазвичай, для роботи з задачами машинного навчання використовують популярні бібліотеки, такі як Python та інші залежно від поставленої задачі.

Приклад імпортування бібліотек наведено на рисунку 3.2.

```

import warnings
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")

import tensorflow as tf
import os
from tensorflow.keras import layers, models
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
import plotly.express as px
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

import numpy as np
from sklearn.model_selection import RandomizedSearchCV
from tensorflow.keras.wrappers.scikit_learn import KerasRegressor
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

import tensorflow as tf

import plotly.graph_objects as go

```

Рисунок 3.2 – Імпорт бібліотек, які будуть використовуватись на різних етапах розв'язання задачі

Зчитування вхідного набору даних відбувається за допомогою бібліотеки pandas та її методу `read_csv`. Приклад зчитування даних та вивід загальної інформації по кожному набору даних, за допомогою команди `info()`, відображено на рисунках 3.3 – 3.5.

```

data = pd.read_csv("../input/natural-gas-price-actually-2023/ngpf_data_actually.csv")
data.info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6897 entries, 0 to 6896
Data columns (total 2 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   Day                                         6897 non-null   object
 1   Price in Dollars per Million Btu         6896 non-null   float64
dtypes: float64(1), object(1)
memory usage: 107.9+ KB

```

Рисунок 3.3 – Фрагмент коду для зчитування набору даних з загальною інформацією по кожній ознаці набору даних.

```
data['Day'] = pd.to_datetime(data['Day'], format = "%d/%m/%Y")
data.head(5)
```

|   | Day        | Price in Dollars per Million Btu |
|---|------------|----------------------------------|
| 0 | 2023-10-01 | 2.84                             |
| 1 | 2023-09-30 | 2.93                             |
| 2 | 2023-09-29 | 2.93                             |
| 3 | 2023-09-28 | 2.93                             |
| 4 | 2023-09-27 | 2.89                             |

Рисунок 3.4 – Фрагмент коду для відображення вірного формату дати з фрагментом 5 найсвіжіших зчитаних даних

```
data.tail(5)
```

|      | Day        | Price in Dollars per Million Btu |
|------|------------|----------------------------------|
| 6892 | 1997-01-13 | 4.00                             |
| 6893 | 1997-01-10 | 3.92                             |
| 6894 | 1997-01-09 | 3.61                             |
| 6895 | 1997-01-08 | 3.80                             |
| 6896 | 1997-01-07 | 3.82                             |

Рисунок 3.5 – Фрагмент коді для відображення 5 найстаріших зчитаних даних

З рисунку 3.3 видно, що наш набір даних має 6897 значень, ознака “Day” має тип object, а ознака “Price in Dollars per Million Btu”. Спостерігається одне пусте значення в ознаці “Price in Dollars per Million Btu”, яке буде заповнено в подальшій роботі (рис. 3.6).

Data is from 7 Jan 1997 to 1 October 2023 with 6897 records.

Рисунок 3.6 – Фрагмент зображення на якому відображено з якої по яку дату відібрані дані та яку кількість значень має датасет

За допомогою команди `data.rename()`, зробимо перейменування стовпців. Далі за допомогою команди `data.set_index()` встановимо стовпець “data” як індекс.



За допомогою команди `tail()`, відобразимо 5 найстаріших підготовлених даних (рис. 3.7, 3.8).

```
# Renaming columns
data = data.rename({'Day': 'date', 'Price in Dollars per Million Btu': 'gas_price'}, axis = 1)

# Setting Date as index
data = data.set_index('date')
```

Рисунок 3.7 – Фрагмент коду для перейменування стовпців та встановлення індексу

| data.tail(5) |           |
|--------------|-----------|
|              | gas_price |
| date         |           |
| 1997-01-13   | 4.00      |
| 1997-01-10   | 3.92      |
| 1997-01-09   | 3.61      |
| 1997-01-08   | 3.80      |
| 1997-01-07   | 3.82      |

Рисунок 3.8 – Фрагмент коду з відображенням 5 найстаріших підготовлених даних

Отже, на даному етапі наш набір даних пройшов підготовку для подальшого використання та задовольняє усі вимоги.

### 3.2 Очищення даних

Очищення даних є важливим етапом в підготовці даних для задач машинного навчання. Цей процес включає в себе ряд дій для обробки та підготовки даних перед їх використанням для навчання моделей машинного навчання. Основні кроки очищення даних в машинному навчанні включають таке:

- Виявлення та обробка пропущених значень: Визначення та обробка відсутніх значень в наборі даних, включаючи їх заповнення або видалення, в залежності від контексту;
- Обробка викидів: Виявлення і обробка викидів або аномальних значень, що можуть вплинути на моделювання. Це може включати видалення викидів або застосування методів для їх зменшення впливу;
- Кодування категоріальних змінних: Конвертація категоріальних змінних у числовий формат, наприклад, за допомогою методів як one-hot encoding або заміни категорій числовими індексами;
- Масштабування змінних: Масштабування числових змінних, щоб зробити їх масштаб більш однорідним, наприклад, за допомогою стандартизації;
- Видалення зайвих ознак: Видалення змінних, які не приносять користі у моделюванні або можуть спричинити шум;
- Обробка незбалансованих даних: Вирішення проблеми незбалансованих класів, яка виникає в задачах класифікації;
- Розбиття набору даних: Розбиття даних на тренувальний, тестовий та, можливо, валідаційний набори для оцінки та тестування моделі;
- Видалення дублікатів: Виявлення та видалення дублікатів записів у наборі даних;
- Обробка часових рядів (якщо відповідає контексту): Якщо ви працюєте з часовими рядами, можуть виникати додаткові завдання, такі як витягування ознак, згортка даних та інші обробні процедури.

Очищення даних спрямоване на створення даних, які можуть бути використані для тренування моделей машинного навчання. Цей процес допомагає зменшити шум у даних та підготувати їх для подальших операцій, таких як побудова моделі, тестування та валідації.

За допомогою команди `data.isnull().sum()`, було виявлено, що одна з ознак не містить одного значення (рис. 3.9).

```
print(data.isnull().sum())  
  
gas_price    1  
dtype: int64
```

Рисунок 3.9 – Фрагмент коду з виявленою ознакою та кількістю відсутніх значень у ній

Було прийнято рішення за допомогою команди `data.fillna(method='pad')`, яка використовує метод заповнення пропущених значень в даних та методу “pad”, який вказує на заповнення пропущених значень за допомогою значень з попереднього запису в стовпці. Цей метод також відомий як метод заповнення вперед (forward fill). Таким чином, пусте значення було заповнено ціною на природній газ з попереднього дня та за допомогою команди `data.isnull().sum()`, було виведено, що нульові дані відсутні та можна продовжувати подальшу роботу (рис. 3.10).

```
# There is one missing value in the data, we fill it with  
previous day's price.  
  
data = data.fillna(method = 'pad')  
print(data.isnull().sum())  
  
gas_price    0  
dtype: int64
```

Рисунок 3.10 – Фрагмент коду з використання методу для заповнення пустих значень

### 3.3 Розвідувальний аналіз даних

Розвідувальний аналіз даних (EDA) – це процес дослідження та аналізу набору даних з метою зрозуміти їхню структуру, основні характеристики та взаємозв'язки між ознаками. EDA включає в себе використання методів візуалізації та описової статистики для вивчення даних без застосування складних моделей.

Основні цілі розвідувального аналізу даних включають таке:

- Виявлення патернів;
- Виявлення аномалій;
- Зрозуміння структури даних;
- Виявлення взаємозв'язків;
- Генерація гіпотез;
- Підготовка даних.

Розвідувальний аналіз даних допомагає аналітикам та науковцям отримати загальне уявлення про дані та сформулювати гіпотези для подальшого дослідження [24]. Під час розвідувального аналізу даних, аналітики можуть також використовувати інструменти, які дозволяють виявляти розподіли, аналізувати взаємозв'язки за допомогою кореляційних матриць та графіків, а також використовувати техніки групування для виділення класів або кластерів в наборі даних. Це допомагає отримати більше інсайтів та поглибити розуміння особливостей досліджуваних даних.

Це важливий крок в аналізі даних, оскільки він допомагає зорієнтуватися в даних та визначити, які подальші кроки необхідно вжити для досягнення конкретних цілей аналізу.

Бібліотека `pandas` у мові програмування `Python` визнана своєю високою ефективністю та потужністю, роблячи її однією з найшвидших та найбільш корисних інструментів для проведення розвідувального аналізу даних (EDA) [25]. Її широкий функціонал дозволяє аналітикам та науковцям ефективно виконувати завдання, пов'язані з вивченням структури даних, виявлення патернів, аналізу

аномалій та встановлення взаємозв'язків між різними ознаками. Застосування бібліотеки `pandas` стає ключовим етапом у використанні розвідувального аналізу даних для отримання глибокого розуміння набору даних та формулювання перспективних гіпотез для подальших досліджень.

Використовуючи імпортовану бібліотеку `Plotly`, побудуємо лінійну діаграму з ретельним аналізом (рис. 3.11, 3.12). `Plotly` визначається своєю здатністю до ефективної візуалізації даних в мові програмування `Python`, надаючи можливості не лише для швидкої побудови графіків, але й для налаштування їхнього вигляду та інтерактивності, забезпечуючи високий рівень інформативності [26].

```
fig = px.line(data, title = 'Natural Gas Spot Prices', template = 'plotly_dark')  
fig.show()
```

Рисунок 3.11 – Фрагмент коду для побудови лінійної діаграми

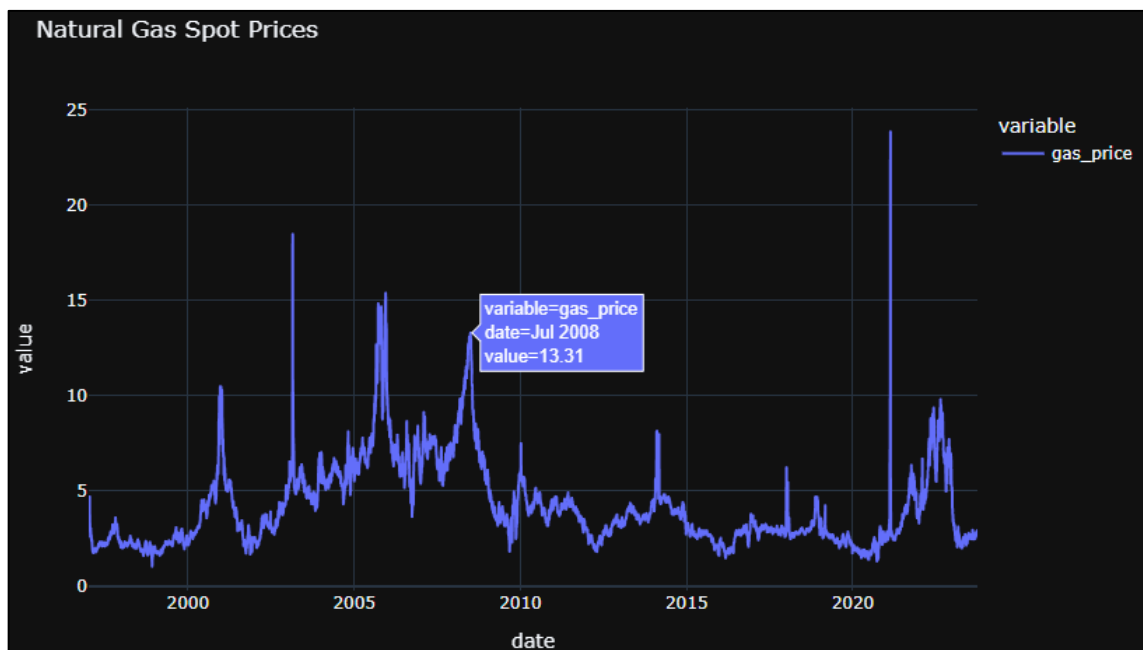


Рисунок 3.12 – Лінійна діаграма

Також побудуємо гістограму за допомогою функції `“px.histogram”` з бібліотеки `Plotly Express` (рис. 3.13, 3.14).

```
fig = px.histogram(data, x = "gas_price", template = 'plotly_dark')
fig.show()
```

Рисунок 3.13 – Фрагмент коду для побудови гистограми

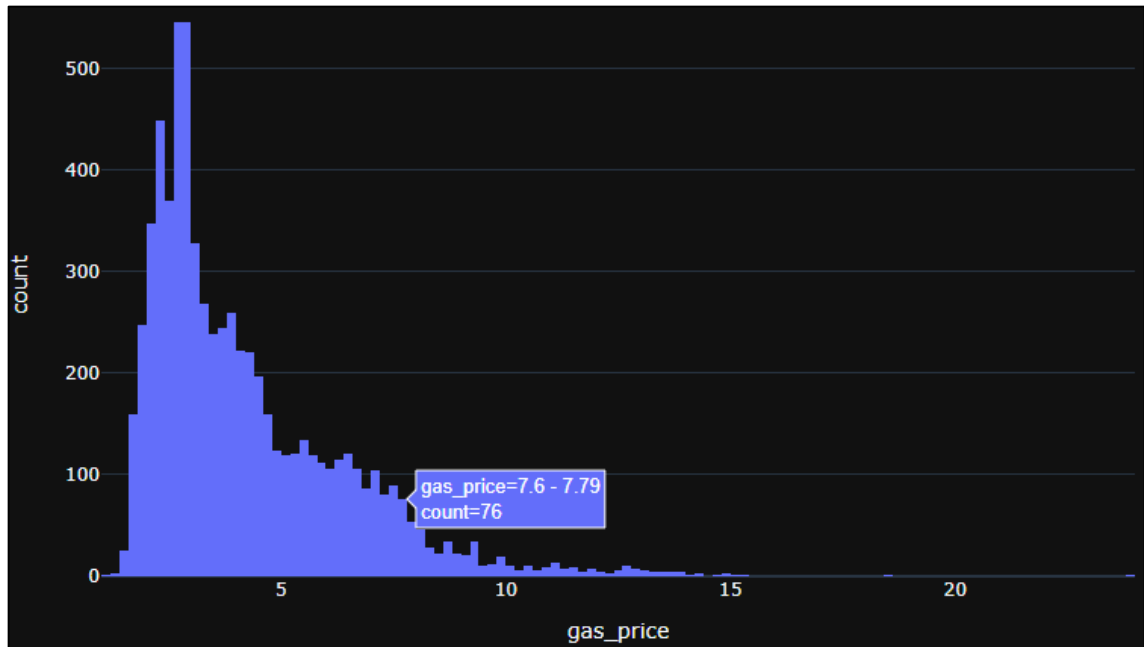


Рисунок 3.14 – Гістограма

Наступним кроком роботи буде перевірка часового ряду на стаціонарність.

Перевірка часового ряду на стаціонарність важлива у аналізі часових рядів, оскільки стаціонарність може вказувати на стабільність властивостей ряду в часі. Вона є передумовою для застосування багатьох методів аналізу та прогнозування часових рядів.

Часовий ряд називається стаціонарним, якщо його статистичні властивості залишаються постійними з часом. Це означає, що середнє значення, дисперсія і інші характеристики ряду залишаються сталими або не залежать від часу.

Існує два основних методи перевірки стаціонарності часових рядів: побудова ковзаючої статистики та тест Дікі-Фуллера.

Побудова ковзаючої статистики (Moving Statistics): Цей метод включає обчислення статистичних метрик, таких як середнє значення та дисперсія, на основі

підвибірок (вікон) даних, які ковзаються вздовж часового ряду. Якщо ці метрики змінюються значно з часом, це може бути ознакою нестационарності [27].

Тест Дікі-Фуллера (Dickey-Fuller Test): Це статистичний тест, який перевіряє гіпотезу про стаціонарність часового ряду. Тест Дікі-Фуллера визначає, чи існують тренди або сезонні компоненти, які роблять часовий ряд нестационарним. Результат тесту може вказати на стаціонарність або нестационарність ряду [28].

Обидва методи є корисними для оцінки стаціонарності часового ряду, і їх можна використовувати як окремо, так і разом для підтвердження результатів. Тест Дікі-Фуллера є широко використовуваним інструментом для перевірки стаціонарності та визначення, чи потрібні додаткові перетворення даних перед моделюванням. У тесті Дікі-Фуллера нульова гіпотеза виглядає наступним чином: "Часовий ряд є нестационарним." Це означає, що якщо результати тесту Дікі-Фуллера демонструють статистичну значущість (якщо р-значення нижче порогового рівня значущості), то ми маємо підстави відкинути нульову гіпотезу. Тобто, ми можемо вважати, що часовий ряд є стаціонарним (рис. 3.15 – 3.17).

```
def test_stationarity(timeseries):

    # Determining rolling statistics
    rolmean = timeseries.rolling(25).mean()
    rolstd = timeseries.rolling(25).std()

    # Plot rolling statistics:
    plt.figure(figsize = (20,10))
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    # Perform Dickey-Fuller test:
    print('Results of Dickey-Fuller Test:')
    dfctest = adfuller(timeseries, autolag='AIC')
    dfcoutput = pd.Series(dfctest[0:4], index=['Test Statistic', 'p-value', '#Lags Used', 'Number of Observations Used'])
    for key, value in dfctest[4].items():
        dfcoutput['Critical Value (%s)' % key] = value

    print(dfcoutput)

    # Check for stationarity
    p_value = dfctest[1]
    significance_level = 0.05
    if p_value <= significance_level:
        print("Ряд є стаціонарним (p-value <= 0.05)")
    else:
        print("Ряд не є стаціонарним (p-value > 0.05)")
```

+ Code + Markdown

test\_stationarity(data)

Рисунок 3.15 – Фрагмент коду для тестування часового ряду на стаціонарність

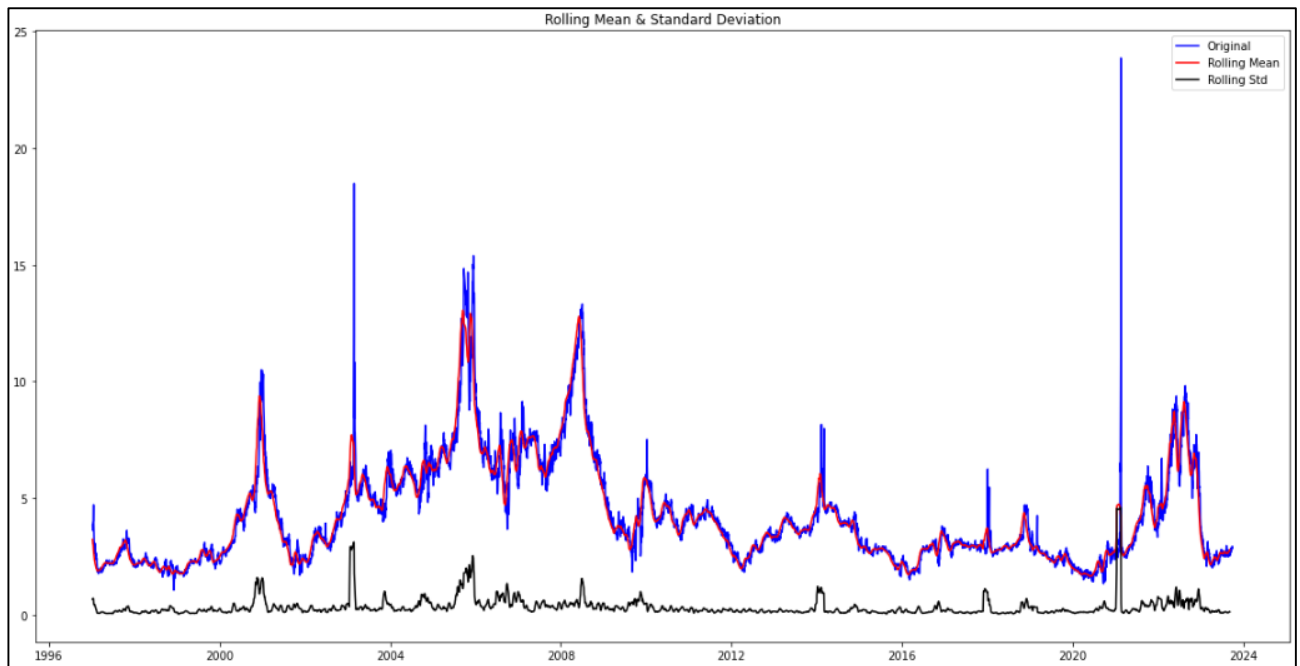


Рисунок 3.16 – Графік перевірки часового ряду на стаціонарність

```
Results of Dickey-Fuller Test:
Test Statistic           -4.037120
p-value                  0.001228
#Lags Used               8.000000
Number of Observations Used 6888.000000
Critical Value (1%)     -3.431300
Critical Value (5%)    -2.861960
Critical Value (10%)   -2.566993
dtype: float64
Ряд є стаціонарним (p-value <= 0.05)
```

Рисунок 3.17 – Результат перевірки часового ряду на стаціонарність

Статистика тесту показує, наскільки точно спостережувані дані відповідають очікуваному розподілу згідно з нульовою гіпотезою цього статистичного тесту.

Статистика тесту (Test Statistic) < критичне значення (Critical Value) & p-value < 0,05. Тому ми відкидаємо нульову гіпотезу. Та доходимо до висновку, що дані стаціонарні.

Наступним етапом буде оцінка та усунення тенденцій. До простих методів зменшення тренду відносяться: ковзне середнє (Moving Average) та експоненціальне зважене ковзне середнє (Exponential Weighted Moving Average). Прості методи зменшення тренду не працюють із високою сезонністю. Тому, будем



використовувати такі методи, як розбіжність (Differencing) та декомпозиція (Decomposition).

**Розбіжність (Differencing):** Розбіжність – це метод, який включає віднімання значень часового ряду на попередньому кроці часу з поточного значення для того, щоб видалити тренд або локальні залежності від часу. Це може бути корисно для зроблення часового ряду більш стаціонарним, особливо якщо ви спостерігаєте тренд або сезонність.

