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

Комплексна магістерська кваліфікаційна робота на тему:

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ СТАНУ АТМОСФЕРНОГО ПОВІТРЯ МІСТА ВІННИЦІ. ЧАСТИНА 2. ПРОГНОЗУВАННЯ ДАНИХ»

Виконав: студент 2 курсу, групи 2ІСТ-22м спеціальності 126 «Інформаційні системи та технології»
_____Владислав ПІНЧУК
Керівник: к.т.н., доц. каф. САІТ
_____Сергій ЖУКОВ
«30» _____11_____ 2023 р.
Опонент: к.т.н., доц. каф. КН
_____ Ігор АРСЕНЮК
«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 – ІІ-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
«04» 09 2023 р.

ЗАВДАННЯ
НА КОМПЛЕКСНУ МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
СТУДЕНТУ
Пінчуку Владиславу Павловичу

1. Тема роботи: «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних», керівник роботи: Сергій ЖУКОВ к.т.н., доц. каф. САІТ, затверджені наказом закладу вищої освіти від «18» 09 2023 року №247
2. Строк подання студентом роботи «50» 11 2023 року
3. Вихідні дані до роботи:
   - Kaggle датасет, на основі даних громадського моніторингу міста Вінниці: Air quality in Vinnytsia 2023 (September) (https://www.kaggle.com/datasets/pirogovskiy/air-quality-in-vinnytsia-2023-september.
4. Зміст текстової частини:
   - загальна характеристика об'єкту дослідження;
   - опис моделей, які будуть використовуватись для передбачення даних;
   - вибір мови програмування, бібліотек та середовища розробки;
   - розроблення інформаційної технології та її застосування на реальних даних
   - економічна частина.
5. Перелік ілюстративного матеріалу:

- графік динаміки значень частинок PM2.5;
- графік динаміки середньодобових значень частинок PM2.5;
- графіки прогнозу моделей;
- графік аномалій
- результати роботи моделей.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Наталія БУРЄННІКОВА, д. е. н., проф. каф. ЕПВМ	04.11.23	15.11.23

7. Дата видачі завдання «04» 09 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	04.09	20.09	вик
2	Вибір оптимальних інформаційних технологій	20.09	05.10	вик
3	Розроблення інформаційної технології	05.10	15.10	вик
4	Тренування та тестування моделей машинного навчання	15.10	25.10	вик
5	Економічна частина	25.10	15.11	вик
6	Оформлення матеріалів до захисту МКР	15.11	30.11	вик

Студент Владислав ПІНЧУК

Керівник роботи Сергій ЖУКОВ

АНОТАЦІЯ

УДК 004.09

Пінчук В. П. Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних. Комплексна магістерська кваліфікаційна робота зі спеціальності 126 – інформаційні системи та технології, освітньо-професійна програма – інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2023. 91 с.
На укр. мові. Бібліогр.: 24 назв; рис.: 61; табл.: 11.

В даній магістерській кваліфікаційній роботі розроблено інформаційну систему прогнозування якості атмосферного повітря міста Вінниці. Під час виконання магістерської кваліфікаційної роботи використовувались дані громадського моніторингу. Проведено огляд існуючих методів та технологій для аналізу даних.

Здійснено розвідувальний аналіз даних, знайдено аномальні дані. Виконано прогнозування даних з використанням трьох моделей машинного навчання: Prophet, LSTM та ARIMA.

Об’єкт досліджень – процес моніторингу якості атмосферного повітря у місті Вінниця.

Ілюстративна частина складається з 8 плакатів з результатами роботи.

У розділі економічної частини розглянуто питання про доцільність розробки та впровадження інформаційної системи прогнозування якості атмосферного повітря.

Ключові слова: інформаційна технологія, моніторинг, повітря, якість повітря, прогноз, датасет, модель.

ABSTRACT

Pinchuk V. P. Information technology of analysis and forecasting of the state of atmospheric air in the city of Vinnytsia. Part 2. Data forecasting. Complex master's thesis on specialty 126 – information systems and technologies, educational and professional program – information technologies of analysis data and images. Vinnytsia: VNTU, 2023. 91p.