**Декомпозиція (Decomposition):** Декомпозиція полягає в розбитті часового ряду на складові частини, такі як тренд, сезонність та залишки. Тренд представляє загальну тенденцію часового ряду з часом. Сезонність відображає повторювані коливання або патерни в часовому ряді. Залишки є частиною ряду, яка не може бути пояснена трендом або сезонністю. Декомпозиція допомагає краще розуміти складові ряду та може покращити його стаціонарність.

Обидва ці методи часто використовуються для підготовки часових рядів до подальшого аналізу та моделювання. Після розбіжності та/або декомпозиції можна використовувати статистичні моделі, такі як ARIMA (Autoregressive Integrated Moving Average), для прогнозування майбутніх значень часового ряду (рис. 3.18 – 3.24).

```
Exponential Weighted Moving Average

ts_sqrt = np.sqrt(data)
expwighted_avg = ts_sqrt.ewm(halflife = 25).mean()

ts_sqrt_ewma_diff = ts_sqrt - expwighted_avg
test_stationarity(ts_sqrt_ewma_diff)
```

Рисунок 3.18 – Фрагмент коду для простого методу зменшення тренду (Exponential Weighted Moving Average)

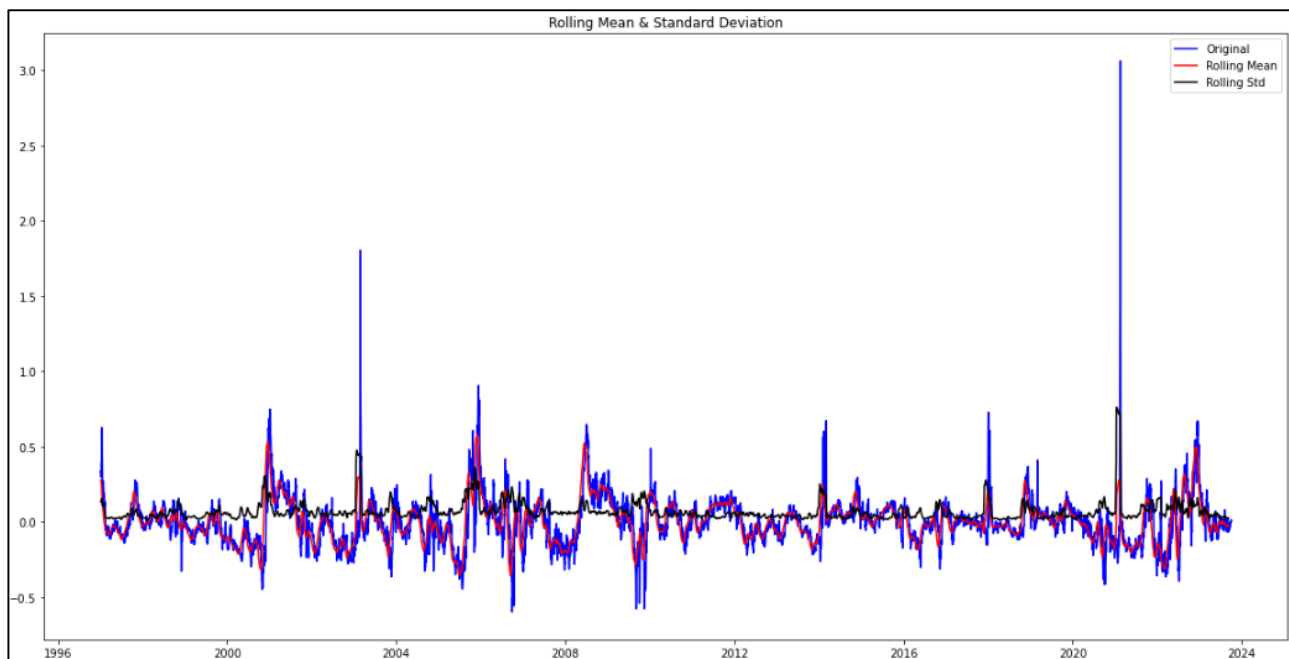


Рисунок 3.19 – Графік використання методу “Exponential Weighted Moving Average”

```

Results of Dickey-Fuller Test:
Test Statistic          -1.128488e+01
p-value                 1.426164e-20
#Lags Used              1.000000e+01
Number of Observations Used  6.886000e+03
Critical Value (1%)     -3.431300e+00
Critical Value (5%)     -2.861960e+00
Critical Value (10%)    -2.566993e+00
dtype: float64
Ряд є стаціонарним (p-value <= 0.05)

```

Рисунок 3.20 – Результат використання методу “Exponential Weighted Moving Average”

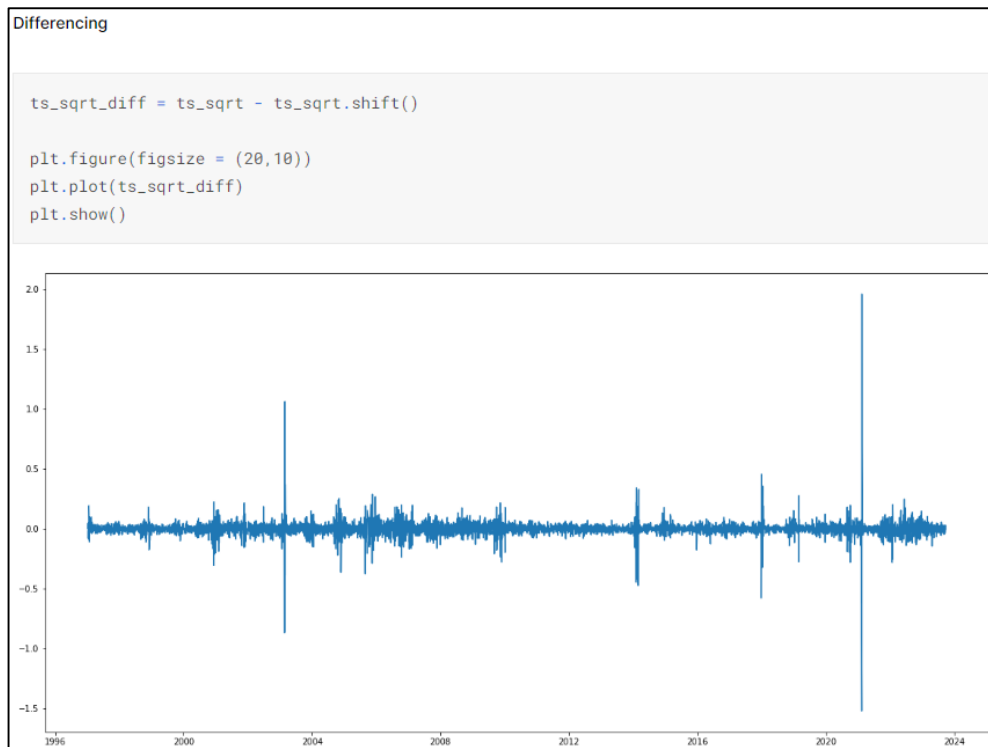


Рисунок 3.21 – Фрагмент коду та відображення графіку різниць за допомогою використання методу “Differencing”

```
ts_sqrt = np.sqrt(data)
ts_sqrt_diff = ts_sqrt - ts_sqrt.shift()
ts_sqrt_diff.dropna(inplace = True)
test_stationarity(ts_sqrt_diff)
```

Рисунок 3.22 – Фрагмент коду для підготовки та аналізу часового ряду

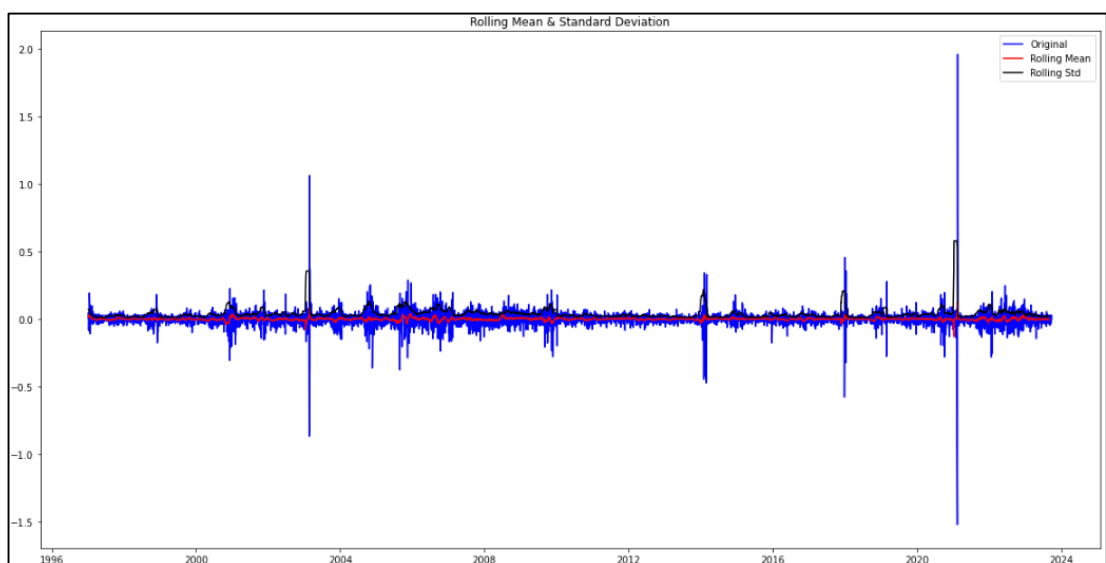


Рисунок 3.23 – Графік використання методу розбіжності (Differencing)

```

Results of Dickey-Fuller Test:
Test Statistic      -26.758561
p-value             0.000000
#Lags Used          11.000000
Number of Observations Used  6884.000000
Critical Value (1%)  -3.431300
Critical Value (5%)  -2.861960
Critical Value (10%) -2.566994
dtype: float64
Ряд є стаціонарним (p-value <= 0.05)

```

Рисунок 3.24 – Результат використання методу розбіжності (Differencing)

У підсумку, отримуємо, що статистика тесту (Test Statistic) < критичне значення (Critical Value), а також менше відхилень у середньому та стандартному. Таким чином, можна дійти висновку, що це ідеальний стаціонарний часовий ряд.

### 3.4 Моделювання даних

Моделювання даних – це процес створення математичних або статистичних моделей, які відображають реальні дані або процеси для здійснення аналізу, прогнозування або розв'язання практичних завдань. Моделі можуть бути використані для дослідження взаємозв'язків між різними змінними, а також для створення сценаріїв або передбачень майбутніх подій [29].

Існують різні методи та підходи до моделювання даних, зокрема:

- Лінійна регресія;
- Класифікація;
- Часові ряди;
- Машинне навчання;
- Байєсівське моделювання;
- Симуляція;
- Нейромережі.

Кожен з цих методів використовується в залежності від конкретної задачі та природи даних, які вивчаються. Важливо враховувати, що вибір правильного методу моделювання може суттєво вплинути на точність та надійність результатів.

Один із ключових аспектів моделювання – аналіз та перевірка моделей. Цей процес необхідний для забезпечення ефективності та адекватності моделі у вирішенні конкретних завдань. Важливо регулярно переглядати та оптимізувати моделі, особливо якщо у них виникають зміни в даних чи у вихідних умовах задачі.

Здійснимо підготовку даних для моделювання часового ряду, використовуючи розділення даних на навчальну та тестову вибірки. Важливо розділити дані на ці дві частини для того, щоб навчити модель на певному періоді (навчальна вибірка) та перевірити її ефективність на іншому періоді (тестова вибірка) (рис. 3.25 – 3.29).

```

Train Test Split

data = data.sort_values(by = 'date')

train = data['1997-01-06': '2022-12-31'] # 7 Jan 1997 to 31 Dec 2022
test = data['2023-01-01': '2023-10-01'] # 1 Jan 2023 to 1 Oct 2023

print("Length of Train Data: ", len(train))
print("Length of Test Data: ", len(test))

Length of Train Data: 6623
Length of Test Data: 274

```

Рисунок 3.25 – Фрагмент коду та відображення розміру навчальної (Train Data) та тестової вибірки (Test Data)

```

train.tail(5)

Out[18]:

```

| date       | gas_price |
|------------|-----------|
| 2022-12-27 | 5.18      |
| 2022-12-28 | 4.78      |
| 2022-12-29 | 4.15      |
| 2022-12-30 | 4.08      |
| 2022-12-31 | 4.11      |

Рисунок 3.26 – Фрагмент коду та відображення 5 останніх значень навчальної вибірки (Train Data)

```
test.tail(5)
```

Out[20]:

|            | gas_price |
|------------|-----------|
| date       |           |
| 2023-09-27 | 2.89      |
| 2023-09-28 | 2.93      |
| 2023-09-29 | 2.93      |
| 2023-09-30 | 2.93      |
| 2023-10-01 | 2.84      |

Рисунок 3.27 – Фрагмент коду та відображення 5 останніх значень тестової вибірки (Test Data)

```
ax = train.plot(figsize=(20, 10), color='b')
test.plot(ax=ax, color='black')

# Отримуємо останній індекс з train для визначення межі між train та test
train_last_index = train.index[-1]

# Малюємо вертикальну лінію, розділяючи train від test
plt.axvline(x=train_last_index, color='r', linestyle='--', linewidth=2)

plt.legend(['train set', 'test set'])
plt.show()
```

Рисунок 3.28 – Фрагмент коду для відображення на графіку навчальної та тестової вибірки

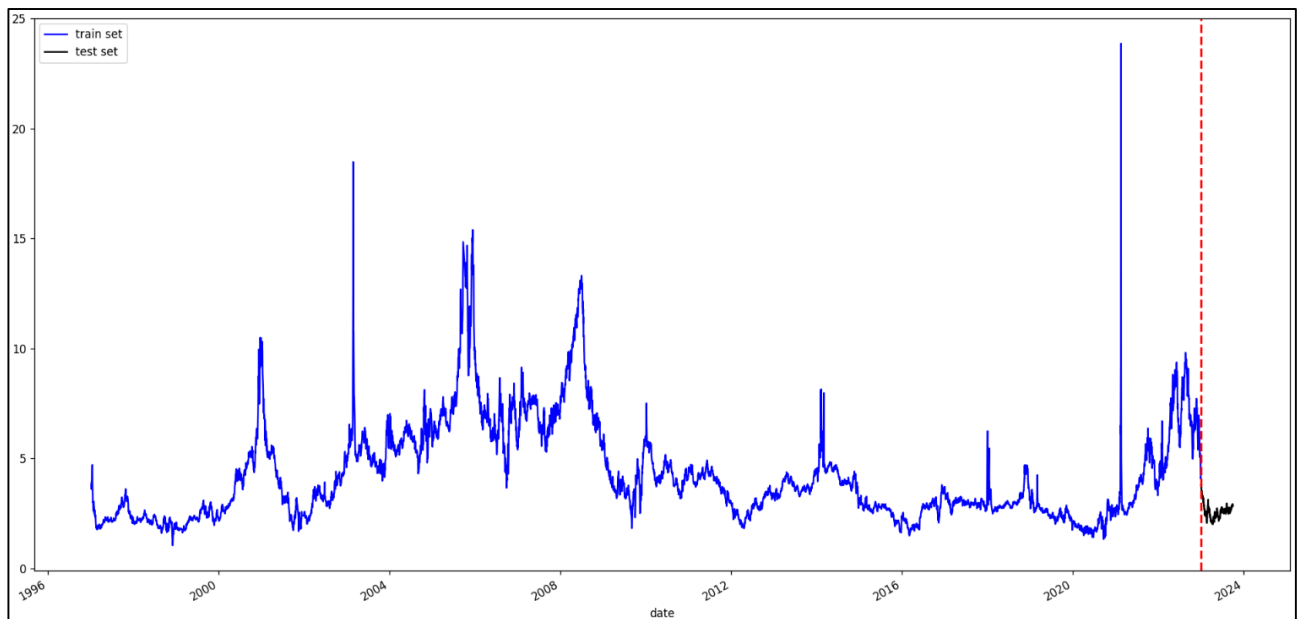


Рисунок 3.29 – Графік відображення навчальної та тестової вибірки

Наступним етапом роботи буде використання моделі ARIMA. Моделі ARIMA є потужними інструментами для прогнозування часових рядів і широко використовуються в різних галузях, включаючи фінанси, економіку та природничі науки, серед інших. Вони особливо корисні для моделювання та прогнозування часово-залежних даних з трендами та сезонністю.

Насправді це клас моделей, які «пояснюють» певний часовий ряд на основі його власних минулих значень, тобто його власних затримок і помилок прогнозу з відставанням, так що це рівняння можна використовувати для прогнозування майбутніх значень.

Необхідна умова: дані повинні бути нерухомими.

Модель ARIMA характеризується трьома членами:  $p$ : порядок члена AR,  $q$ : термін MA,  $d$  — кількість різниць, необхідних для того, щоб часовий ряд став стаціонарним [30].

Визначення порядку диференціювання, часто позначеного як " $d$ ," є важливим етапом аналізу та моделювання часових рядів. Диференціювання – це техніка, яка використовується для зроблення часового ряду стаціонарним, що є передумовою для моделювання за допомогою моделей, таких як ARIMA (авторегресійна інтегрована модель ковзного середнього) (рис. 3.30, 3.31).

```
Finding d: Number of differences required.

plt.rcParams.update({'figure.figsize':(9,7), 'figure.dpi':120})

# Original Series
fig, axes = plt.subplots(4, 2, sharex=True)
axes[0, 0].plot(data.gas_price); axes[0, 0].set_title('Original Series')
plot_acf(data.gas_price, ax=axes[0, 1])

# 1st Differencing
axes[1, 0].plot(data.gas_price.diff()); axes[1, 0].set_title('1st Order Differencing')
plot_acf(data.gas_price.diff().dropna(), ax=axes[1, 1])

# 2nd Differencing
axes[2, 0].plot(data.gas_price.diff().diff()); axes[2, 0].set_title('2nd Order Differencing')
plot_acf(data.gas_price.diff().diff().dropna(), ax=axes[2, 1])

# 3rd Differencing
axes[3, 0].plot(data.gas_price.diff().diff().diff()); axes[3, 0].set_title('3rd Order Differencing')
plot_acf(data.gas_price.diff().diff().diff().dropna(), ax=axes[3, 1])

plt.show()
```

Рисунок 3.30 – Фрагмент коду для визначення порядку диференціювання та представлення графіків

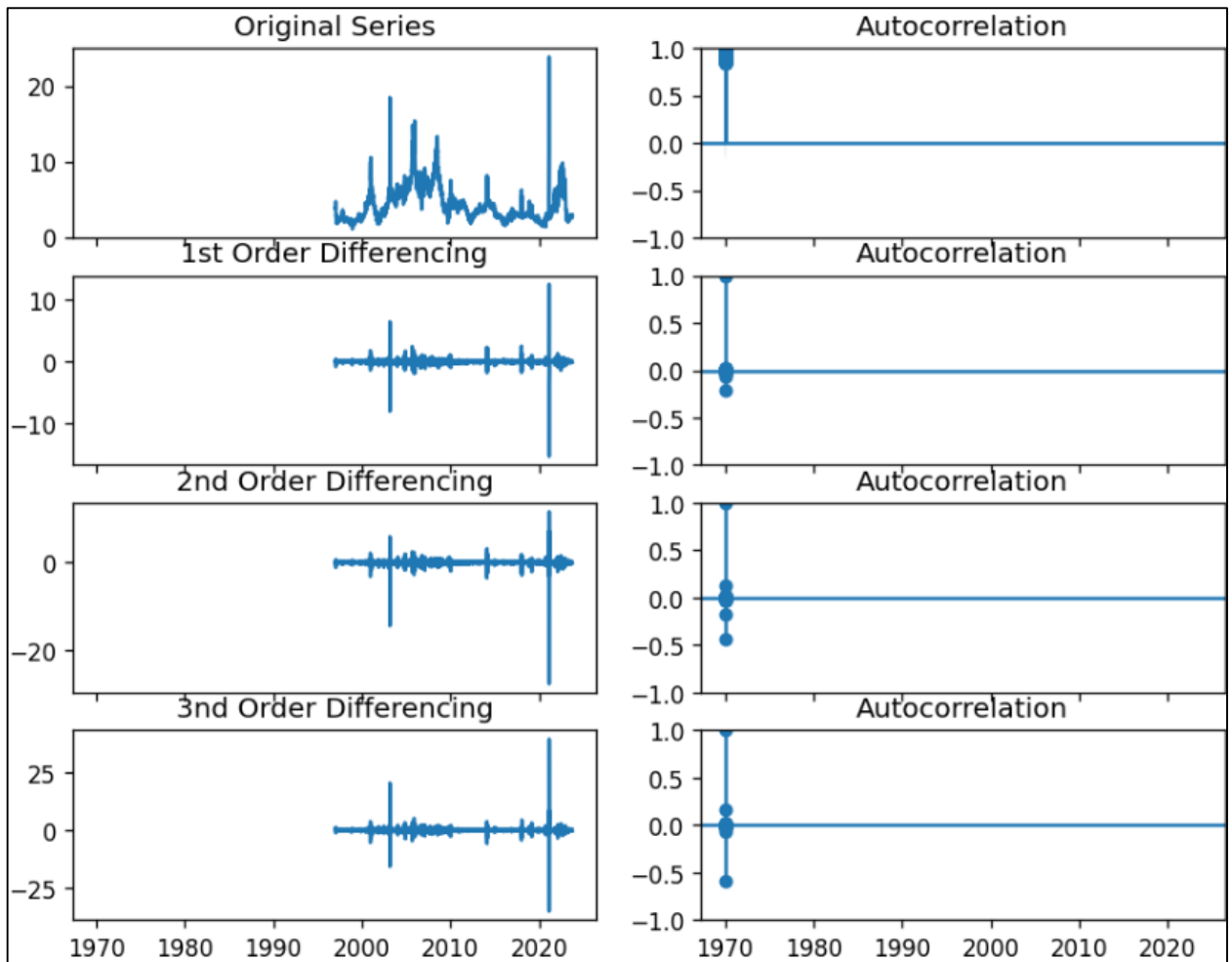


Рисунок 3.31 – Автокореляційні функції для ряду даних ціни на природній газ та його першої, другої та третьої різниць

За допомогою коду, який зображено на рисунку 3.32, отримуємо, що оптимальне  $d = 1$ .

```
# Функція для визначення порядку різницювання за допомогою ADF-тесту
def find_optimal_d(data):
    max_d = 5 # Максимальна кількість різницювань, яку ви розглядаєте
    for d in range(max_d + 1):
        result = adfuller(data.diff(d).dropna())
        p_value = result[1]
        if p_value < 0.05:
            return d # Повертаємо перший d, при якому p-value стає менше 0.05
    return max_d # Якщо жодне значення d не влаштовує, повертаємо максимальне значення

# Викликаємо функцію для знаходження оптимального d
optimal_d = find_optimal_d(data.gas_price)
print(f"Optimal order of differencing (d): {optimal_d}")
```

Optimal order of differencing (d): 1

Рисунок 3.32 – Фрагмент коду



Пошук порядку авторегресійного (AR) члена, позначеного як "p" в ARIMA (p, d, q), є важливим етапом в моделюванні часових рядів. AR-член визначає кількість затримок значень часового ряду, які повинні бути включені до моделі. Щоб визначити відповідний порядок AR-члена, ви можете використовувати наступні методи:

- Графік часткової автокореляційної функції (PACF): Графік часткової автокореляційної функції допомагає визначити порядок AR-члена. Значущі пікові значення при певних затримках вказують на кореляцію цих затримок із поточним значенням. Порядок AR-члена визначається кількістю значущих пікових значень, за якими PACF стає незначущим. Наприклад, якщо на графіку PACF спостерігаються значущі пікові значення на затримці 2 та затримці 3, і подальші піки не є значущими, це вказує на модель AR(2);

- Інформаційні критерії: Інформаційні критерії, такі як AIC (критерій Акаїке) і BIC (Байєсівський інформаційний критерій), можуть бути використані для порівняння різних моделей ARIMA з різними значеннями "p." Нижчі значення AIC або BIC вказують на кращу модель. Ви можете спробувати різні значення "p" і вибрати порядок, який мінімізує інформаційні критерії;

- Пошук за сіткою: Підхід з пошуку за сіткою передбачає встановлення моделей ARIMA з різними значеннями "p" та оцінювання їх продуктивності за допомогою методів, таких як перехресна валідація. Ви можете систематично тестувати різні значення "p" і вибирати той, який призводить до кращих результатів моделювання;

- Знання в галузі. У деяких випадках знання в галузі або експертиза можуть вказати на вибір порядку AR. Якщо у вас є глибоке розуміння процесів, які лежать в основі даних, ви можете зробити обґрунтований вибір порядку "p".

Варто зауважити, що вибір порядку AR не завжди є прямолінійним, і може знадобитися експериментувати та тестувати різні варіанти. Важливо знайти баланс між складністю моделі та точністю прогнозування. Перенавчання може виникнути, якщо ви виберете дуже велике значення "p," тому важливо перевірити вибрану

модель на вибіркових даних, щоб переконатися, що вона добре справляється з прогнозуванням майбутніх значень.

Далі дізнаємося необхідну кількість термінів AR, перевіряючи графік часткової автокореляції (PACF). Часткову автокореляцію можна уявити як кореляцію між рядом і його відставанням (рис. 3.33, 3.34).

```
plt.rcParams.update({'figure.figsize':(9,3), 'figure.dpi':120})

fig, axes = plt.subplots(1, 2, sharex=True)
axes[0].plot(data.gas_price.diff()); axes[0].set_title('1st Differencing')
axes[1].set(ylim=(0,5))
plot_pacf(data.gas_price.diff().dropna(), ax=axes[1])

plt.show()
```

Рисунок 3.33 – Фрагмент коду для пошуку порядку авторегресійного (AR) члена та представлення графіка

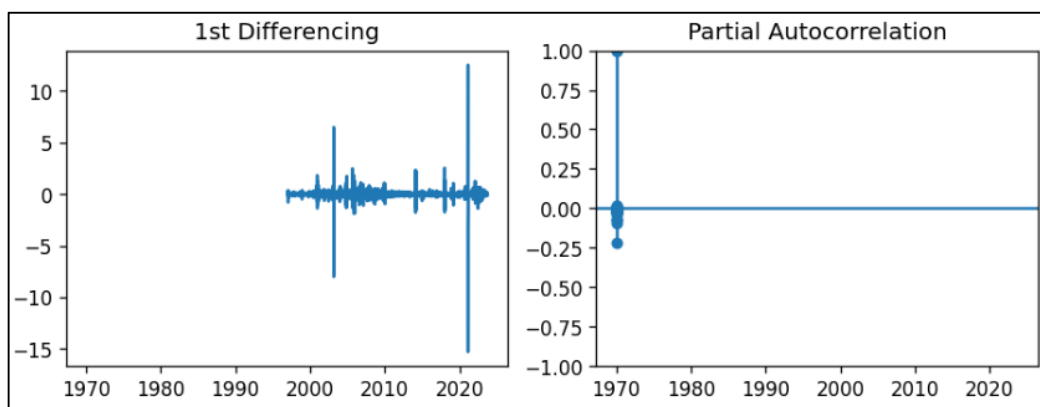


Рисунок 3.34 – Часткова автокореляційна функція першої різниці ціни на природній газ

Визначення порядку члену ковзного середнього (MA), позначеного як "q" в ARIMA (p, d, q), є ще одним важливим етапом у моделюванні часових рядів. Член MA представляє собою кількість запізнених прогнозних помилок (залишкових компонентів) з минулих прогнозів, які слід включити до моделі. Для визначення відповідного порядку члена MA можна використовувати наступні методи:

– Автокореляційна функція (ACF): Графік ACF може допомогти визначити порядок члена MA. На графіку ACF значущі піки при певних запізненнях вказують на кореляцію цих запізнень із поточним значенням. Порядок члена MA визначається кількістю значущих піків, за якими значення ACF стають несуттєвими. Наприклад, якщо на графіку ACF спостерігаються значущі піки на запізненні 2 та запізненні 3, і подальші піки не є значущими, це вказує на модель MA(2);

– Інформаційні критерії: Інформаційні критерії, такі як AIC (критерій Акаїке) і BIC (Байєсівський інформаційний критерій), можуть бути використані для порівняння різних моделей ARIMA з різними значеннями "q." Нижчі значення AIC або BIC вказують на кращу модель. Ви можете спробувати різні значення "q" і вибрати порядок, який мінімізує інформаційні критерії;

– Пошук за сіткою: Аналогічно до члена AR, ви можете виконувати підхід з пошуку за сіткою для підгонки моделей ARIMA з різними значеннями "q" і оцінювання їх продуктивності за допомогою методів, таких як перехресна валідація. Систематично тестуйте різні значення "q" і вибирайте той, який призводить до кращих результатів моделювання;

– Знання в галузі: Як і в разі члена AR, в деяких випадках знання в галузі або експертиза можуть вказати на вибір порядку члена MA. Якщо у вас є глибоке розуміння підлеглих процесів, ви можете прийняти обґрунтоване рішення щодо відповідного значення "q".

Вибір порядку члена MA також може вимагати експериментів і тестувань, і важливо знайти баланс між складністю моделі та точністю прогнозування. Перенавчання може виникнути, якщо ви виберете дуже велике значення "q," тому важливо перевірити вибрану модель на вибіркових даних, щоб переконатися, що вона добре справляється з прогнозуванням майбутніх значень. Технічно термін MA є помилкою відкладеного прогнозу (3.35 – 3.36).

```
plt.rcParams.update({'figure.figsize':(9,3), 'figure.dpi':120})

fig, axes = plt.subplots(1, 2, sharex=True)

axes[0].plot(data.gas_price.diff()); axes[0].set_title('1st Differencing')
axes[1].set(ylim=(0,1.2))
plot_acf(data.gas_price.diff().dropna(), ax=axes[1])

plt.show()
```

Рисунок 3.35 – Фрагмент коду для визначення порядку члену ковзного середнього (МА) та представлення графіка

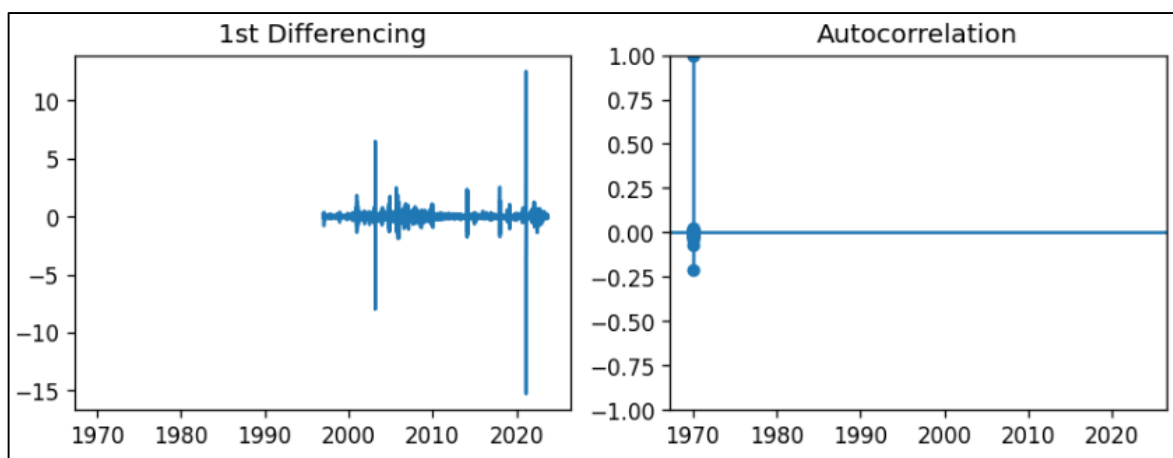


Рисунок 3.36 – Автокореляційна функція першої різниці ціни на природній газ

Після аналізу можна зробити висновок, що оптимальною моделлю є:

$$ARIMA(p, d, q) = ARIMA(1,1,1). \quad (3.1)$$

Здійснимо за допомогою коду, який представлений на рисунку 3.37, прогнозування ціни на природній газ за оптимальною моделлю ARIMA (3.1) та представимо його на рисунку 3.38.

```
model = sm.tsa.arima.ARIMA(train, order = (1, 1, 1))
arima_model = model.fit()
print(arima_model.summary())
```

Рисунок 3.37 – Фрагмент коду для представлення прогнозування ціни на природній газ за оптимальною моделлю ARIMA (3.1)

| SARIMAX Results   |                  |                   |                   |             |        |        |
|---|------------------|-------------------|-------------------|-------------|--------|--------|
| Dep. Variable:  | gas_price        | No. Observations: | 6623              |             |        |        |
| Model:  | ARIMA(1, 1, 1)   | Log Likelihood    | -2672.144         |             |        |        |
| Date:   | Sun, 10 Dec 2023 | AIC               | 5350.289          |             |        |        |
| Time:   | 23:21:18         | BIC               | 5370.683          |             |        |        |
| Sample:   | 0                | HQIC              | 5357.336          |             |        |        |
|   | - 6623           |                   |                   |             |        |        |
| Covariance Type:  | opg              |                   |                   |             |        |        |
|   | coef             | std err           | z                 | P> z        | [0.025 | 0.975] |
| ar.L1   | 0.6503           | 0.006             | 105.451           | 0.000       | 0.638  | 0.662  |
| ma.L1   | -0.8151          | 0.006             | -147.378          | 0.000       | -0.826 | -0.804 |
| sigma2  | 0.1312           | 0.000             | 943.247           | 0.000       | 0.131  | 0.131  |
| Ljung-Box (L1) (Q):   |                  | 16.25             | Jarque-Bera (JB): | 88970665.61 |        |        |
| Prob(Q):  |                  | 0.00              | Prob(JB):         | 0.00        |        |        |
| Heteroskedasticity (H):   |                  | 1.93              | Skew:             | 3.96        |        |        |
| Prob(H) (two-sided):  |                  | 0.00              | Kurtosis:         | 570.80      |        |        |
| Warnings:   |                  |                   |                   |             |        |        |
| [1] Covariance matrix calculated using the outer product of gradients (complex-step). |                  |                   |                   |             |        |        |

Рисунок 3.38 – Результат прогнозування ціни на природній газ за оптимальною моделлю ARIMA (3.1)

За допомогою коду, який зображеного на рисунку 3.39, використовується модель ARIMA для проведення прогнозу на навчальних і тестових даних, а також для обчислення середньоквадратичної помилки (MSE) з метою оцінки точності прогнозу.

MSE використовується для оцінки точності прогнозів, де низьке значення вказує на високу точність. Отримані результати зображено на рисунку 3.40, демонструючи значення MSE для навчальних та тестових даних.

```
yp_train = arima_model.predict(start=0, end=(len(train)-1))
yp_test = arima_model.predict(start=0, end=(len(test)-1))

print("Train Data:\nMean Square Error: {}".format(mean_squared_error(train, yp_train)))
print("\nTest Data:\nMean Square Error: {}".format(mean_squared_error(test, yp_test)))
```

Рисунок 3.39 – Фрагмент коду для оцінки точності прогнозу

```

Train Data:
Mean Square Error: 0.13340883188569427

Test Data:
Mean Square Error: 0.250044865036274

```

Рисунок 3.40 – Оцінка точності прогнозів

Відобразимо на рисунках 3.41, 3.42 наглядний приклад порівняння прогнозованих тестових даних з оригінальними тестовими даними за моделлю ARIMA (3.1)

```

print(yp_test)

```

| date       |          |
|------------|----------|
| 1997-01-07 | 0.000000 |
| 1997-01-08 | 3.820000 |
| 1997-01-09 | 3.802563 |
| 1997-01-10 | 3.638849 |
| 1997-01-13 | 3.896761 |
|            | ...      |
| 1998-02-02 | 2.090079 |
| 1998-02-03 | 2.206993 |
| 1998-02-04 | 2.244655 |
| 1998-02-05 | 2.215932 |
| 1998-02-06 | 2.285349 |

Рисунок 3.41 – Прогнозовані значення на тестових даних за моделлю ARIMA (3.1)

```

print(train)

```

| date       | gas_price |
|------------|-----------|
| 1997-01-07 | 3.82      |
| 1997-01-08 | 3.80      |
| 1997-01-09 | 3.61      |
| 1997-01-10 | 3.92      |
| 1997-01-13 | 4.00      |
|            | ...       |
| 2022-12-27 | 5.18      |
| 2022-12-28 | 4.78      |
| 2022-12-29 | 4.15      |
| 2022-12-30 | 4.08      |
| 2022-12-31 | 4.11      |

Рисунок 3.42 – Оригінальні значення тестових даних

Наступним етапом роботи буде використання моделі довготривалої короткочасної пам'яті (LSTM) – це вид рекурентних нейронних мереж, який спроектований для обробки послідовностей даних та моделювання довготривалих залежностей в таких даних. Вона є розширенням звичайних рекурентних нейронних мереж (RNN) і вирішує проблему втрати короткотермінової пам'яті у звичайних RNN.

Основні особливості LSTM:

- Довготривала пам'ять: LSTM має можливість зберігати і використовувати довготривалу інформацію. Це допомагає вирішити проблему зникнення та вибування градієнта, яка властива звичайним RNN;
- Ворота: LSTM включає в себе спеціальні ворота, такі як ворота забування, ворота входу та вихідне ворота, які контролюють, що додається до довготривалої пам'яті та що передається на вихід. Це дозволяє LSTM визначати, коли і яка інформація повинна зберігатися або викидатися;
- Модуль контролю згубності: Модель LSTM також включає модуль контролю згубності, який дозволяє вирішити проблему згубності градієнта, що виникає в звичайних RNN;
- Додатковий стан школи входу і виходу: LSTM може зберігати стан школи для кожного входу та виходу. Це дає йому можливість визначати, яку частину інформації тримати на поточному кроці часу та яку виводити.

LSTM моделі використовуються для різних завдань, таких як машинний переклад, розпізнавання мови, прогнозування часових рядів, обробка природних мов, зображень і багато інших. Вони особливо корисні в задачах, де важливо враховувати довготривалі залежності в даних [31].

Використаємо код, який має на меті побудову і компіляцію моделі глибокого навчання для роботи з часовими рядами або послідовностями даних. Основним елементом моделі є шари LSTM (Long Short-Term Memory), які призначені для роботи з послідовними даними і здатні враховувати довгострокові залежності (рис. 3.43).

```

# Функція для побудови моделі, яку будемо оптимізувати
def build_lstm_model(units, layers, dropout, learning_rate):
    model = Sequential()
    model.add(LSTM(units=units, input_shape=(slot, 1), return_sequences=True, activation='relu'))
    for _ in range(layers - 1):
        model.add(LSTM(units=units, return_sequences=True, activation='relu'))
    model.add(LSTM(units=units, return_sequences=False, activation='relu'))
    model.add(Dense(units=1))
    model.compile(loss='mean_squared_error', optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rate))
    return model

```

Рисунок 3.43 – Фрагмент коду для побудови моделі, яку будемо оптимізувати

Здійснимо підготовку даних для навчання моделі LSTM та відобразимо результат на рисунку 3.44.

```

slot = 10
x_train = []
y_train = []

for i in range(slot, len(train)):
    x_train.append(train.iloc[i-slot:i, 0])
    y_train.append(train.iloc[i, 0])

x_train, y_train = np.array(x_train), np.array(y_train)

x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
print(x_train.shape, y_train.shape)

```

Рисунок 3.44 – Фрагмент коду для підготовки даних для навчання моделі LSTM

За допомогою коду, зображеного на рисунку 3.45, створимо модель для оптимізації за допомогою бібліотеки Keras. Ця модель буде використовуватися в процесі оптимізації.

```

# Створення моделі для оптимізації
model = KerasRegressor(build_fn=build_lstm_model, verbose=0)

```

Рисунок 3.45 – Фрагмент коду створення моделі для оптимізації

Використаємо код, який визначає гіперпараметри, які будуть оптимізовані під час тренування моделі. Для кожного гіперпараметра вказані деякі значення або



діапазони, які будуть розглядатися в процесі оптимізації для вибору оптимальних значень цих параметрів (рис. 3.46).

```
# Визначення гіперпараметрів, які ви хочете оптимізувати та їх діапазонів
param_dist = {
    'units': [25, 50, 75],
    'layers': [1, 2, 3],
    'dropout': [0.1, 0.2, 0.3],
    'learning_rate': [0.001, 0.01, 0.1]
}
```

Рисунок 3.46 – Фрагмент коду для визначення гіперпараметрів

За допомогою методу `RandomizedSearchCV`, виконаємо процедуру оптимізації гіперпараметрів. Використовується модель `model`, визначена раніше, та гіперпараметри зазначені у `param_dist`. Алгоритм випадкового пошуку (`RandomizedSearchCV`) шукає найкращі комбінації гіперпараметрів, оцінюючи їх вплив на модель за метрикою негативної середньої квадратичної помилки (`neg_mean_squared_error`). Така оптимізація відбувається в рамках крос-валідації з 3 фолдами (рис. 3.47).

```
# Виконання RandomizedSearchCV для пошуку найкращих параметрів
random_search = RandomizedSearchCV(model, param_distributions=param_dist, n_iter=10, scoring='neg_mean_squared_error', cv=3)
random_search.fit(x_train, y_train)
```

Рисунок 3.47 – Фрагмент коду для виконання процедури оптимізації гіперпараметрів

Використовуючи код, зображений на рисунку 3.48, отримаєм вивід на екран найкращих параметрів та найкращий результат, які були отримані під час оптимізації гіперпараметрів за допомогою методу `RandomizedSearchCV` (рис. 3.49). Найкращі параметри знаходяться в змінній `best_params`, а найкращий результат (негативна середньо-квадратична помилка) – в змінній `best_score`.

```
# Отримання найкращих параметрів і результатів
best_params = random_search.best_params_
best_score = -random_search.best_score_

print("Найкращі параметри:", best_params)
print("Найкращий результат (MSE):", best_score)
```

Рисунок 3.48 – Фрагмент коду для отримання та виводу найкращих показників

```
(6613, 10, 1) (6613,)
Найкращі параметри: {'units': 25, 'learning_rate': 0.01, 'layers': 1, 'dropout': 0.1}
Найкращий результат (MSE): 0.264644388181367
```

Рисунок 3.49 – Вивід найкращих параметрів та результату

За допомогою даного коду, зображеного на рисунку 3.50, визначаємо та компілюємо модель LSTM для обробки часових рядів з 6613 прикладами, кожен із 10 відмірюваннями. Модель використовує 25 нейронів у шарі LSTM, функцію активації ReLU, а також оптимізатор Adam та функцію втрат mean squared error для навчання. Розмірність вхідних даних задається як (10, 1). Завершальний вивід `summary()` дозволяє переглянути параметри та структуру побудованої моделі (рис. 3.51).

```
# Зазначте розмір вхідних даних. У вас є 6613 прикладів з 10 відмірюваннями в кожному прикладі.
input_shape = (10, 1)

# Створіть модель LSTM з найкращими параметрами
lstm_model = tf.keras.Sequential()
lstm_model.add(tf.keras.layers.LSTM(units=25, input_shape=input_shape, return_sequences=False, activation='relu'))
lstm_model.add(tf.keras.layers.Dense(units=1))

# Скомпілюйте модель з функцією втрати MSE і оптимізатором Adam
lstm_model.compile(loss='mean_squared_error', optimizer='adam')

# Виведіть інформацію про модель
lstm_model.summary()
```

Рисунок 3.50 – Фрагмент коду для створення моделі LSTM з найкращими параметрами

| Layer (type)            | Output Shape | Param # |
|-------------------------|--------------|---------|
| lstm_89 (LSTM)          | (None, 25)   | 2700    |
| dense_31 (Dense)        | (None, 1)    | 26      |
| Total params: 2,726     |              |         |
| Trainable params: 2,726 |              |         |
| Non-trainable params: 0 |              |         |

Рисунок 3.51 – Розгорнута інформація про модель, включаючи конфігурацію шару і кількість параметрів моделі.