In Ukrainian speech Bibliography: 24 titles; Fig.: 61; tab.: 11.

In this master's thesis, an information system for forecasting the quality of atmospheric air in the city of Vinnytsia was developed. Data from public monitoring were used during the master's qualification work. An overview of existing methods and technologies for data analysis was conducted.

Exploratory data analysis was carried out, anomalous data were found. Data forecasting was performed using three machine learning models: Prophet, LSTM, and ARIMA.

The object of research is the process of monitoring atmospheric air quality in the city of Vinnytsia.

The illustrative part consists of 8 drawings with the results of the work.

In the section of the economic part, the issue of the feasibility of developing and implementing an information system for forecasting the quality of atmospheric air is considered.

Keywords: information technology, monitoring, air, air quality, forecast, dataset, model.

ЗМІСТ

ВСТУП

Актуальність теми. Якість атмосферного повітря безпосередньо впливає на якість життя людини та має важливе значення для нормального функціонування її організму. На сучасному етапі відзначається стрімке збільшення впливу людської діяльності на оточуюче середовище, зокрема на атмосферний шар планети. Розширення автотранспорту, розвиток промисловості, неконтрольована рубка лісів та використання технологій, що не відповідають екологічним стандартам, призвели до погіршення якості повітря у багатьох містах України та світу. У зв'язку з цим виникла проблема поліпшення та відновлення якості атмосферного повітря. Для вирішення цієї проблеми важливо вдосконалити системи державного моніторингу якості повітря та впровадити нові технології у цій сфері.

Мета і завдання роботи. Метою даної магістерської кваліфікаційної роботи є підвищення точності прогнозування якості атмосферного повітря міста Вінниці, використовуючи дані громадського моніторингу. Для досягнення поставленої мети необхідно розв’язати наступні завдання:

- повести огляд існуючих систем;
- виконати збір даних та сформувати датасет;
- підготувати дані для подальшої роботи;
- провести розвідувальних аналіз даних;
- побудувати моделі та виконати прогнозування;
- оцінити результати роботи моделей.

Об’єктом дослідження магістерської кваліфікаційної роботи є процес розроблення інформаційної технології аналізу та прогнозування повітря у місті Вінниця.

Предметом дослідження магістерської кваліфікаційної роботи є інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці.

Новизна отриманих результатів полягає в подальшому розвитку інформаційної технології прогнозування якості атмосферного повітря у місті Вінниці. Це можливо за рахунок сучасних алгоритмів машинного навчання, які мають змогу працювати з часовими рядами, що і дало змогу підвищити точність прогнозованих результатів.

Практичне значення. Отримані результати є цінними для моніторингу екологічного забруднення атмосферного повітря і для осіб, які мають алергічну реакцію на пил. Крім того, вони можуть служити основою для розробки ефективних стратегій зменшення впливу шкідливих речовин на атмосферу та підтримки здоров'я громади.

Апробація результатів магістерської кваліфікаційної роботи. Результати роботи апробовані на LIII Всеукраїнській науково-технічній конференції підрозділів Вінницького національного технічного університету (2023-2024).

Публікації результатів магістерської кваліфікаційної роботи. За даною темою дослідження зроблено доповідь на тему «Інформаційна технологія аналізу та прогнозування PM2.5 а атмосферному повітрі міста Вінниці на основі моделей Prophet, LSTM та ARIMA» на LIII Всеукраїнській науково-технічній конференції підрозділів Вінницького національного технічного університету (2023-2024) з публікацією тез [1].