За допомогою коду, зображеного на рисунку 3.52, налаштуємо і навчимо побудовану модель LSTM на вхідних даних `x_train` та `y_train`. Використовуючи реалізований механізм `EarlyStopping`, навчання автоматично припиняється, якщо значення функції втрати не покращується протягом 10 епох. Тривалість навчання обмежена 100 епохами, а розмір партії встановлено на 64. Вивід навчання виводиться для кожної епохи, а дані не перемішуються перед кожною епохою для забезпечення послідовності введення (рис. 3.53).

```
early_stopping = tf.keras.callbacks.EarlyStopping(monitor = 'loss', patience = 10)

history = lstm_model.fit(x_train, y_train, epochs = 100,
                        batch_size = 64,
                        verbose = 1, shuffle = False,
                        callbacks = [early_stopping])
```

Рисунок 3.52 – Фрагмент коду для налаштування та навчання моделі LSTM

```

Epoch 1/100
104/104 [=====] - 3s 16ms/step - loss: 3.8741
Epoch 2/100
104/104 [=====] - 2s 16ms/step - loss: 1.7401
Epoch 3/100
104/104 [=====] - 2s 15ms/step - loss: 0.4505
Epoch 4/100
104/104 [=====] - 2s 16ms/step - loss: 0.3029
Epoch 5/100
104/104 [=====] - 2s 16ms/step - loss: 0.2426
Epoch 6/100
104/104 [=====] - 2s 16ms/step - loss: 0.2339
Epoch 7/100
104/104 [=====] - 2s 16ms/step - loss: 0.2211
Epoch 8/100
104/104 [=====] - 2s 16ms/step - loss: 0.2275
Epoch 9/100
104/104 [=====] - 2s 16ms/step - loss: 0.2187
Epoch 10/100
104/104 [=====] - 2s 16ms/step - loss: 0.2161
Epoch 11/100
104/104 [=====] - 2s 16ms/step - loss: 0.1994
Epoch 12/100
104/104 [=====] - 2s 16ms/step - loss: 0.2046
Epoch 13/100
104/104 [=====] - 2s 18ms/step - loss: 0.1949
Epoch 14/100
104/104 [=====] - 2s 16ms/step - loss: 0.1832
Epoch 15/100
104/104 [=====] - 2s 16ms/step - loss: 0.1812

```

Рисунок 3.53 – Навчання моделі

### 3.5 Оцінка моделі

За допомогою коду, зображеного на рисунку 3.54, отримуємо прогнозовані значення за допомогою навченої моделі LSTM на тренувальних даних “x\_train”. Отримані прогнози зберігаються в DataFrame, а також перейменовані в стовпець 'gr\_pred'. Індеси цього DataFrame встановлюються таким чином, щоб відповідати індексам вихідних тренувальних даних. Остаточний результат, “train\_compare”, є об'єднанням вихідних тренувальних даних та прогнозованих значень для подальшого порівняння.

```

yp_train = lstm_model.predict(x_train)
a = pd.DataFrame(yp_train)
a.rename(columns = {0: 'gp_pred'}, inplace = True);
a.index = train.iloc[slot:].index
train_compare = pd.concat([train.iloc[slot:], a], 1)

```

Рисунок 3.54 – Фрагмент коду

За допомогою коду, зображеного на рисунку 3.55, виводимо перші п'ять рядків DataFrame “train\_compare”, що містить порівняння між вихідними тренувальними даними та прогнозованими значеннями, отриманими за допомогою навченої моделі LSTM.

| train_compare.head(5) |           |          |
|-----------------------|-----------|----------|
| date                  | gas_price | gp_pred  |
| 1997-01-21            | 2.99      | 3.247921 |
| 1997-01-22            | 3.05      | 2.935241 |
| 1997-01-23            | 2.96      | 3.030873 |
| 1997-01-24            | 2.62      | 2.918512 |
| 1997-01-27            | 2.98      | 2.574125 |

Рисунок 3.55 – Фрагмент коду з виводом 5 перших значень

Аналогічним чином, за допомогою коду, зображеного на рисунку 3.56, виводимо останні п'ять рядків DataFrame “train\_compare”, який містить порівняння між вихідними тренувальними даними та прогнозованими значеннями, отриманими за допомогою навченої моделі LSTM.

```
train_compare.tail(5)
```

|            | gas_price | gp_pred  |
|------------|-----------|----------|
| date       |           |          |
| 2022-12-27 | 5.18      | 5.158491 |
| 2022-12-28 | 4.78      | 5.113295 |
| 2022-12-29 | 4.15      | 4.636959 |
| 2022-12-30 | 4.08      | 4.026400 |
| 2022-12-31 | 4.11      | 4.048673 |

Рисунок 3.56 – Фрагмент коду з виводом 5 останніх значень

Використовуючи код, зображений на рисунку 3.57, побудуємо графік порівнянь прогнозованих та фактичних значень цін на природній газ на тренувальних даних. Червона лінія представляє фактичні ціни на природній газ, тоді як синя лінія відображає прогнозовані ціни, отримані за допомогою навченої моделі LSTM. Графік дозволяє візуально порівняти прогнози з реальними даними та оцінити ефективність моделі (рис. 3.58).

```
plt.figure(figsize = (15, 5))
plt.plot(train_compare['gas_price'], color = 'red', label = "Actual Natural Gas Price")
plt.plot(train_compare['gp_pred'], color = 'blue', label = 'Predicted Price')
plt.title("Natural Gas Price Prediction on Train Data")
plt.xlabel('Time')
plt.ylabel('Natural gas price')
plt.legend(loc = 'best')
plt.show()
```

Рисунок 3.57 – Фрагмент коду

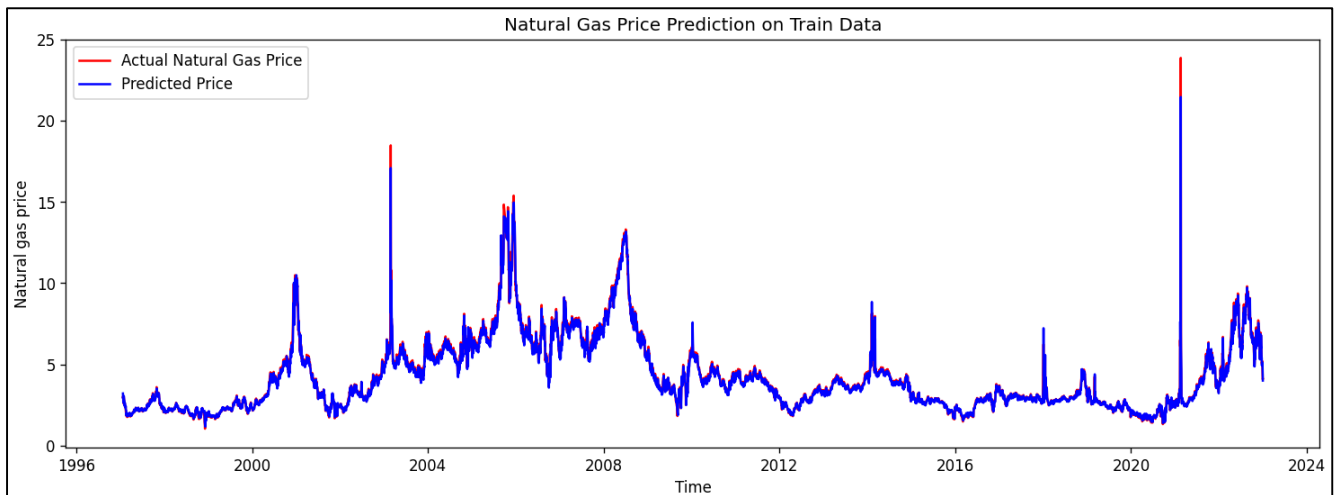


Рисунок 3.58 – Графік порівнянь

Здійснимо підготовчий етап для прогнозування цін на природній газ на тестовому періоді за допомогою попередньо навченої моделі LSTM (рис. 3.59). Відобразимо за допомогою коду зображеного на рисунку 3.60, порівняння фактичних цін на природній газ з прогнозованими цінами на тестовому періоді.

```
dataset_total = pd.concat((train, test), axis = 0)
inputs = dataset_total[len(dataset_total) - len(test) - slot:].values
inputs = inputs.reshape(-1, 1)

x_test = []
y_test = []
for i in range(slot, len(test)+slot): #Test+15
    x_test.append(inputs[i-slot:i, 0])
    y_test.append(train.iloc[i, 0])

x_test, y_test = np.array(x_test), np.array(y_test)

x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
pred_price = lstm_model.predict(x_test)
```

Рисунок 3.59 – Фрагмент коду

```

b = pd.DataFrame(pred_price)
b.rename(columns = {0: 'gp_pred'}, inplace = True)
b.index = test.index
test_compare = pd.concat([test, b], 1)

```

Рисунок 3.60 – Фрагмент коду

Використовуючи код, зображений на рисунку 3.61, відобразимо порівняння 5 перших фактичних цін на природній газ з прогнозованими цінами на тестовому періоді.

| test_compare.head(5) |           |          |
|----------------------|-----------|----------|
|                      | gas_price | gp_pred  |
| date                 |           |          |
| 2023-01-01           | 3.79      | 4.062461 |
| 2023-01-02           | 3.79      | 3.687189 |
| 2023-01-03           | 3.67      | 3.766954 |
| 2023-01-04           | 3.77      | 3.600213 |
| 2023-01-05           | 3.45      | 3.763173 |

Рисунок 3.61 – Фрагмент коду з виводом 5 перших значень

Аналогічним чином, за допомогою коду, зображеного на рисунку 3.62, відобразимо порівняння 5 останніх фактичних цін на природній газ з прогнозованими цінами на тестовому періоді.



```
test_compare.tail(5)
```

|            | gas_price | gp_pred  |
|------------|-----------|----------|
| date       |           |          |
| 2023-09-27 | 2.89      | 2.818312 |
| 2023-09-28 | 2.93      | 2.868621 |
| 2023-09-29 | 2.93      | 2.899367 |
| 2023-09-30 | 2.93      | 2.897109 |
| 2023-10-01 | 2.84      | 2.900004 |

Рисунок 3.62 – Фрагмент коду з виводом 5 останніх значень

Використовуючи код, зображений на рисунку 3.63, побудуємо графік порівнянь прогнозованих та фактичних значень цін на природній газ на тестових даних. Червона лінія представляє фактичні ціни на природній газ, тоді як синя лінія відображає прогнозовані ціни, отримані за допомогою навченої моделі LSTM. Графік дозволяє візуально порівняти прогнози з реальними даними та оцінити ефективність моделі (рис. 3.64).

```
plt.figure(figsize = (15,5))
plt.plot(test_compare['gas_price'], color = 'red', label = "Actual Natural Gas Price")
plt.plot(test_compare['gp_pred'], color = 'blue', label = 'Predicted Price')
plt.title("Natural Gas Price Prediction On Test Data")
plt.xlabel('Time')
plt.ylabel('Natural gas price')
plt.legend(loc = 'best')
plt.show()
```

Рисунок 3.63 – Фрагмент коду

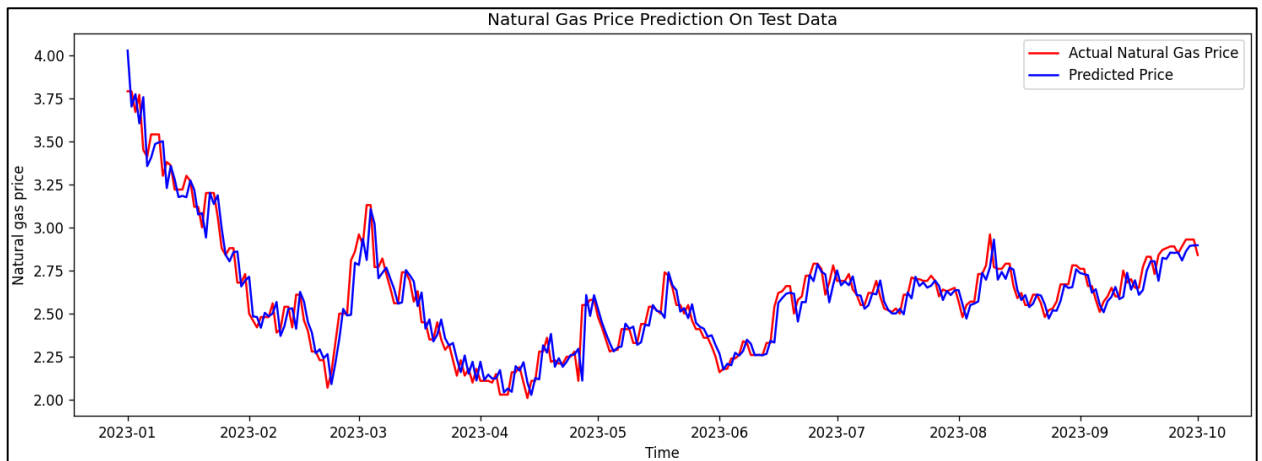


Рисунок 3.64 – Графік порівнянь

За допомогою коду, зображеного на рисунку 3.65, який обчислює значення середньої квадратичної помилки (MSE) та коефіцієнта детермінації ( $R^2$ ) для тренувальних та тестових даних, порівнюючи фактичні ціни на природний газ з прогнозованими значеннями, отриманими за допомогою моделі LSTM. Здійснимо оцінку точності моделі прогнозування ціни на природний газ. Результати для тренувального та тестового наборів виводяться на екран у вигляді середньої квадратичної помилки (MSE) та коефіцієнта детермінації ( $R^2$ ) (рис. 3.66).

```
mse_train = mean_squared_error(train_compare['gas_price'], train_compare['gp_pred'])
mse_test = mean_squared_error(test_compare['gas_price'], test_compare['gp_pred'])

r2_train = r2_score(train_compare['gas_price'], train_compare['gp_pred'])
r2_test = r2_score(test_compare['gas_price'], test_compare['gp_pred'])

print("Train Data:\nMSE: {}\nR Square: {}".format(mse_train, r2_train))
print("\nTest Data:\nMSE: {}\nR Square: {}".format(mse_test, r2_test))
```

Рисунок 3.65 – Фрагмент коду

```
Train Data:
MSE: 0.0884121685052636
R Square: 0.9819837816986361

Test Data:
MSE: 0.008638569695534469
R Square: 0.9203256292415846
```

Рисунок 3.66 – Показники оцінки точності

Використовуючи код, зображений на рисунку 3.67, будемо графік, на якому відображена динаміка зміни значень функції втрат під час тренування моделі LSTM протягом різних епох (рис. 3.68). Цей графік служить зручним інструментом для візуалізації того, як втрати моделі змінюються під час процесу навчання. Поступове зменшення втрат вказує на ефективне навчання моделі. З іншого боку, великі втрати можуть свідчити про перенавчання, що означає високу придатність моделі до тренувальних даних, але менш ефективну генералізацію до нових даних.

```
plt.plot(history.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train'], loc='upper right')
plt.show()
```

Рисунок 3.67 – Фрагмент коду

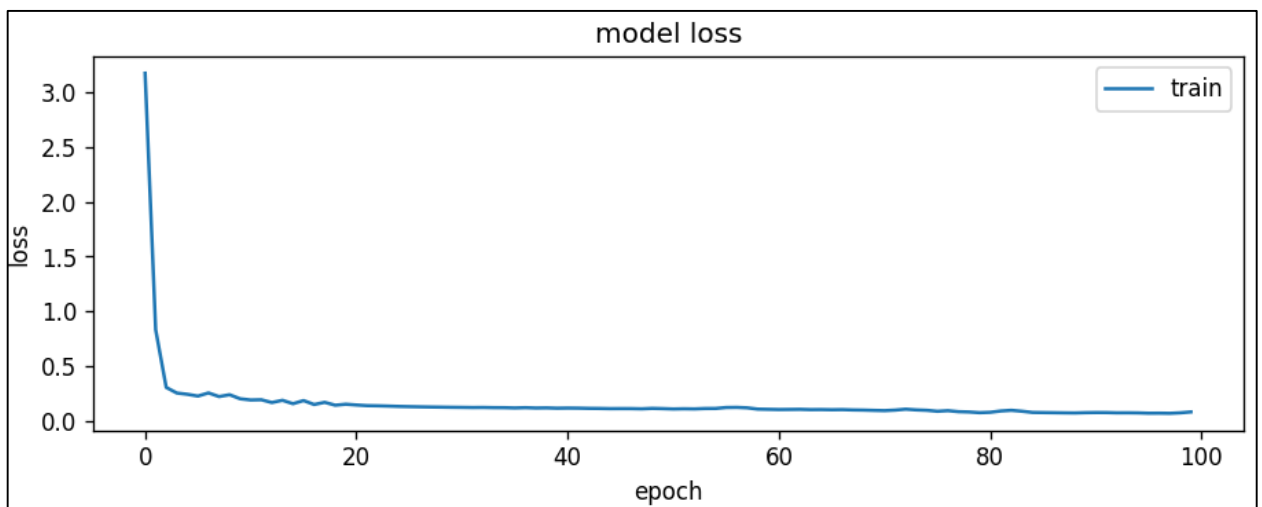


Рисунок 3.68 – Графік функції втрат протягом різних епох

Після виконаних дій було прийнято рішення сформувати таблицю 3.1 під назвою "Оцінка точності моделей", де детально представлені показники ефективності кожної обраної моделі.

Таблиця 3.1 – Оцінка точності моделей

| Назва моделі | Точність прогнозу на навчальній вибірці | Точність прогнозу на тестовій вибірці |
|--------------|---|---------------------------------------|
| ARIMA        | MSE = 0.133                             | MSE = 0.25                            |
| LSTM         | MSE = 0.088<br>R2_score = 0.98          | MSE = 0.008<br>R2_score = 0.92        |

Після аналізу даної таблиці було вирішено віддати перевагу використанню моделі LSTM для подальших прогнозів, враховуючи її високу точність та надійність у вирішенні поставленої задачі.

### 3.6 Прогнозування

За допомогою коду, наведеного на рисунку 3.69, формуємо прогноз цін на природній газ для періоду з 2 жовтня 2023 року по 12 жовтня 2023 року. Спочатку створюємо DataFrame “forecast”, що містить дати цього періоду. Далі визначаємо вхідні дані inputs на основі останніх значень тестового набору. У циклі, починаючи з розміру вікна (slot), отримуємо прогнозні ціни на природній газ за допомогою раніше навченої моделі LSTM. Прогнозні значення додаються до вхідних даних для подальшого використання в наступних ітераціях циклу. Завершальний DataFrame “forecast” містить прогнозовані ціни на природній газ з відповідними датами як індексами.

```
forecast = pd.DataFrame({'date': pd.date_range(start = '10/2/2023', end = '10/12/2023')}) # 2 October to 10 October
inputs = test[len(test) - slot: ].values

for i in range(slot, len(forecast)):
    inputs = inputs.T
    inputs = np.reshape(inputs, (inputs.shape[0], inputs.shape[1], 1))
    pred_price = lstm_model.predict(inputs[:,i-slot:i])
    inputs = np.append(inputs, pred_price)
    inputs = np.reshape(inputs, (inputs.shape[0], 1))

forecast['gp_pred'] = inputs
forecast = forecast.set_index('date')
```

Рисунок 3.69 – Фрагмент коду

Використовуючи код, зображений на рисунку 3.70, знову встановлюємо індекс у DataFrame “forecast”, що містить прогноз цін на природній газ на період з 2 жовтня 2023 року по 12 жовтня 2023 року. Після цього створюємо лінійний графік за допомогою бібліотеки Plotly Express (px.line) (рис. 3.71).

```
forecast.reset_index(inplace=True)

fig = px.line(forecast, x="date", y="gp_pred", title='Natural Gas Price Forecasting', template='plotly_dark',
             text=[f": {price:.4f}" for price in forecast['gp_pred']])

fig.update_layout(title='Natural Gas Price Forecasting',
                  xaxis_title='Date',
                  yaxis_title='Price',
                  template='plotly_dark',
                  height=600,
                  width=1000,
                  xaxis=dict(tickmode='linear'))

fig.show()
```

Рисунок 3.70 – Фрагмент коду

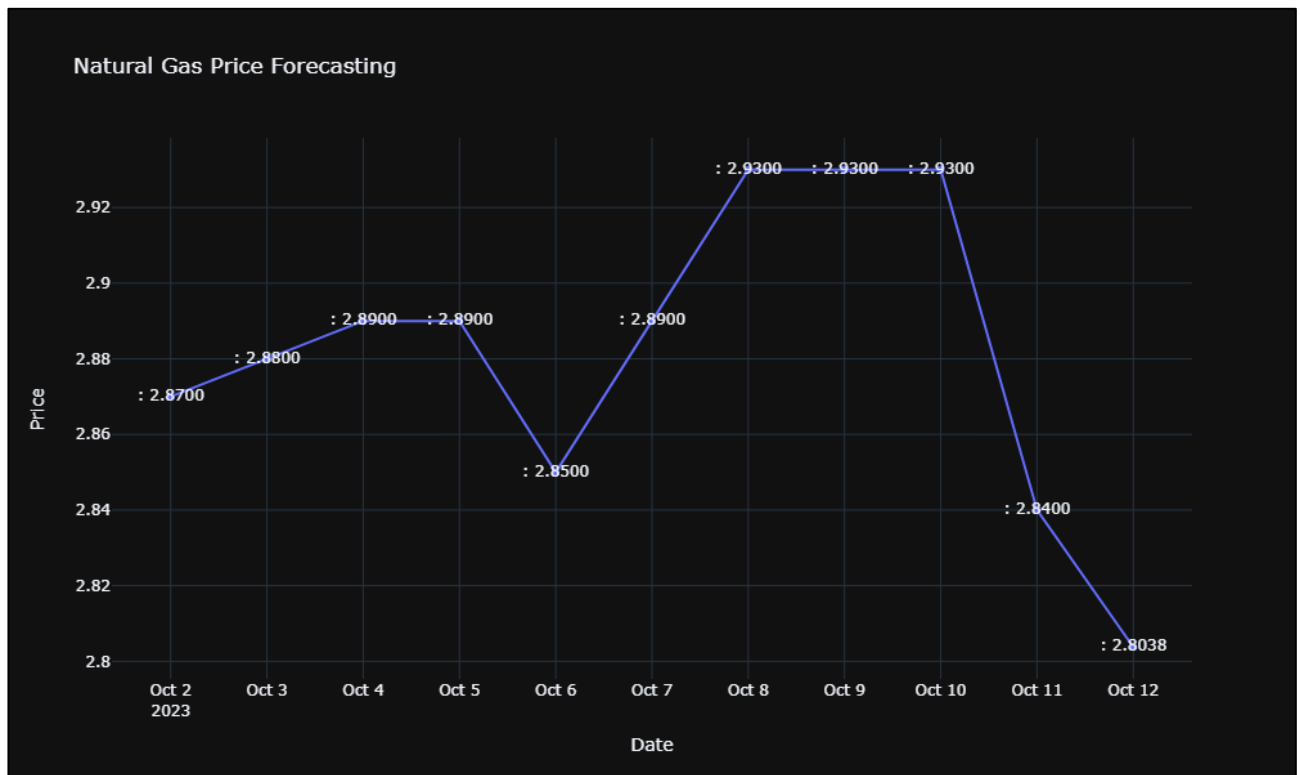


Рисунок 3.71 – Графік прогнозованих цін на природній газ на 10 днів

Графік відображає прогнозовані ціни на природній газ на зазначений період і має темний дизайн. Використання інтерактивного графічного візуалізатора дозволяє зручно оглядати та аналізувати прогнозовані значення цін.