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ'ЄКТУ ДОСЛІДЖЕНЬ

1.1 Опис проблематики забруднення атмосферного повітря

Якість атмосферного повітря безпосередньо впливає на комфорт та здоров'я людей, граючи важливу у їх житті. На сьогоднішній час спостерігається стрімке зростання антропогенного впливу на навколишнє середовище, зокрема і на атмосферу планети. Цей вплив зумовлений збільшенням кількості транспортних засобів, розвитком промисловості, та використанням неекологічних технологій, неконтрольованою вирубкою лісів тощо. Ці фактори призвели до того, що у багатьох регіонах планети спостерігається погіршення якості атмосферного повітря.

До прикладу лише 13 країн і територій світу минулого року мали "здорову" якість повітря. Лише шість країн – Австралія, Естонія, Фінляндія, Гренада, Ісландія та Нова Зеландія – і сім територій у Тихому океані та Карибському басейні, відповідають вимогам ВООЗ. Сім країн – Чад, Ірак, Пакистан, Бахрейн, Бангладеш, Буркіна-Фасо, Кувейт та Індія – мали погану якість повітря, яка значно перевищувала рекомендації із середнім рівнем забруднення повітря понад 50 мікрограмів на кубічний метр. Встановлено, що рівень забруднення, приблизно в 90% перевірених територій перевищував рекомендації Всесвітньої організації охорони здоров'я [2-4]. На рисунку 1.1 – Зображено інтерактивну мапу індексу якості повітря у світі.

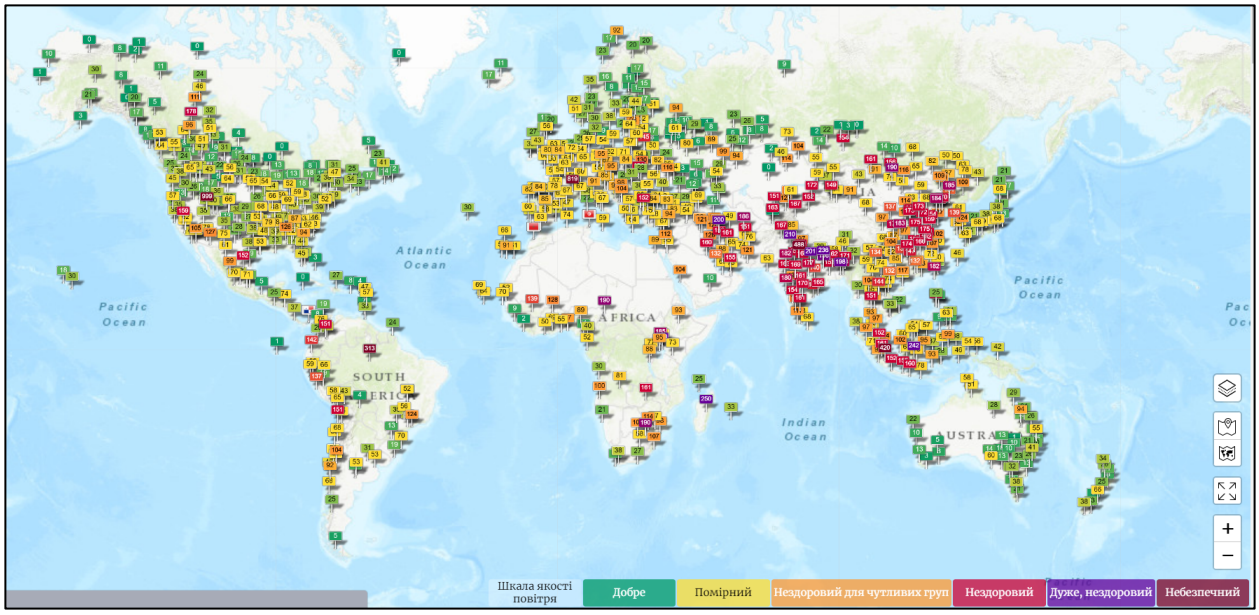


Рисунок 1.1 – Інтерактивна мапа індексу якості повітря у світі

Дослідження розглядало дрібні тверді частинки PM2,5. PM2.5 представляє собою мікрочастинки з діаметром менше 2,5 мікрометрів, які вважаються одними з найбільш небезпечних забруднювачів, що призводять до погіршення якості повітря. При вдиханні дані частинки проникають глибоко в легеневу тканину , звідки можуть потрапити в кров. Частинки утворюються внаслідок спалювання викопного палива, пилових бур та лісових пожеж. Таким чином, території з високим рівнем промислової діяльності та частими заторами на дорогах, ймовірно, мають підвищені концентрації PM2.5, що також може сприяти забрудненню повітря та негативно впливати на здоров'я людей. Щороку мільйони людей помирають внаслідок проблем зі здоров’ям, які були викликані забрудненням повітря.