Використовуючи код, зображений на рисунку 3.72, створюється DataFrame “real\_data”, який містить реальні ціни на природній газ з 2 по 12 жовтня 2023 року.

```
# Реальні дані
real_price_data = [2.845, 2.954, 2.974, 3.059, 3.045, 3.105, 3.157, 3.187, 3.190, 3.084, 2.909]

# Створення DataFrame для реальних даних
real_data = pd.DataFrame({'date': pd.date_range(start='10/2/2023', end='10/12/2023'),
                          'real_price': real_price_data})
```

Рисунок 3.72 – Фрагмент коду для створення DataFrame “real\_data”

Далі використовується існуючий код для прогнозу цін на природній газ для кожної дати з 2 по 12 жовтня 2023 року (рис. 3.73).

```
# Існуючий код для прогнозу
forecast = pd.DataFrame({'date': pd.date_range(start='10/2/2023', end='10/12/2023')}) # 2 October to 12 October

inputs = test[len(test) - slot:].values

for i in range(slot, len(forecast)):
    inputs = inputs.T
    inputs = np.reshape(inputs, (inputs.shape[0], inputs.shape[1], 1))
    pred_price = lstm_model.predict(inputs[:, i - slot:i])
    inputs = np.append(inputs, pred_price)
    inputs = np.reshape(inputs, (inputs.shape[0], 1))

forecast['gp_pred'] = inputs
forecast = forecast.set_index('date')
```

Рисунок 3.73 – Фрагмент коду

За допомогою коду, зображеного на рисунку, 3.74, здійснюється об’єднання прогнозованих та реальних даних за допомогою дати.

```
# Об’єднання прогнозованих та реальних даних
merged_data = pd.merge(forecast, real_data, on='date', how='left')

# Скидання індексу для побудови графіку
merged_data.reset_index(inplace=True)
```

Рисунок 3.74 – Фрагмент коду

Далі здійснюється побудова графіка з реальними та прогнозованими даними, за допомогою коду, зображеного на рисунку 3.75 (рис. 3.76).

```
# Побудова графіку з реальними та прогнозованими даними
fig = go.Figure()

# Додаємо прогнозовані дані
fig.add_trace(go.Scatter(x=merged_data['date'], y=merged_data['gp_pred'],
                        mode='lines+text', # змінено 'markers' на 'text'
                        name='Predicted Price',
                        text=[f"{price:.4f}" for price in merged_data['gp_pred']],
                        textposition="top center", # розташування тексту над точкою
                        hoverinfo='skip'))

# Додаємо реальні дані
fig.add_trace(go.Scatter(x=merged_data['date'], y=merged_data['real_price'],
                        mode='lines+text', # змінено 'markers' на 'text'
                        name='Real Price',
                        text=[f"{price:.4f}" for price in merged_data['real_price']],
                        textposition="top center", # розташування тексту над точкою
                        hoverinfo='skip'))

# Налаштування графіку
fig.update_layout(title='Natural Gas Price Forecasting',
                  xaxis_title='Date',
                  yaxis_title='Price',
                  template='plotly_dark',
                  height=600,
                  width=1000,
                  xaxis=dict(tickmode='linear')) # Змінено tickmode на 'linear'

# Показ графіку
fig.show()
```

Рисунок 3.75 – Фрагмент коду

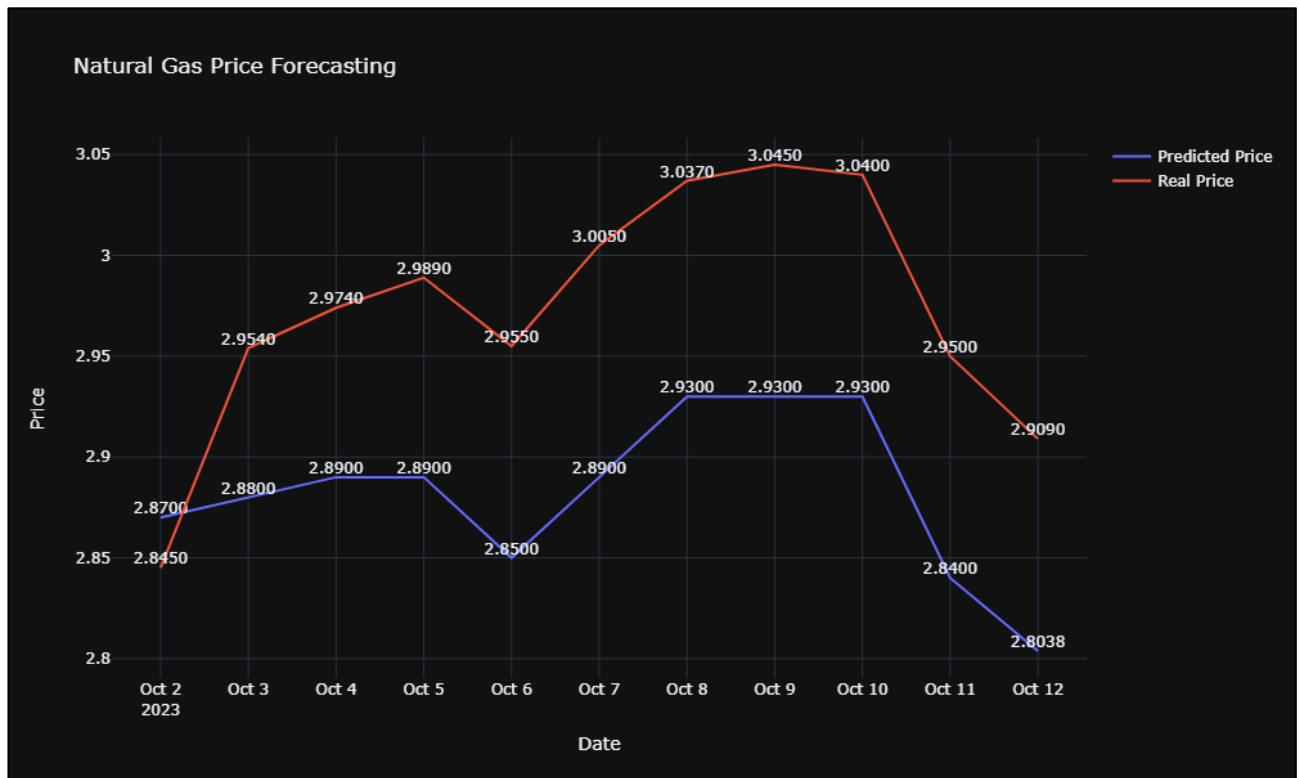


Рисунок 3.76 – Графік порівняння прогнозованих даних з реальними

За допомогою використання коду, зображеного на рисунку 3.77, виводяться всі значення DataFrame “merged\_data” та зберігає DataFrame у CSV-файлі. Весь DataFrame виводиться у табличному форматі, що допомагає візуалізувати дані (рис. 3.78).

```
print("All values:")
print(merged_data)

# Save to CSV
merged_data.to_csv("all_values.csv", index=False)
```

Рисунок 3.77 – Фрагмент коду



| All values: |       |            |          |            |
|-------------|-------|------------|----------|------------|
|             | index | date       | gp_pred  | real_price |
| 0           | 0     | 2023-10-02 | 2.870000 | 2.845      |
| 1           | 1     | 2023-10-03 | 2.880000 | 2.954      |
| 2           | 2     | 2023-10-04 | 2.890000 | 2.974      |
| 3           | 3     | 2023-10-05 | 2.890000 | 2.989      |
| 4           | 4     | 2023-10-06 | 2.850000 | 2.955      |
| 5           | 5     | 2023-10-07 | 2.890000 | 3.005      |
| 6           | 6     | 2023-10-08 | 2.930000 | 3.037      |
| 7           | 7     | 2023-10-09 | 2.930000 | 3.045      |
| 8           | 8     | 2023-10-10 | 2.930000 | 3.040      |
| 9           | 9     | 2023-10-11 | 2.840000 | 2.950      |
| 10          | 10    | 2023-10-12 | 2.803773 | 2.909      |

Рисунок 3.78 – Таблиця порівняння прогнозованих даних з реальними

Кінцевим етапом буде збереження DataFrame “merged\_data”, який містить прогнозовані дані з цінами на природній газ у CSV – файлі під назвою “predicted\_price.csv”, а також буде виконуватися збереження моделі LSTM у файлі з назвою “lstm\_model.h5”, фрагмент коду буде наведено на рисунку 3.79. Збереження моделі дозволяє використовувати її в майбутньому без необхідності повторного тренування, що може бути важливим для подальшого прогнозування чи використання моделі в інших застосунках.

```
# Збереження прогнозованих цін
merged_data.to_csv("predicted_price.csv", index=False)

# Збереження моделі LSTM
lstm_model.save("lstm_model.h5")
```

Рисунок 3.79 – Фрагмент коду

### 3.7 Висновки

У даному розділі було детально розглянуто ключові аспекти розробки моделі прогнозування. Починаючи з опису ознак та зчитування вхідного набору даних, далі провели етап очищення даних, щоб підготувати їх до подальшого аналізу.

Розвідувальний аналіз даних дозволив краще зрозуміти характеристики даних та їх взаємозв'язки. На етапі моделювання було визначено оптимальну модель ARIMA та навчено модель LSTM для прогнозування цін на природній газ.

Отримані результати свідчать про високу точність моделі LSTM: для тренувальних даних, значення MSE складає 0.088, а коефіцієнт детермінації ( $r^2$  score) – 0.981. Для тестових даних отримали MSE на рівні 0.008 та  $r^2$  score – 0.92. Це свідчить про високу ефективність та точність розробленої технології в прогнозуванні цін на природній газ. Також було проведено прогноз ціни на природній газ з 2 по 12 жовтня 2023 року.

## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія аналізу та прогнозування ціни на природній газ» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 4.1 [32].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

| Бали (за 5-ти бальною шкалою)           |  |   |   |   |  |
|---|--|---|---|---|--|
|   | 0  | 1   | 2   | 3   | 4  |
| <b>Технічна здійсненність концепції</b> |  |   |   |   |  |
| 1                                       | Достовірність концепції не підтверджена    | Концепція підтверджена експертними висновками | Концепція підтверджена розрахунками             | Концепція перевірена на практиці          | Перевірено працездатність продукту в реальних умовах |
| <b>Ринкові переваги (недоліки)</b>      |  |   |   |   |  |
| 2                                       | Багато аналогів на малому ринку            | Мало аналогів на малому ринку                 | Кілька аналогів на великому ринку               | Один аналог на великому ринку             | Продукт не має аналогів на великому ринку            |
| 3                                       | Ціна продукту значно вища за ціни аналогів | Ціна продукту дещо вища за ціни аналогів      | Ціна продукту приблизно дорівнює цінам аналогів | Ціна продукту дещо нижче за ціни аналогів | Ціна продукту значно нижче за ціни аналогів          |

## Продовження таблиці 4.1

| Ринкові переваги (недоліки) |   |   |   |   |  |
|-----------------------------|---|---|---|---|--|
| 4                           | Технічні та споживчі властивості продукту значно гірші, ніж в аналогів              | Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів                     | Технічні та споживчі властивості продукту на рівні аналогів     | Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів | Технічні та споживчі властивості продукту значно кращі, ніж в аналогів |
| 5                           | Експлуатаційні витрати значно вищі, ніж в аналогів                                  | Експлуатаційні витрати дещо вищі, ніж в аналогів  | Експлуатаційні витрати на рівні експлуатаційних витрат аналогів | Експлуатаційні витрати трохи нижчі, ніж в аналогів                    | Експлуатаційні витрати значно нижчі, ніж в аналогів                    |
| Ринкові перспективи         |   |   |   |   |  |
| 6                           | Ринок малий і не має позитивної динаміки  | Ринок малий, але має позитивну динаміку   | Середній ринок з позитивною динамікою                           | Великий стабільний ринок  | Великий ринок з позитивною динамікою                                   |
| 7                           | Активна конкуренція великих компаній на ринку                                       | Активна конкуренція   | Помірна конкуренція   | Незначна конкуренція  | Конкурентів немає  |
| Практична здійсненність     |   |   |   |   |  |
| 8                           | Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї                | Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців | Необхідне незначне навчання фахівців та збільшення їх штату     | Необхідне незначне навчання фахівців                                  | Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї |
| 9                           | Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні | Потрібні незначні фінансові ресурси. Джерела фінансування відсутні                        | Потрібні значні фінансові ресурси. Джерела фінансування є       | Потрібні незначні фінансові ресурси. Джерела фінансування є           | Не потребує додаткового фінансування                                   |

## Продовження таблиці 4.1

| Практична здійсненність |   |  |   |   |   |
|-------------------------|---|--|---|---|---|
| 10                      | Необхідна розробка нових матеріалів   | Потрібні матеріали, що використовують ся у військово промисловом у комплексі   | Потрібні дорогі матеріали   | Потрібні досяжні та дешеві матеріали  | Всі матеріали для реалізації ідеї відомі та давно використовують ся у виробництві   |
| 11                      | Термін реалізації ідеї більший за 10 років  | Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років  | Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років                       | Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років | Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років |
| 12                      | Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту | Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу | Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу | Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту  | Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту      |

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 4.2.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

| Критерії  | Експерт (ПІБ, посада) |    |    |
|---|-----------------------|----|----|
|   | 1                     | 2  | 3  |
|   | Бали:                 |    |    |
| 1. Технічна здійсненність концепції                         | 5                     | 5  | 4  |
| 2. Ринкові переваги (наявність аналогів)                    | 2                     | 2  | 3  |
| 3. Ринкові переваги (ціна продукту)                         | 3                     | 4  | 3  |
| 4. Ринкові переваги (технічні властивості)                  | 3                     | 3  | 3  |
| 5. Ринкові переваги (експлуатаційні витрати)                | 2                     | 2  | 2  |
| 6. Ринкові перспективи (розмір ринку)                       | 3                     | 3  | 3  |
| 7. Ринкові перспективи (конкуренція)                        | 2                     | 2  | 2  |
| 8. Практична здійсненність (наявність фахівців)             | 5                     | 5  | 5  |
| 9. Практична здійсненність (наявність фінансів)             | 2                     | 3  | 2  |
| 10. Практична здійсненність (необхідність нових матеріалів) | 4                     | 5  | 5  |
| 11. Практична здійсненність (термін реалізації)             | 3                     | 4  | 5  |
| 12. Практична здійсненність (розробка документів)           | 4                     | 5  | 4  |
| Сума балів  | 38                    | 43 | 41 |
| Середньоарифметична сума балів $CB_c$                       | 40,7                  |    |    |

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 4.3 [32].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

| Середньоарифметична сума балів СБ, розрахована на основі висновків експертів | Науково-технічний рівень та комерційний потенціал розробки |
|--|--|
| 41...48  | Високий  |
| 31...40  | Вище середнього  |
| 21...30  | Середній   |
| 11...20  | Нижче середнього   |
| 0...10   | Низький  |

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу та прогнозування ціни на природній газ»

становить 40,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

#### 4.2 Визначення рівня конкурентоспроможності розробки

В процесі визначення економічної ефективності науково-технічної розробки також доцільно провести прогноз рівня її конкурентоспроможності за сукупністю параметрів, що підлягають оцінюванню.

Одиничний параметричний індекс розраховуємо за формулою [32]:

$$q_i = \frac{P_i}{P_{\text{базі}}}, \quad (4.1)$$

де  $q_i$  – одиничний параметричний індекс, розрахований за  $i$ -м параметром;

$P_i$  – значення  $i$ -го параметра виробу;

$P_{\text{базі}}$  – аналогічний параметр базового виробу-аналога, з яким проводиться порівняння.

Загальні технічні та економічні характеристики розробки представлено в таблиці 4.4.

Таблиця 4.4 – Основні техніко-економічні показники аналога та розробки, що проектується

| Показники (параметри)                   | Одиниця вимірювання | Аналог | Проектований пристрій | Відношення параметрів нової розробки до аналога | Питома вага показника |
|---|---------------------|--------|-----------------------|---|-----------------------|
| Кількість доступних показників в запиті | од.                 | 15     | 24                    | 1,6   | 0,1                   |

Продовження таблиці 4.4

| Показники (параметри)                                    | Одиниця вимірювання | Аналог | Проектований пристрій | Відношення параметрів нової розробки до аналога | Питома вага показника |
|--|---------------------|--------|-----------------------|---|-----------------------|
| Кількість аналітичних функцій                            | од.                 | 32     | 54                    | 1,68  | 0,2                   |
| Кількість аналітичних модулів                            | од.                 | 8      | 11                    | 1,37  | 0,25                  |
| Швидкість пошуку та видачі запиту (тест)                 | мс                  | 0,9    | 0,45                  | 2   | 0,3                   |
| Швидкість та легкість доступу (дружність інтерфейсу)     | бали                | 6      | 8                     | 1,33  | 0,15                  |
| Експлуатаційні витрати (підтримка інформаційної системи) | грн                 | 4000   | 3100                  | 0,77  | 0,5                   |
| Вартість підписки (доступу до ресурсу)                   | грн                 | 300    | 250                   | 0,83  | 0,5                   |

Нормативні параметри оцінюємо показником, який отримує одне з двох значень: 1 – пристрій відповідає нормам і стандартам; 0 – не відповідає.

Груповий показник конкурентоспроможності за нормативними параметрами розраховуємо як добуток частинних показників за кожним параметром за формулою [32]:



$$I_{\text{НП}} = \prod_{i=1}^n q_i, \quad (4.2)$$

де  $I_{\text{НП}}$  – загальний показник конкурентоспроможності за нормативними параметрами;

$q_i$  – одиничний (частинний) показник за  $i$ -м нормативним параметром;

$n$  – кількість нормативних параметрів, які підлягають оцінюванню.

За нормативними параметрами розроблюваний пристрій відповідає вимогам ДСТУ, тому  $I_{\text{НП}} = 1$ .

Значення групового параметричного індексу за технічними параметрами визначаємо з урахуванням вагомості (частки) кожного параметра [32]:

$$I_{\text{ТП}} = \sum_{i=1}^n q_i \cdot \alpha_i, \quad (4.3)$$

де  $I_{\text{ТП}}$  – груповий параметричний індекс за технічними показниками (порівняно з виробом-аналогом);

$q_i$  – одиничний параметричний показник  $i$ -го параметра;

$\alpha_i$  – вагомість  $i$ -го параметричного показника,  $\sum_{i=1}^n \alpha_i = 1$ ;

$n$  – кількість технічних параметрів, за якими оцінюється конкурентоспроможність.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{\text{mn}} = 1,6 \cdot 0,1 + 1,68 \cdot 0,2 + 1,37 \cdot 0,25 + 2 \cdot 0,3 + 1,33 \cdot 0,15 = 1.64.$$

Груповий параметричний індекс за економічними параметрами розраховуємо за формулою [32]:

$$I_{EP} = \sum_{i=1}^m q_i \cdot \beta_i, \quad (4.4)$$

де  $I_{EP}$  – груповий параметричний індекс за економічними показниками;

$q_i$  – економічний параметр  $i$ -го виду;

$\beta_i$  – частка  $i$ -го економічного параметра,  $\sum_{i=1}^m \beta_i = 1$ ;

$m$  – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо аналіз параметрів згідно даних таблиці 4.4.

$$I_{EP} = 0,77 \cdot 0,5 + 0,83 \cdot 0,5 = 0,80.$$

На основі групових параметричних індексів за нормативними, технічними та економічними показниками розрахуємо інтегральний показник конкурентоспроможності за формулою [32]:

$$K_{INT} = I_{НП} \cdot \frac{I_{ТП}}{I_{EP}}, \quad (4.5)$$

$$K_{INT} = 1 \cdot 1,64 / 0,80 = 2,05.$$

Інтегральний показник конкурентоспроможності  $K_{INT} > 1$ , отже розробка переважає відомі аналоги за своїми техніко-економічними показниками.

### 4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія аналізу та прогнозування ціни на природній газ», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

### 4.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою [32]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.6)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$M_{ni}$  – місячний посадовий оклад конкретного дослідника, грн;

$t_i$  – число днів роботи конкретного дослідника, дн.;

$T_p$  – середнє число робочих днів в місяці,  $T_p=21$  дні.

$$Z_o = 18350,00 \cdot 34 / 21 = 29709,52 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на заробітну плату дослідників

| Найменування посади | Місячний посадовий оклад, грн | Оплата за робочий день, грн | Число днів роботи | Витрати на заробітну плату, грн |
|---------------------|-------------------------------|-----------------------------|-------------------|---------------------------------|
| Провідний фахівець  | 8650,00                       | 411,90                      | 32                | 13180,95                        |

Продовження таблиці 4.5

| Найменування посади   | Місячний посадовий оклад, грн | Оплата за робочий день, грн | Число днів роботи | Витрати на заробітну плату, грн |
|---|-------------------------------|-----------------------------|-------------------|---------------------------------|
| Керівник науково-дослідної роботи з формування інформаційної технології аналізу та прогнозування ціни | 18350,00                      | 873,81                      | 34                | 29709,52                        |
| Інженер-розробник інформаційних систем та технологій  | 17340,00                      | 825,71                      | 32                | 26422,86                        |
| Експерт функціонування ринкового середовища   | 20000,00                      | 952,38                      | 10                | 9523,81                         |
| Всього  |                               |                             |                   | 78837,14                        |

#### Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ( $Z_p$ ) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія аналізу та прогнозування ціни на природній газ» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.7)$$

де  $C_i$  – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{3M}}, \quad (4.8)$$

де  $M_M$  – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo  $M_M=6700,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [32];

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_p$  – середнє число робочих днів в місяці, приблизно  $T_p = 21$  дн;

$t_{зм}$  – тривалість зміни, год.

$$C_l = 6700,00 \cdot 1,10 \cdot 1,35 / (21 \cdot 8) = 59,22 \text{ (грн)}.$$

$$Z_{pl} = 59,22 \cdot 7,50 = 444,17 \text{ (грн)}.$$

Таблиця 4.6 – Величина витрат на основну заробітну плату робітників

| Найменування робіт  | Тривалість роботи, год | Розряд роботи | Тарифний коефіцієнт | Погодинна тарифна ставка, грн | Величина оплати на робітника грн |
|---|------------------------|---------------|---------------------|-------------------------------|----------------------------------|
| Підготовка робочого місця розробника інформаційної технології аналізу     | 7,50                   | 2             | 1,10                | 59,22                         | 444,17                           |
| Інсталяція програмного забезпечення середовища розробки і моделювання     | 6,20                   | 3             | 1,35                | 72,68                         | 450,63                           |
| Компіляція програмних блоків моделювання інформаційної технології аналізу | 6,00                   | 4             | 1,50                | 80,76                         | 484,55                           |

Продовження таблиці 4.6

| Найменування робіт  | Тривалість роботи, год | Розряд роботи | Тарифний коефіцієнт | Погодинна тарифна ставка, грн | Величина оплати на робітника грн |
|---|------------------------|---------------|---------------------|-------------------------------|----------------------------------|
| Підготовка локального серверного обладнання для проведення досліджень | 8,00                   | 5             | 1,70                | 91,53                         | 732,21                           |
| Налагодження програмних блоків збору інформації                       | 10,00                  | 5             | 1,70                | 91,53                         | 915,27                           |
| Набір бази даних ринкового середовища                                 | 12,00                  | 2             | 1,10                | 59,22                         | 710,68                           |
| Тестування інформаційної технології аналізу                           | 7,50                   | 2             | 1,10                | 59,22                         | 444,17                           |
| Всього  |                        |               |                     |                               | 4181,70                          |

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.9)$$

де  $H_{\text{дод}}$  – норма нарахування додаткової заробітної плати. Прийmemo 10%.

$$Z_{\text{дод}} = (78837,14 + 4181,70) \cdot 10 / 100\% = 8301,88 \text{ (грн)}.$$

### 4.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_n = (З_o + З_p + З_{\text{дод}}) \cdot \frac{H_{zn}}{100\%}, \quad (4.10)$$

де  $H_{zn}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$З_n = (78837,14 + 4181,70 + 8301,88) \cdot 22 / 100\% = 20090,56 \text{ (грн)}.$$

### 4.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія аналізу та прогнозування ціни на природній газ».

Витрати на матеріали ( $M$ ), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^n B_j \cdot Ц_{вj}, \quad (4.11)$$

де  $H_j$  – норма витрат матеріалу  $j$ -го найменування, кг;

$n$  – кількість видів матеріалів;

$Ц_j$  – вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$B_j$  – маса відходів  $j$ -го найменування, кг;

$C_{ej}$  – вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 4,0 \cdot 192,00 \cdot 1,06 - 0,0 \cdot 0,0 = 814,08 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на матеріали

| Найменування матеріалу, марка, тип, сорт | Ціна за 1 кг, грн | Норма витрат, кг | Величина відходів, кг | Ціна відходів, грн/кг | Вартість витраченого матеріалу, грн |
|--|-------------------|------------------|-----------------------|-----------------------|-------------------------------------|
| Папір офісний SCHOLZ A4 (500)            | 192,00            | 4,0              | 0,0                   | 0,0                   | 814,08                              |
| Папір для записів ЕКОС А5 (200)          | 116,00            | 4,0              | 0,0                   | 0,0                   | 491,84                              |
| Диск оптичний (CD-R)                     | 26,00             | 4,0              | 0,0                   | 0,0                   | 110,24                              |
| Органайзер офісний ЕКОС-ВІС              | 210,00            | 4,0              | 0,0                   | 0,0                   | 890,40                              |
| Канцелярське приладдя                    | 206,00            | 4,0              | 0,0                   | 0,0                   | 873,44                              |
| Картридж змінний HP 21 laser             | 980,00            | 2,0              | 0,0                   | 0,0                   | 2077,60                             |
| USB-пам'ять Traveler 128 GB              | 329,00            | 1,0              | 0,0                   | 0,0                   | 348,74                              |
| Кабель мережевий                         | 18,00             | 5,0              | 0,0                   | 0,0                   | 95,40                               |
| Всього                                   |                   |                  |                       |                       | 5701,74                             |



#### 4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі ( $K_e$ ), які використовують при проведенні НДР на тему «Інформаційна технологія аналізу та прогнозування ціни на природній газ», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_B = \sum_{j=1}^n H_j \cdot C_j \cdot K_j, \quad (4.12)$$

де  $H_j$  – кількість комплектуючих  $j$ -го виду, шт.;

$C_j$  – покупна ціна комплектуючих  $j$ -го виду, грн;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ ).

$$K_e = 1 \cdot 4350,00 \cdot 1,06 = 4611,00 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.8.

Таблиця 4.8– Витрати на комплектуючі

| Найменування комплектуючих | Кількість, шт. | Ціна за штуку, грн | Сума, грн |
|----------------------------|----------------|--------------------|-----------|
| Маршрутизатор LAP12        | 1              | 4350,00            | 4611,00   |
| Інтерфейси абонента        | 3              | 158,00             | 502,44    |
| Всього                     |                |                    | 5113,44   |

#### 4.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (4.13)$$

де  $C_i$  – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$  – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань устаткування.

$$B_{\text{спец}} = 35680,00 \cdot 1 \cdot 1,06 = 37820,80 \text{ (грн)}.$$

Отримані результати зведемо до таблиці 4.9:

Таблиця 4.9 – Витрати на придбання спецустаткування по кожному виду

| Найменування устаткування                             | Кількість, шт | Ціна за одиницю, грн | Вартість, грн |
|---|---------------|----------------------|---------------|
| Сервер – Компютер Expert PC Balance (I91F8H1S115E429) | 1             | 35680,00             | 37820,80      |
| Маршрутизатор VPN TP-Link ER7212PC                    | 1             | 9999,00              | 10598,94      |
| Всього  |               |                      | 48419,74      |

#### 4.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прг}} = \sum_{i=1}^k C_{i\text{прг}} \cdot C_{\text{прг}.i} \cdot K_i, \quad (4.14)$$

де  $C_{i\text{прг}}$  – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прг}.i}$  – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{\text{прг}} = 5800,00 \cdot 1 \cdot 1,06 = 6148,00 \text{ (грн)}.$$

Отримані результати зведемо до таблиці 4.10:

Таблиця 4.10 – Витрати на придбання програмних засобів по кожному виду

| Найменування програмного засобу  | Кількість, шт | Ціна за одиницю, грн | Вартість, грн   |
|--|---------------|----------------------|-----------------|
| Емулятор серверної інтернет-платформи для моделювання поведінки інформаційного ресурсу | 1             | 5800,00              | 6148,00         |
| Прикладний пакет розробки інформаційних систем (мови Java, C#, C++, Perl, PHP)         | 1             | 6820,00              | 7229,20         |
| Система керування реляційними базами даних MySQL                                       | 1             | 8459,00              | 8966,54         |
| Доступ до мережі Internet (високошвидкісний) грн/місяць                                | 2             | 299,00               | 633,88          |
| <b>Всього</b>  |               |                      | <b>22977,62</b> |

#### 4.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.15)$$

де  $Ц_{\text{б}}$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$  – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{в}}$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{\text{обл}} = (25125,00 \cdot 2) / (2 \cdot 12) = 2093,75 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.11.

Таблиця 4.11 – Амортизаційні відрахування по кожному виду обладнання

| Найменування обладнання                            | Балансова вартість, грн | Строк корисного використання, років | Термін використання обладнання, місяців | Амортизаційні відрахування, грн |
|--|-------------------------|-------------------------------------|---|---------------------------------|
| Персональний комп'ютер розробника                  | 25125,00                | 2                                   | 2                                       | 2093,75                         |
| Обчислювально-графічна система програмної розробки | 22498,00                | 2                                   | 2                                       | 1874,83                         |

## Продовження таблиці 4.11

| Найменування обладнання                                 | Балансова вартість, грн | Строк корисного використання, років | Термін використання обладнання, місяців | Амортизаційні відрахування, грн |
|---|-------------------------|-------------------------------------|---|---------------------------------|
| Пристрій виводу інформації HP-2100                      | 8765,00                 | 4                                   | 2                                       | 365,21                          |
| Ноутбук HP Laptop 15s-eq2037ua Natural Silver (422G7EA) | 32780,00                | 2                                   | 2                                       | 2731,67                         |
| Робоче місце розробника програмного забезпечення        | 9580,00                 | 5                                   | 2                                       | 319,33                          |
| Оргтехніка  | 8925,00                 | 4                                   | 2                                       | 371,88                          |
| Приміщення лабораторії                                  | 315000,00               | 25                                  | 2                                       | 2100,00                         |
| ОС Windows 11   | 7400,00                 | 3                                   | 2                                       | 411,11                          |
| Прикладний пакет Microsoft Office 2019                  | 6700,00                 | 3                                   | 2                                       | 372,22                          |
| <b>Всього</b>   |                         |                                     |   | <b>10640,00</b>                 |

#### 4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (4.16)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $C_e = 7,50$  грн;

$K_{впi}$  – коефіцієнт, що враховує використання потужності,  $K_{впi} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_e = 0,36 \cdot 220,0 \cdot 7,50 \cdot 0,95 / 0,97 = 594,00 \text{ (грн)}.$$

Проведені розрахунки зведемо до таблиці 4.12.

Таблиця 4.12 – Витрати на електроенергію

| Найменування обладнання                                 | Встановлена потужність, кВт | Тривалість роботи, год | Сума, грн |
|---|-----------------------------|------------------------|-----------|
| Персональний комп'ютер розробника                       | 0,36                        | 220,0                  | 594,00    |
| Обчислювально-графічна система програмної розробки      | 0,25                        | 220,0                  | 412,50    |
| Ноутбук HP Laptop 15s-eq2037ua Natural Silver (422G7EA) | 0,05                        | 220,0                  | 82,50     |
| Робоче місце розробника програмного забезпечення        | 0,10                        | 200,0                  | 150,00    |
| Пристрій виводу інформації HP-2100                      | 0,22                        | 2,5                    | 4,13      |
| Оргтехніка  | 0,50                        | 2,0                    | 7,50      |
| Сервер – Компютер Expert PC Balance (I91F8H1S115E429)   | 0,32                        | 120,0                  | 288,00    |
| Маршрутизатор VPN TP-Link ER7212PC                      | 0,03                        | 120,0                  | 27,00     |
| Маршрутизатор LAP12                                     | 0,01                        | 120,0                  | 9,00      |
| Інтерфейси абонента                                     | 0,01                        | 120,0                  | 9,00      |
| Всього  |                             |                        | 1583,63   |

#### 4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія аналізу та прогнозування ціни на природній газ» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за

договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cb} = (Z_o + Z_p) \cdot \frac{H_{cb}}{100\%}, \quad (4.17)$$

де  $H_{cb}$  – норма нарахування за статтею «Службові відрядження», прийmemo  $H_{cb} = 20\%$ .

$$B_{cb} = (78837,14 + 4181,70) \cdot 20 / 100\% = 16603,77 \text{ (грн)}.$$

#### **4.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації**

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (4.18)$$

де  $H_{cn}$  – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo  $H_{cn} = 30\%$ .

$$B_{cn} = (78837,14 + 4181,70) \cdot 30 / 100\% = 24905,65 \text{ (грн)}.$$

### 4.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (З_{\text{o}} + З_{\text{р}}) \cdot \frac{H_{\text{ів}}}{100\%}, \quad (4.19)$$

де  $H_{\text{ів}}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{\text{ів}} = 55\%$ .

$$I_{\text{в}} = (78837,14 + 4181,70) \cdot 55 / 100\% = 45660,36 \text{ (грн)}.$$

### 4.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (З_{\text{o}} + З_{\text{р}}) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.20)$$



де  $B_{нзв}$  – норма нарахування за статтею «Накладні (загально виробничі) витрати», приймемо  $B_{нзв} = 100\%$ .

$$B_{нзв} = (78837,14 + 4181,70) \cdot 100 / 100\% = 83018,84 \text{ (грн)}.$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія аналізу та прогнозування ціни на природній газ» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{дод} + Z_n + M + K_v + B_{спец} + B_{прг} + A_{обл} + B_e + B_{св} + B_{сн} + I_v + B_{нзв}. \quad (4.21)$$

$$\begin{aligned} B_{заг} &= 78837,14 + 4181,70 + 8301,88 + 20090,56 + 5701,74 + 5113,44 + 48419,74 + \\ &22977,62 + 10640,00 + 1583,63 + 16603,77 + 24905,65 + 45660,36 + 83018,84 = \\ &= 376036,07 \text{ (грн)}. \end{aligned}$$

Загальні витрати  $ZB$  на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{заг}}{\eta}, \quad (4.22)$$

де  $\eta$  – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо  $\eta=0,9$ .

$$ZB = 376036,07 / 0,9 = 417817,86 \text{ (грн)}.$$

#### 4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Інформаційна технологія аналізу та прогнозування ціни на природній газ» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

$\Delta N$  – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються:

| Показник                              | 1-й рік | 2-й рік | 3-й рік | 4-й рік |
|---------------------------------------|---------|---------|---------|---------|
| Збільшення кількості споживачів, осіб | 5200    | 8000    | 10000   | 7500    |

$N$  – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, прийmemo 32000 осіб;

$C_o$  – вартість послуги у році до впровадження інформаційної системи, прийmemo 200,00 грн;

$\pm \Delta C_o$  – зміна вартості послуги від впровадження результатів, прийmemo 47,60 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta \Pi_i$  для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [32, 33]:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.23)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту).  
Прийmemo  $\rho = 40\%$ ;

$\vartheta$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році  $\vartheta = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (47,60 \cdot 32000,00 + 247,60 \cdot 5200) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 765190,41 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (47,60 \cdot 32000,00 + 247,60 \cdot 13200) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1304443,40 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (47,60 \cdot 32000,00 + 247,60 \cdot 23200) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1978509,64 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (47,60 \cdot 32000,00 + 247,60 \cdot 30700) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 2484059,32 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків  $\Pi\Pi$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.24)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,15$ ;

$t$  – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 765190,41/(1+0,15)^1 + 1304443,40/(1+0,15)^2 + 1978509,64/(1+0,15)^3 + \\ &+ 2484059,32/(1+0,15)^4 = 665382,97 + 986346,62 + 1300902,21 + 1420268,98 = \\ &= 4372900,78 \text{ (грн)}. \end{aligned}$$

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot ЗВ, \quad (4.25)$$

де  $k_{инв}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $k_{инв} = 2$ ;

$ЗВ$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 417817,86 грн.

$$PV = k_{инв} \cdot ЗВ \cdot PV = 2 \cdot 417817,86 = 835635,71 \text{ (грн)}.$$

Абсолютний економічний ефект  $E_{абс}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV, \quad (4.26)$$

де  $ПП$  – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 4372900,78 грн;

$PV$  – теперішня вартість початкових інвестицій, 835635,71 грн.

$$E_{abc} = III - PV = 4372900,78 - 835635,71 = 3537265,06 \text{ (грн)}.$$

Внутрішня економічна дохідність інвестицій  $E_e$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_B = \sqrt[T_{ж}] \left( 1 + \frac{E_{abc}}{PV} \right) - 1, \quad (4.27)$$

де  $E_{abc}$  – абсолютний економічний ефект вкладених інвестицій, 3537265,06 грн;

$PV$  – теперішня вартість початкових інвестицій, 835635,71 грн;

$T_{ж}$  – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_B = \sqrt[4] \left( 1 + \frac{E_{abc}}{PV} \right) - 1 = (1 + 3537265,06 / 835635,71)^{1/4} = 0,51.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій  $\tau_{min}$ :

$$\tau_{min} = d + f, \quad (4.27)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d=0,1$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,3.

$\tau_{min} = 0,1 + 0,3 = 0,4 < 0,51$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_v$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційна технологія аналізу та прогнозування ціни на природній газ» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_v}, \quad (4.28)$$

де  $E_v$  – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 0,51 = 1,95 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

#### 4.5 Висновки

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу та прогнозування ціни на природній газ» становить 40,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні рівня конкурентоспроможності, згідно узагальненого коефіцієнту конкурентоспроможності розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 2,05 рази.

Також термін окупності становить 1,95 року, що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення роботи за темою «Інформаційна технологія аналізу та прогнозування ціни на природній газ».

## ВИСНОВКИ

У даній магістерській кваліфікаційній роботі була ретельно розглянута та обґрунтована актуальність теми «Інформаційна технологія аналізу та прогнозування ціни на природній газ». Результати проведеного аналізу об'єкта дослідження вказують на необхідність використання технологій машинного навчання для досягнення найкращих результатів у цій сфері.

Було проведено огляд об'єкта дослідження, визначено важливість прогнозування ціни на природній газ та переваги технологій машинного навчання. Аналіз аналогів, таких як Bloomberg, Oil Price, NYMEX та система Kaggle, підтверджує актуальність обраних методів.

Розглянуто сутність прогнозування ціни на природній газ та обрано технології машинного навчання. Також підготовлено необхідний датасет для подальшого аналізу, що стане основою для розвитку методології та моделей прогнозування у наступних частинах роботи.

Детально розглянуто процес визначення оптимальних налаштувань для моделі та обрання мови програмування та середовища розробки, що суттєво впливає на ефективність прогнозування. Також було визначено конкретний вид машинного навчання та проведено відбір та докладний опис моделей, які використовуються у процесі машинного навчання.

Проведено аналіз опису обраної моделі для прогнозування даних, вибору мови програмування та середовища розробки, а також визначено конкретний вид машинного навчання для подальших досліджень. Детально розглянуті моделі ARIMA та LSTM, які відіграють ключову роль у прогнозуванні даних, з наданням опису їхньої структури та основних принципів функціонування.

В контексті даної роботи також було визначено, що використання мови програмування Python та середовища розробки Kaggle є оптимальними для реалізації обраних моделей та вирішення поставленої задачі. Цей етап виявляється вирішальним для подальшої розробки інформаційної технології аналізу та



прогнозування ціни на природній газ, оскільки правильний вибір налаштувань та моделей машинного навчання визначає успішність системи прогнозування.

Детально розглянуто ключові аспекти розробки прогностичної моделі, розпочинаючи від опису ознак та зчитування вхідних даних і завершуючи процесом навчання моделі LSTM. Отримані результати свідчать про високу точність цієї моделі: для тренувальних даних значення MSE складає 0.088, а коефіцієнт детермінації ( $r^2$  score) – 0.981. Для тестових даних отримали MSE на рівні 0.008 та  $r^2$  score – 0.92. Це підтверджує високу ефективність та точність розробленої технології в прогнозуванні цін на природній газ. Також проведено прогноз ціни на природній газ з 2 по 12 жовтня 2023 року.

Економічна частина роботи підкреслює вигідність та конкурентоспроможність розробленого програмного продукту. Розрахунки витрат, чистого прибутку, періоду окупності та економічного ефекту вказують на реальність та ефективність використання даної розробки.