Згідно із звітом State of Global Air 2020, забруднення повітря є четвертим за розміром фактором ризику передчасної смерті у всьому світі, призводячи до понад 6,67 мільйонів смертей внаслідок впливу забрудненого повітря. Більш детальний аналіз глобального рейтингу факторів ризику за загальною кількістю смертей у 2019 році доступний на рисунку 1.2.

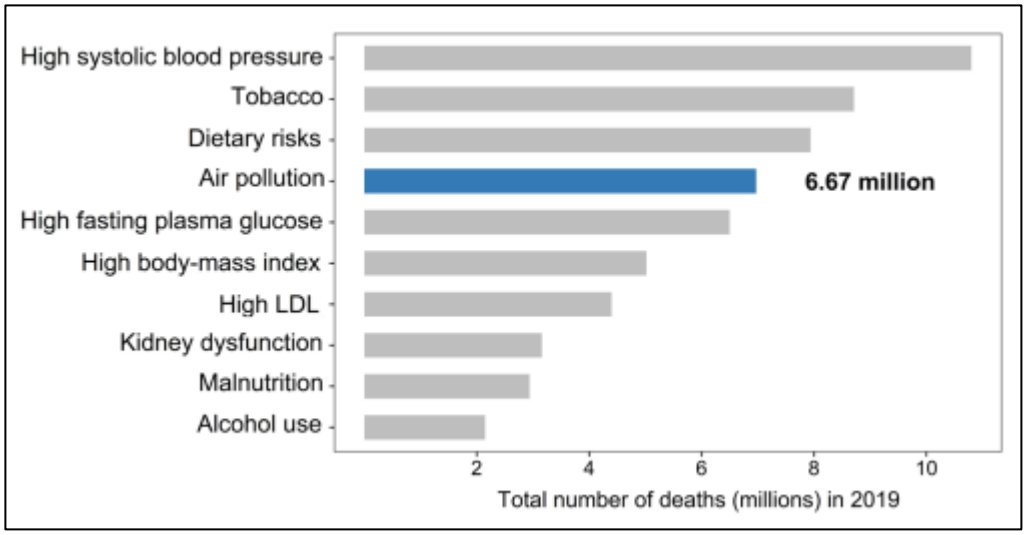


Рисунок 1.2 – Рейтинг факторів ризику за загальною кількістю смертей за 2019 рік

Виходячи з факторів описаних вище, є важливим забезпечити ефективне та точне прогнозування якості повітря для забезпечення заздалегідь попередження населення та поліпшення процесів прийняття рішень щодо необхідних заходів для зменшення можливих наслідків.

1.2 Моніторинг якості повітря міста Вінниці

Основними забруднювачами повітря в Україні є підприємства чорної металургії, енергетики, вугільної, хімічної та нафтохімічної промисловості. Однак значний вплив мають також викиди від теплових електростанцій та автотранспорту, кількість яких з кожним роком зростає.

На сьогоднішній день, в Україні, виміри стану атмосферного повітря проводяться Гідрометцентром та різноманітними громадськими організаціями. Виміри здійснюються в 39 містах на 129 стаціонарних постах моніторингу. Програма обов’язкового моніторингу атмосферного повітря включає сім

забруднюючих речовин: пил, формальдегід, оксид вуглецю, двооксид азоту, свинець, двооксид сірки та бензапірен [5].