У результаті проведення дослідження у рамках магістерської кваліфікаційної роботи з використанням передових методів машинного навчання розроблено інформаційну технологію аналізу та прогнозування ціни на природній газ. Розроблена технологія виявилася ефективною, підвищуючи точність прогнозування цінових трендів на ринку природного газу. Таким чином, поставлені завдання магістерської роботи були успішно виконані, що підкреслює вагомий внесок у розвиток інформаційних технологій в аналізі та прогнозуванні цін на природній газ.

За даною темою дослідження зроблено доповідь на тему «Інформаційна технологія аналізу та прогнозування ціни на природній газ» на LIII Науково-технічній конференції факультету інтелектуальних інформаційних технологій та автоматизації ВНТУ з публікацією тез (м. Вінниця, 2023-2024 рр.) [1].

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ладуб О.В., Козачко О.М. Інформаційна технологія аналізу та прогнозування ціни на природній газ. *ЛІІІ Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2023-2024)*. Вінниця, 2023-2024. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/19634/16258> (дата звернення 08.12.2023)
2. Faramawy, S., Zaki, T., & Sakr, A. E. Natural gas origin, composition, and processing: A review. *Journal of Natural Gas Science and Engineering*. 2016. Т. 34. С. 34-54. URL: <https://doi.org/10.1016/j.jngse.2016.06.030> (дата звернення 10.09.2023)
3. Li, Xiao-Sen, et al. "Investigation into gas production from natural gas hydrate: A review.". *Applied Energy*. 2016. Т. 172. С. 286-322. URL: <https://doi.org/10.1016/j.apenergy.2016.03.101> (дата звернення 12.09.2023)
4. Jana R. K., Ghosh I. A residual driven ensemble machine learning approach for forecasting natural gas prices: Analyses for pre-and during-COVID-19 phases. *Annals of Operations Research*. 2022. С. 1-22. URL: <https://link.springer.com/article/10.1007/s10479-021-04492-4> (дата звернення 12.09.2023)
5. Yaseen, Zaher Mundher, et al. "Artificial intelligence based models for stream-flow forecasting: 2000–2015.". *Journal of Hydrology*. 2015 Т. 530 С. 829-844. URL: <https://doi.org/10.1016/j.jhydrol.2015.10.038> (дата звернення 14.09.2023)
6. Cabaneros S. M., Calautit J. K., Hughes B. R. A review of artificial neural network models for ambient air pollution prediction. *Environmental Modelling & Software*. 2019. Т. 119. С. 285-304. URL: <https://doi.org/10.1016/j.envsoft.2019.06.014> (дата звернення 14.09.2023)
7. Sarker I. H. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*. 2021. Т. 2. №. 3. С. 160. URL: <https://link.springer.com/article/10.1007/s42979-021-00592-x> (дата звернення 15.09.2023)

8. Hopkins E. et al. Machine learning tools, algorithms, and techniques. *Journal of Self-Governance and Management Economics*. 2022. Т. 10. №. 1. С. 43-55. URL: <https://www.ceeol.com/search/article-detail?id=1030033> (дата звернення 15.09.2023)
9. Сайт Bloomberg. URL: <https://www.bloomberg.com/> (дата звернення 18.09.2023)
10. Сайт OilPrice. URL: <https://oilprice.com/> (дата звернення 18.09.2023)
11. Сайт CME Group. URL: <https://www.cmegroup.com/> (дата звернення 19.09.2023)
12. Siddharth Kulkarni Kaggle Notebook «Natural Gas Price Prediction using Time Series». URL: <https://www.kaggle.com/code/mykeysid10/natural-gas-price-prediction-using-time-series> (дата звернення 21.09.2023)
13. Мінфін “Ціни на газ на Нью-Йоркській товарній біржі (NYMEX)”. URL: <https://index.minfin.com.ua/ua/markets/gas/nymex/> (дата звернення 23.09.2023)
14. Ladub Oleksandr Kaggle Dataset «Natural Gas Price Actually 2023». URL: <https://www.kaggle.com/datasets/laduboleksandr/natural-gas-price-actually-2023> (дата звернення 24.09.2023)
15. Шостя С. П. Машинне навчання. *Матеріали VII Всеукраїнської науково-практичної Інтернет-конференції молодих учених та студентів Полтава, 24-25 листопада 2021 р.* Полтава С. 146. URL: [http://elcat.pnpu.edu.ua/docs/IICTE\\_2021.pdf#page=148](http://elcat.pnpu.edu.ua/docs/IICTE_2021.pdf#page=148) (дата звернення: 01.10.2023)
16. Sulstonov S. IMPORTANCE OF PYTHON PROGRAMMING LANGUAGE IN MACHINE LEARNING. *International Bulletin of Engineering and Technology*. 2023. Т. 3. №. 9. С. 28-30. URL: <https://internationalbulletins.com/intjour/index.php/ibet/article/view/1020/658#> (дата звернення 02.10.2023)
17. Siegmund N., Ruckel N., Siegmund J. Dimensions of software configuration: on the configuration context in modern software development. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering*

*Conference and Symposium on the Foundations of Software Engineering*. 2020. С. 338-349. URL: <https://doi.org/10.1145/3368089.3409675> (дата звернення 02.10.2023)

18. Сайт Kaggle. URL: <https://www.kaggle.com/> (дата звернення 03.10.2023)

19. Janiesch C., Zschech P., Heinrich K. Machine learning and deep learning. *Electronic Markets*. 2021. Т. 31. №. 3. С. 685-695. URL: <https://link.springer.com/article/10.1007/s12525-021-00475-2> (дата звернення 04.10.2023)

20. Kontopoulou V. I. et al. A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet*. 2023. Т. 15. №. 8. С. 255. URL: <https://doi.org/10.3390/fi15080255> (дата звернення 05.10.2023)

21. Sagheer A., Kotb M. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*. 2019. Т. 323. С. 203-213. URL: <https://doi.org/10.1016/j.neucom.2018.09.082> (дата звернення 08.10.2023)

22. Bernard J., Bernard J. Python data analysis with pandas. *Python Recipes Handbook: A Problem-Solution Approach*. 2016. С. 37-48. URL: [https://doi.org/10.1007/978-1-4842-0241-8\\_5](https://doi.org/10.1007/978-1-4842-0241-8_5) (дата звернення 08.10.2023)

23. Hao J., Ho T. K. Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*. 2019. Т. 44. №. 3. С. 348-361. URL: <https://doi.org/10.3102/1076998619832248> (дата звернення 09.10.2023)

24. Morgenthaler S. Exploratory data analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*. 2009. Т. 1. №. 1. С. 33-44. URL: <https://doi.org/10.1002/wics.2> (дата звернення 15.10.2023)

25. Goloborodko A. A. et al. Pyteomics—a Python framework for exploratory data analysis and rapid software prototyping in proteomics. *Journal of The American Society for Mass Spectrometry*. 2013. Т. 24. №. 2. С. 301-304. URL: <https://doi.org/10.1007/s13361-012-0516-6> (дата звернення 18.10.2023)

26. Podo L., Velardi P. Plotly. plus, an Improved Dataset for Visualization Recommendation. *Proceedings of the 31st ACM International Conference on Information*

- & *Knowledge Management*. 2022. С. 4384-4388. URL: <https://doi.org/10.1145/3511808.3557669> (дата звернення 25.10.2023)
27. Du L. et al. Bayesian optimization based dynamic ensemble for time series forecasting. *Information Sciences*. 2022. Т. 591. С. 155-175. URL: <https://doi.org/10.1016/j.ins.2022.01.010> (дата звернення 30.10.2023)
28. Horváth L., Kokoszka P., Rice G. Testing stationarity of functional time series. *Journal of Econometrics*. 2014. Т. 179. №. 1. С. 66-82. URL: <https://doi.org/10.1016/j.jeconom.2013.11.002> (дата звернення 01.11.2023)
29. Raschka S., Patterson J., Nolet C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*. 2020. Т. 11. №. 4. С. 193. URL: <https://doi.org/10.3390/info11040193> (дата звернення 02.11.2023)
30. Yamak P. T., Yujian L., Gadosey P. K. A comparison between arima, lstm, and gru for time series forecasting. *Proceedings of the 2019 2nd international conference on algorithms, computing and artificial intelligence*. 2019. С. 49-55. URL: <https://doi.org/10.1145/3377713.3377722> (дата звернення 03.11.2023)
31. DiPietro R., Hager G. D. Deep learning: RNNs and LSTM. *Handbook of medical image computing and computer assisted intervention*. Academic Press. 2020. С. 503-519. URL: <https://doi.org/10.1016/B978-0-12-816176-0.00026-0> (дата звернення 03.11.2023)
32. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт : Вінниця: ВНТУ, 2021. 42 с.
33. Кавецький В. В., Козловський В. О., Причепка І.В., Економічне обґрунтування інноваційних рішень : приктикум / за ред. В. В. Кавецький. Вінниця: ВНТУ, 2016. 113 с.

## Додаток А

Міністерство освіти і науки України  
Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

\_\_\_\_\_ д.т.н., проф. Віталій МОКІН

«\_\_» \_\_\_\_\_ 2023 року

ТЕХНІЧНЕ ЗАВДАННЯ  
на магістерську кваліфікаційну роботу  
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЦІНИ НА  
ПРИРОДНІЙ ГАЗ  
08-34.МКР.002.00.000 ТЗ

Керівник: к.т.н., доц. каф. САІТ

\_\_\_\_\_ Олексій КОЗАЧКО

«\_\_» \_\_\_\_\_ 2023 р.

Розробив ст. гр. ЗІСТ-22м

\_\_\_\_\_ Олександр ЛАДУБ

«\_\_» \_\_\_\_\_ 2023 р.

Вінниця 2023

1. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №\_\_ по ВНТУ від «\_\_» \_\_\_\_\_2023р., та індивідуальне завдання на МКР, затверджене протоколом №\_\_ засідання кафедри САІТ від «\_\_» \_\_\_\_\_ 2023р.

2. Джерела розробки:

1) Siddharth Kulkarni Kaggle Notebook «Natural Gas Price Prediction using Time Series». URL: <https://www.kaggle.com/code/mykeysid10/natural-gas-price-prediction-using-time-series>

2) Sarker I. H. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*. 2021. Т. 2. №. 3. С. 160.

3. Мета і призначення роботи.

Метою дослідження є підвищення точності прогнозування ціни на природній газ за рахунок використання методів машинного навчання.

4. Вихідні дані для проведення робіт.

Ladub Oleksandr Kaggle Dataset « Natural Gas Price Actually 2023». URL: <https://www.kaggle.com/datasets/laduboleksandr/natural-gas-price-actually-2023>

5. Методи дослідження.

В даній роботі використовуються методи часових рядів та машинного навчання для прогнозування ціни на природній газ.

6. Етапи роботи і терміни їх виконання:

а) Аналіз предметної області \_\_\_\_\_ – \_\_\_\_\_

б) Вибір оптимальних налаштувань \_\_\_\_\_ – \_\_\_\_\_

в) Розроблення інформаційної технології аналізу та прогнозування ціни на природній газ \_\_\_\_\_ – \_\_\_\_\_

г) Економічна частина \_\_\_\_\_ – \_\_\_\_\_

д) Оформлення матеріалів до захисту МКР \_\_\_\_\_ – \_\_\_\_\_

7. Очікувані результати та порядок реалізації.

Отримані результати по прогнозуванню ціни на природній газ з 02.10.2023 р. по 12.10.2023 р.

8. Вимоги до розробленої документації.

Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання магістерських кваліфікаційних робіт для студентів спеціальності 126 «Інформаційні системи та технології» (освітня програма «Інформаційні технології аналізу даних та зображень»).

9. Порядок приймання роботи.

Публічний захист «\_\_» \_\_\_\_\_ 2023 р.

Початок розробки «\_\_» \_\_\_\_\_ 2023 р.

Граничні терміни виконання МКР «\_\_» \_\_\_\_\_ 2023 р.

Розробив студент групи ЗІСТ-22м \_\_\_\_\_ Олександр ЛАДУБ

## Додаток Б

Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія аналізу та прогнозування ціни на природній газ»

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра САІТ

## Показники звіту подібності Unicheck

Оригінальність 92% Схожість 8%

Аналіз звіту подібності (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку



(підпис)

Сергій ЖУКОВ

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



(підпис)

Олександр ЛАДУБ

Керівник роботи



(підпис)

Олексій КОЗАЧКО



Додаток В  
Лістинг програми

```
import warnings
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
warnings.filterwarnings("ignore")

import tensorflow as tf
import os
from tensorflow.keras import layers, models
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

from sklearn.metrics import mean_squared_error, r2_score
import statsmodels.api as sm
import plotly.express as px
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

import numpy as np
from sklearn.model_selection import RandomizedSearchCV
from tensorflow.keras.wrappers.scikit_learn import KerasRegressor
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

import tensorflow as tf

import plotly.graph_objects as go
data = pd.read_csv("../input/natural-gas-price-actually-2023/ngpf_data_actually.csv")
```

```
data.info()

data['Day'] = pd.to_datetime(data['Day'], format = "%d/%m/%Y")
data.head(5)

data.tail(5)

# Renaming columns
data = data.rename({'Day': 'date', 'Price in Dollars per Million Btu': 'gas_price'}, axis = 1)

# Setting Date as index
data = data.set_index('date')

data.tail(5)

print(data.isnull().sum())

# There is one missing value in the data, we fill it with previous day's price.

data = data.fillna(method = 'pad')
print(data.isnull().sum())

fig = px.line(data, title = 'Natural Gas Spot Prices', template = 'plotly_dark')
fig.show()

fig = px.histogram(data, x = "gas_price", template = 'plotly_dark')
fig.show()

def test_stationarity(timeseries):
# Determing rolling statistics
```

```
rolmean = timeseries.rolling(25).mean()
rolstd = timeseries.rolling(25).std()

# Plot rolling statistics:
plt.figure(figsize = (20,10))
orig = plt.plot(timeseries, color='blue',label='Original')
mean = plt.plot(rolmean, color='red', label='Rolling Mean')
std = plt.plot(rolstd, color='black', label = 'Rolling Std')
plt.legend(loc='best')
plt.title('Rolling Mean & Standard Deviation')
plt.show(block=False)

# Perform Dickey-Fuller test:
print('Results of Dickey-Fuller Test:')
dfctest = adfuller(timeseries, autolag='AIC')
dfcoutput = pd.Series(dfctest[0:4], index=['Test Statistic', 'p-value', '#Lags Used', 'Number
of Observations Used'])
for key, value in dfctest[4].items():
dfcoutput['Critical Value (%)' % key] = value

print(dfcoutput)

# Check for stationarity
p_value = dfctest[1]
significance_level = 0.05
if p_value <= significance_level:
print("Ряд є стаціонарним (p-value <= 0.05)")
else:
print("Ряд не є стаціонарним (p-value > 0.05)")
```

```
test_stationarity(data)

ts_sqrt = np.sqrt(data)
expwighted_avg = ts_sqrt.ewm(halflife = 25).mean()

ts_sqrt_ewma_diff = ts_sqrt - expwighted_avg
test_stationarity(ts_sqrt_ewma_diff)

ts_sqrt_diff = ts_sqrt - ts_sqrt.shift()

plt.figure(figsize = (20,10))
plt.plot(ts_sqrt_diff)
plt.show()

ts_sqrt = np.sqrt(data)
ts_sqrt_diff = ts_sqrt - ts_sqrt.shift()
ts_sqrt_diff.dropna(inplace = True)
test_stationarity(ts_sqrt_diff)

data = data.sort_values(by = 'date')

train = data['1997-01-06': '2022-12-31'] # 6 Jan 1997 to 31 Dec 2022
test = data['2023-01-01': '2023-10-01'] # 1 Jan 2023 to 1 Oct 2023

print("Length of Train Data: ", len(train))
print("Length of Test Data: ", len(test))

train.head(5)

train.tail(5)
```

```
test.head(5)
```

```
test.tail(5)
```

```
ax = train.plot(figsize=(20, 10), color='b')
```

```
test.plot(ax=ax, color='black')
```

```
# Отримання останнього індекса з train для визначення межі між train та test
```

```
train_last_index = train.index[-1]
```

```
# Малюєм вертикальну лінію, розділяючи train від test
```

```
plt.axvline(x=train_last_index, color='r', linestyle='--', linewidth=2)
```

```
plt.legend(['train set', 'test set'])
```

```
plt.show()
```

```
plt.rcParams.update({'figure.figsize':(9,7), 'figure.dpi':120})
```

```
# Original Series
```

```
fig, axes = plt.subplots(4, 2, sharex=True)
```

```
axes[0, 0].plot(data.gas_price); axes[0, 0].set_title('Original Series')
```

```
plot_acf(data.gas_price, ax=axes[0, 1])
```

```
# 1st Differencing
```

```
axes[1, 0].plot(data.gas_price.diff()); axes[1, 0].set_title('1st Order Differencing')
```

```
plot_acf(data.gas_price.diff().dropna(), ax=axes[1, 1])
```

```
# 2nd Differencing
```

```
axes[2, 0].plot(data.gas_price.diff().diff()); axes[2, 0].set_title('2nd Order Differencing')
```

```

plot_acf(data.gas_price.diff().diff().dropna(), ax=axes[2, 1])

# 3rd Differencing
axes[3, 0].plot(data.gas_price.diff().diff().diff()); axes[3, 0].set_title('3rd Order
Differencing')
plot_acf(data.gas_price.diff().diff().diff().dropna(), ax=axes[3, 1])

plt.show()

plt.rcParams.update({'figure.figsize':(9,3), 'figure.dpi':120})

fig, axes = plt.subplots(1, 2, sharex=True)
axes[0].plot(data.gas_price.diff()); axes[0].set_title('1st Differencing')
axes[1].set(ylim=(0,5))
plot_pacf(data.gas_price.diff().dropna(), ax=axes[1])

plt.show()

plt.rcParams.update({'figure.figsize':(9,3), 'figure.dpi':120})

fig, axes = plt.subplots(1, 2, sharex=True)

axes[0].plot(data.gas_price.diff()); axes[0].set_title('1st Differencing')
axes[1].set(ylim=(0,1.2))
plot_acf(data.gas_price.diff().dropna(), ax=axes[1])

plt.show()

model = sm.tsa.arima.ARIMA(train, order = (1, 2, 1))
arima_model = model.fit()

```

```

print(arima_model.summary())

yp_train = arima_model.predict(start=0, end=(len(train)-1))
yp_test = arima_model.predict(start=0, end=(len(test)-1))

print("Train Data:\nMean Square Error: {}".format(mean_squared_error(train,
yp_train)))
print("\nTest Data:\nMean Square Error: {}".format(mean_squared_error(test, yp_test)))

print(yp_test)

print(test)

print(yp_train)

print(train)

# Функція для побудови моделі, яку будемо оптимізувати
def build_lstm_model(units, layers, dropout, learning_rate):
    model = Sequential()
    model.add(LSTM(units=units, input_shape=(slot, 1), return_sequences=True,
activation='relu'))
    for _ in range(layers - 1):
        model.add(LSTM(units=units, return_sequences=True, activation='relu'))
    model.add(LSTM(units=units, return_sequences=False, activation='relu'))
    model.add(Dense(units=1))
    model.compile(loss='mean_squared_error',
optimizer=tf.keras.optimizers.Adam(learning_rate=learning_rate))
    return model

```

```
# Підготовка даних
slot = 10
x_train = []
y_train = []

for i in range(slot, len(train)):
    x_train.append(train.iloc[i-slot:i, 0])
    y_train.append(train.iloc[i, 0])

x_train, y_train = np.array(x_train), np.array(y_train)

x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
print(x_train.shape, y_train.shape)

# Створення моделі для оптимізації
model = KerasRegressor(build_fn=build_lstm_model, verbose=0)

# Визначення гіперпараметрів, які ви хочете оптимізувати та їх діапазонів
param_dist = {
    'units': [25, 50, 75],
    'layers': [1, 2, 3],
    'dropout': [0.1, 0.2, 0.3],
    'learning_rate': [0.001, 0.01, 0.1]
}

# Виконання RandomizedSearchCV для пошуку найкращих параметрів
random_search = RandomizedSearchCV(model, param_distributions=param_dist,
n_iter=10, scoring='neg_mean_squared_error', cv=3)
random_search.fit(x_train, y_train)
```



```
# Отримання найкращих параметрів і результатів
best_params = random_search.best_params_
best_score = -random_search.best_score_

print("Найкращі параметри:", best_params)
print("Найкращий результат (MSE):", best_score)

early_stopping = tf.keras.callbacks.EarlyStopping(monitor = 'loss', patience = 10)

history = lstm_model.fit(x_train, y_train, epochs = 100,
                        batch_size = 64,
                        verbose = 1, shuffle = False,
                        callbacks = [early_stopping])

yp_train = lstm_model.predict(x_train)
a = pd.DataFrame(yp_train)
a.rename(columns = {0: 'gp_pred'}, inplace = True);
a.index = train.iloc[slot:].index
train_compare = pd.concat([train.iloc[slot:], a], 1)

train_compare.head(5)

train_compare.tail(5)

plt.figure(figsize = (15, 5))
plt.plot(train_compare['gas_price'], color = 'red', label = "Actual Natural Gas Price")
plt.plot(train_compare['gp_pred'], color = 'blue', label = 'Predicted Price')
plt.title("Natural Gas Price Prediction on Train Data")
plt.xlabel('Time')
plt.ylabel('Natural gas price')
```

```
plt.legend(loc = 'best')
plt.show()

dataset_total = pd.concat((train, test), axis = 0)
inputs = dataset_total[len(dataset_total) - len(test)- slot:].values
inputs = inputs.reshape(-1, 1)

x_test = []
y_test = []
for i in range (slot, len(test)+slot): #Test+15
    x_test.append(inputs[i-slot:i, 0])
    y_test.append(train.iloc[i, 0])

x_test, y_test = np.array(x_test), np.array(y_test)

x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
pred_price = lstm_model.predict(x_test)

b = pd.DataFrame(pred_price)
b.rename(columns = {0: 'gp_pred'}, inplace = True);
b.index = test.index
test_compare = pd.concat([test, b], 1)

test_compare.head(5)

test_compare.tail(5)

plt.figure(figsize = (15,5))
plt.plot(test_compare['gas_price'], color = 'red', label = "Actual Natural Gas Price")
plt.plot(test_compare['gp_pred'], color = 'blue', label = 'Predicted Price')
```

```

plt.title("Natural Gas Price Prediction On Test Data")
plt.xlabel('Time')
plt.ylabel('Natural gas price')
plt.legend(loc = 'best')
plt.show()

mse_train = mean_squared_error(train_compare['gas_price'], train_compare['gp_pred'])
mse_test = mean_squared_error(test_compare['gas_price'], test_compare['gp_pred'])

r2_train = r2_score(train_compare['gas_price'], train_compare['gp_pred'])
r2_test = r2_score(test_compare['gas_price'], test_compare['gp_pred'])

print("Train Data:\nMSE: {} \nR Square: {}".format(mse_train, r2_train))
print("\nTest Data:\nMSE: {} \nR Square: {}".format(mse_test, r2_test))

plt.plot(history.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train'], loc='upper right')
plt.show()

forecast = pd.DataFrame({'date': pd.date_range(start = '10/2/2023', end = '10/12/2023')})
# 2 October to 12 October

inputs = test[len(test) - slot: ].values

for i in range(slot, len(forecast)):
    inputs = inputs.T
    inputs = np.reshape(inputs, (inputs.shape[0], inputs.shape[1], 1))

```



```

# Існуючий код для прогнозу
forecast = pd.DataFrame({'date': pd.date_range(start='10/2/2023', end='10/12/2023')}) #
2 October to 12 October

inputs = test[len(test) - slot:].values

for i in range(slot, len(forecast)):
    inputs = inputs.T
    inputs = np.reshape(inputs, (inputs.shape[0], inputs.shape[1], 1))
    pred_price = lstm_model.predict(inputs[:, i - slot:i])
    inputs = np.append(inputs, pred_price)
    inputs = np.reshape(inputs, (inputs.shape[0], 1))

forecast['gp_pred'] = inputs
forecast = forecast.set_index('date')

# Об'єднання прогнозованих та реальних даних
merged_data = pd.merge(forecast, real_data, on='date', how='left')

# Скидання індексу для побудови графіку
merged_data.reset_index(inplace=True)

# Побудова графіку з реальними та прогнозованими даними
fig = go.Figure()

# Додаємо прогнозовані дані
fig.add_trace(go.Scatter(x=merged_data['date'], y=merged_data['gp_pred'],
                        mode='lines+text', # змінено 'markers' на 'text'
                        name='Predicted Price',

```

```

text=[f"{price:.4f}" for price in merged_data['gp_pred']],
textposition="top center", # розташування тексту над точкою
hoverinfo='skip'))

# Додаємо реальні дані
fig.add_trace(go.Scatter(x=merged_data['date'], y=merged_data['real_price'],
                        mode='lines+text', # змінено 'markers' на 'text'
                        name='Real Price',
                        text=[f"{price:.4f}" for price in merged_data['real_price']],
                        textposition="top center", # розташування тексту над точкою
                        hoverinfo='skip'))

# Налаштування графіку
fig.update_layout(title='Natural Gas Price Forecasting',
                  xaxis_title='Date',
                  yaxis_title='Price',
                  template='plotly_dark',
                  height=600,
                  width=1000,
                  xaxis=dict(tickmode='linear')) # Змінено tickmode на 'linear'

# Показ графіку
fig.show()

print("All values:")
print(merged_data)

# Save to CSV
merged_data.to_csv("all_values.csv", index=False)

# Збереження прогнозованих цін

```

```
merged_data.to_csv("predicted_price.csv", index=False)
```

```
# Збереження моделі LSTM
```

```
lstm_model.save("lstm_model.h5")
```

Додаток Г

## **ІЛЮСТРАТИВНА ЧАСТИНА**

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЦІНИ НА  
ПРИРОДНІЙ ГАЗ**

Нормоконтроль: к.т.н., доцент

\_\_\_\_\_ Сергій ЖУКОВ

«\_\_» \_\_\_\_\_ 2023 р.

Вінниця 2023



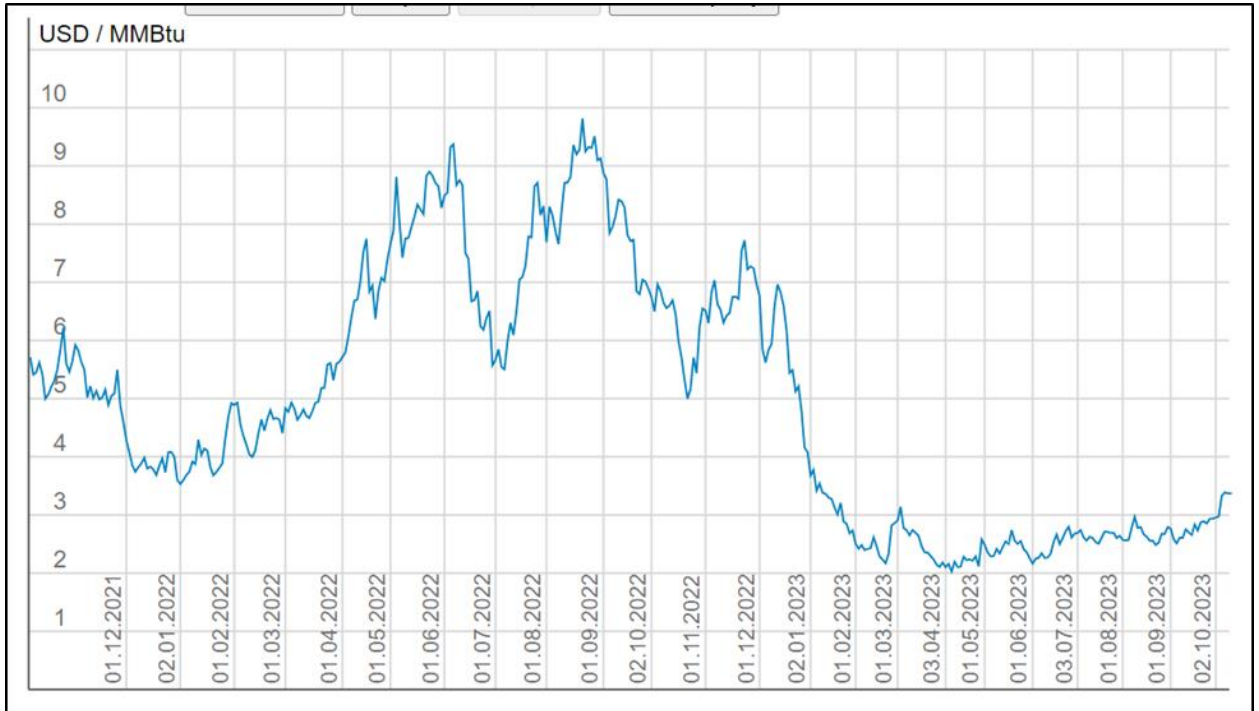


Рисунок Г.1 – Графік динаміки ціни

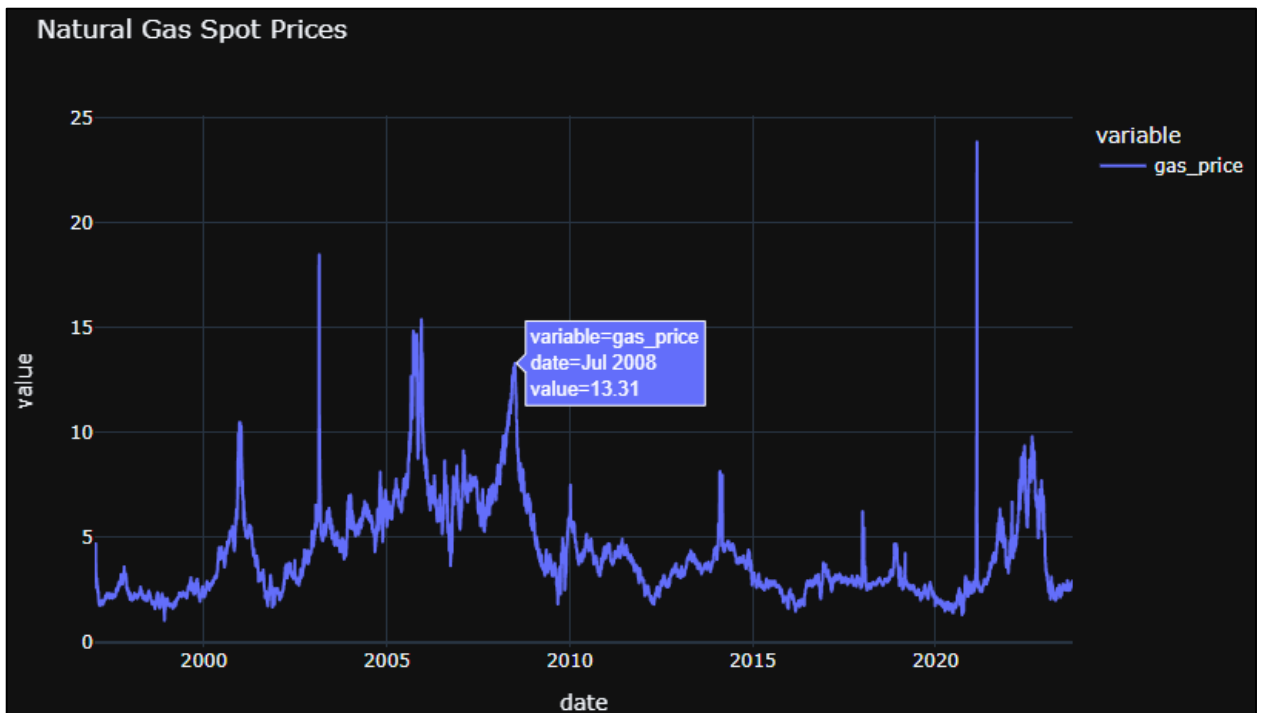


Рисунок Г.2 – Лінійна діаграма з аналізом ціни

```

Results of Dickey-Fuller Test:
Test Statistic      -4.037120
p-value             0.001228
#Lags Used          8.000000
Number of Observations Used  6888.000000
Critical Value (1%)  -3.431300
Critical Value (5%)  -2.861960
Critical Value (10%) -2.566993
dtype: float64
Ряд є стаціонарним (p-value <= 0.05)

```

Рисунок Г.3 – Результат перевірки часового ряду на стаціонарність

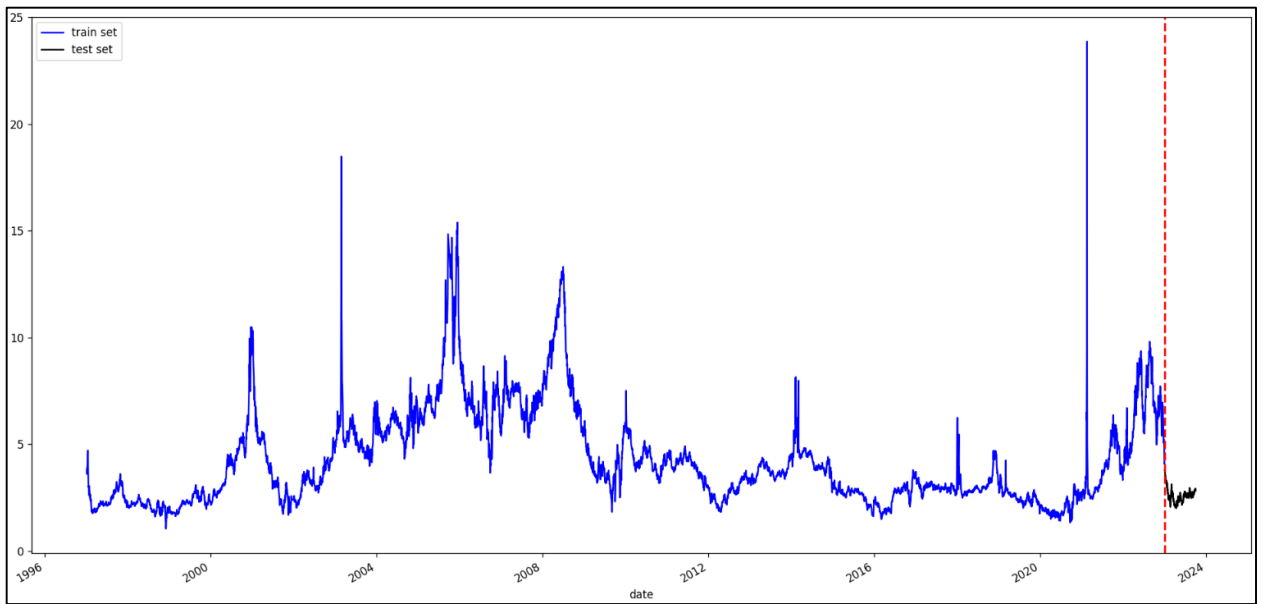


Рисунок Г.4 – Графік відображення навчальної та тестової вибірки

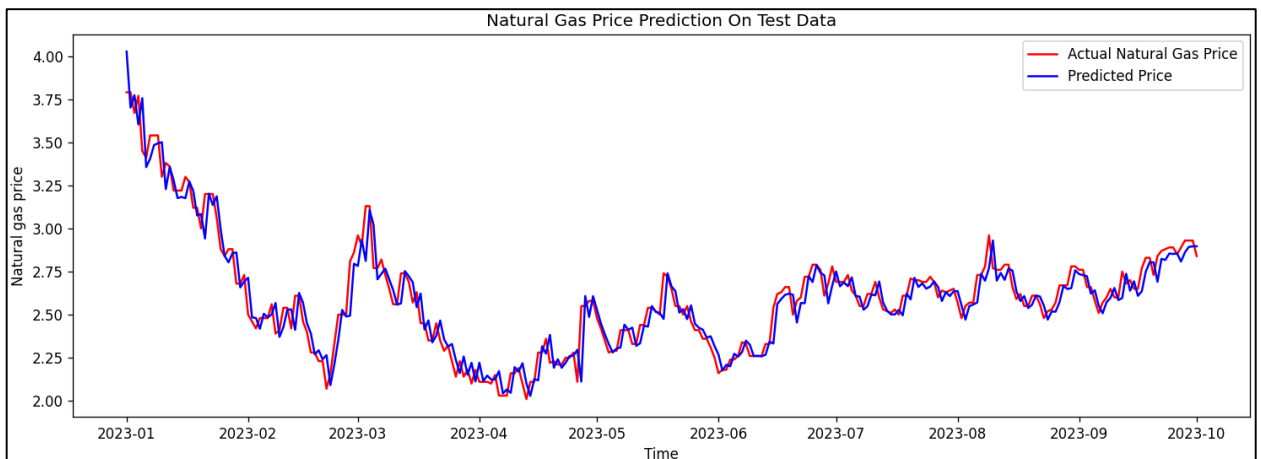


Рисунок Г.5 – Графік порівнянь прогнозованих та фактичних значень на тестових даних

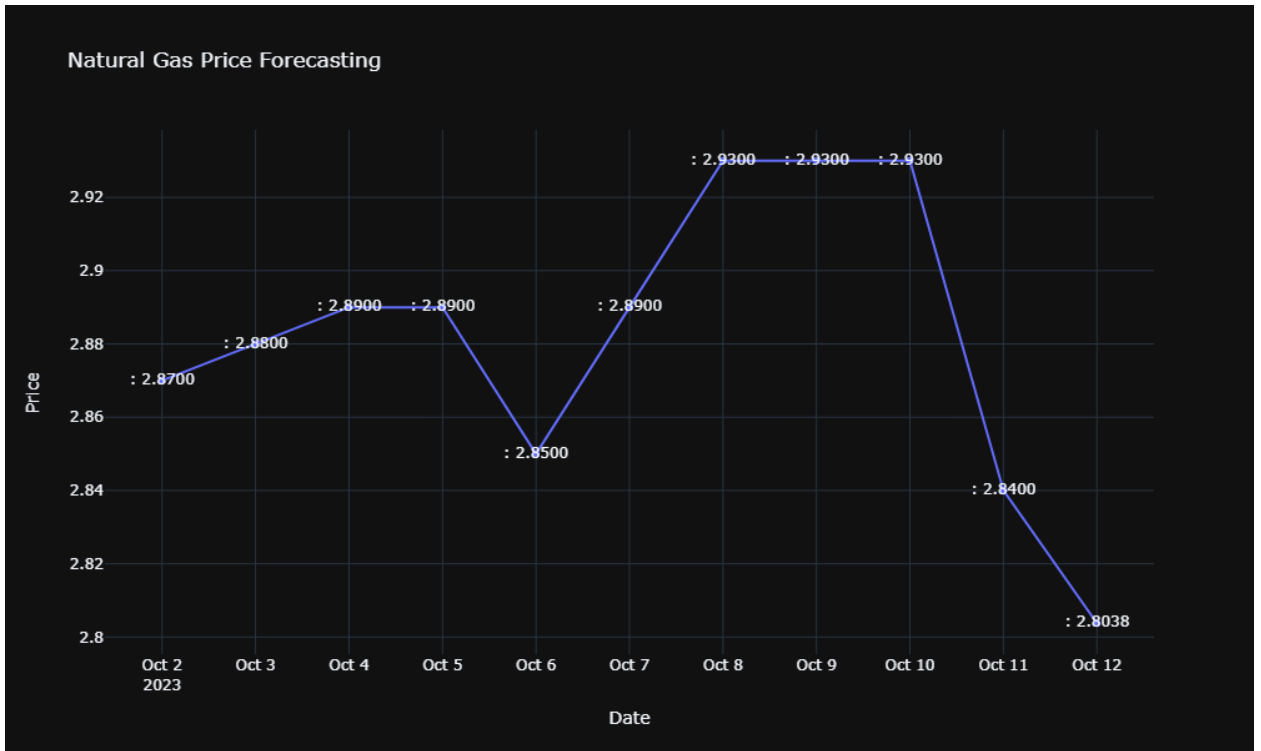


Рисунок Г.6 – Графік прогнозу

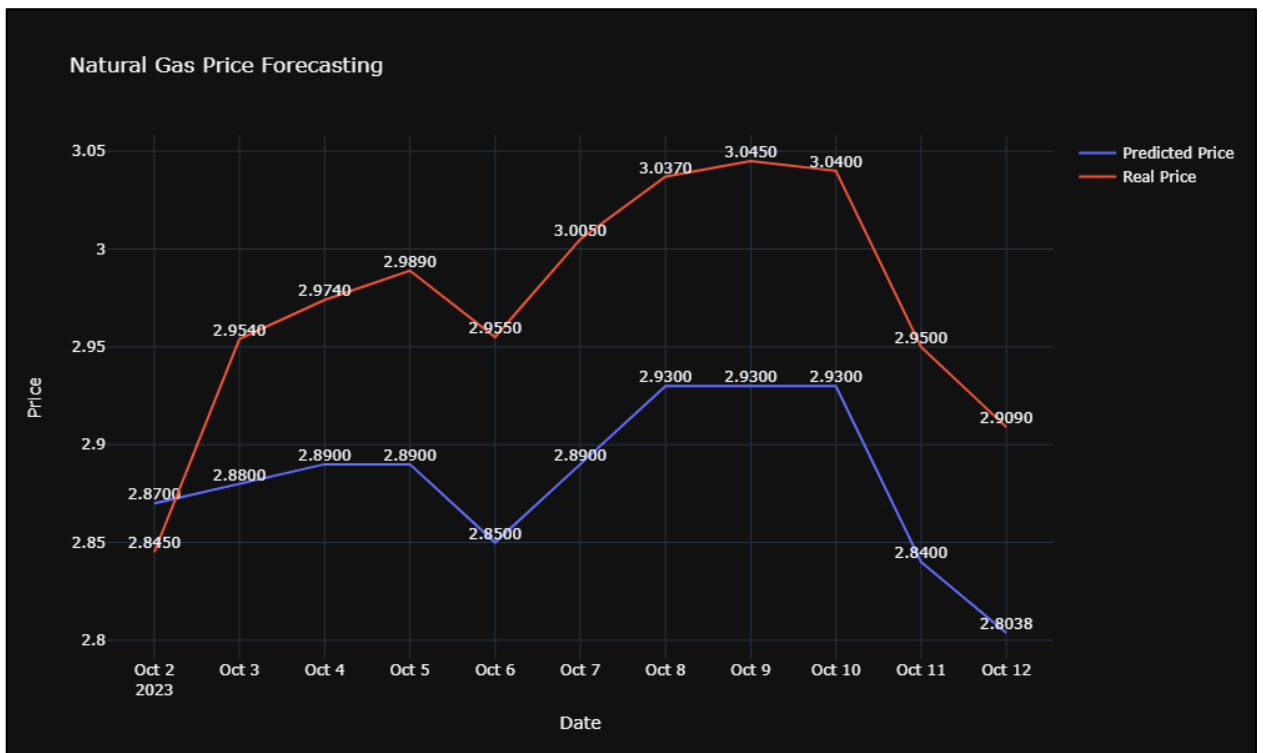


Рисунок Г.7 – Графік порівняння прогнозованих даних з реальними

| Назва моделі | Точність прогнозу на навчальній вибірці | Точність прогнозу на тестовій вибірці |
|--------------|---|---------------------------------------|
| ARIMA        | MSE = 0.139                             | MSE = 0.29                            |
| LSTM         | MSE = 0.088<br>R2_score = 0.98          | MSE = 0.008<br>R2_score = 0.92        |

Рисунок Г.8 - Таблиця порівняння прогнозованих даних з реальними