Тверді частинки (PM) , англійською Particulare Matter — дрібний пил, який складається з найдрібніших твердих і рідких частинок, які розділені на групи в залежності від фракцій. Частинки діаметром до 10 мкм (PM10) називаються твердими частинками. Ці частинки розміром від 3 до 10 мкм осідають в носі і гортані. Частки розміром близько 2,5 мкм (PM2.5) потрапляють в легені при вдиху. Частки розміром менше 1 мкм (PM1) потрапляють на альвеоли і далі в кровоносну систему. Існує тісний взаємозв'язок між впливом невеликої кількості частинок (≤10 мк) та збільшенням смертності та болю, щодня або час від часу. І навпаки, із зменшенням концентрації дрібних і дрібних частинок рівень смертності також зменшується.

Оксид вуглецю (або чадний газ) – це газ, що заважає крові поглинати кисень. Це може спричинити значне зменшення постачання серця киснем, особливо у людей, які страждають на серцеві захворювання. CO з'єднується з гемоглобіном у 250 разів легше, ніж кисень — утворюється карбоксигемоглобін (HbCO), який не може транспортувати кисень, наслідком чого є гіпоксія,утворення вільних радикалів та пошкодження мембранних структур клітин. Оксид вуглецю присутній усюди при спалюванні будь якого виду палива : викопного, мінерального, газу тощо. Як приклад при згорянні 1 кг палива у бензинову двигуні внутрішнього згоряння у легковому автомобілі середнього класу утворюється до 460 г оксиду вуглецю. Тому в першу чергу саме оксид вуглецю повинні знешкодити нейтралізатори і каталізатори сучасних автомобілів [6-7].

Діоксид азоту (NO_2) та оксид азоту (NO) представляють собою хімічні сполуки, які, при короткочасній концентрації, що перевищує 200 мікрограмів на кубічний метр, вважаються токсичними газами, спричиняючи значне запалення дихальних шляхів. NO_2 виступає як основне джерело нітратних аерозолів, які формують невеликі фракції частинок. Головними джерелами викидів NO_2 є

процеси згоряння, такі як опалення, виробництво електроенергії, транспортні засоби та судна. Епідеміологічні дослідження свідчать, що тривалий вплив NO2 збільшує симптоми бронхіту у дітей з астмою, а також пов'язується із зниженням функції легенів. При взаємодії оксидів азоту з водяною парою (з особливою увагою при високій атмосферній вологості) утворюються пари азотних та азотистих кислот, які можуть руйнувати легені та справляти значний негативний вплив на флору і фауну.

Діоксид сірки (SO2) – це газ без кольору із різким запахом, який утворюється під час спалювання викопного палива, такого як вугілля та нафта, а також при переробці мінеральної руди, що містить сірку. Важливо відзначити, що при спалюванні газу SO2 не утворюється. Значні кількості SO2 виникають при використанні сірчаних викопних палив (вугілля, дрова, пелети) для опалення в домогосподарствах, виробництва електроенергії (вугільні теплоелектростанції, теплові електростанції) та автотранспорту (дизельні двигуни). SO2 може впливати на функцію дихальної системи та легенів, викликаючи подразнення очей. Його наявність може призводити до запалення дихальних шляхів, супроводжуючись кашлем, виділенням слизу, астмою та хронічним бронхітом, і робити людей більш схильними до респіраторних інфекцій. Рівень SO2 має прямий вплив на здоров'я, оскільки збільшується кількість пацієнтів, які потрапляють в лікарні через захворювання, і зростає ризик проблем з серцем та смертей у випадках високого рівня цього газу [6-7].

Свинець (Pb), потрапляючи в атмосферу, може мати серйозні наслідки для здоров'я людей та навколишнього середовища. Свинець може бути виділений у повітря в результаті різноманітних промислових процесів, таких як виробництво акумуляторів, автомобільна та промислова діяльність. При вдиханні свинцевих частинок люди стикаються з ризиком різноманітних захворювань. Серед наслідків можуть бути порушення функції нервової системи, зокрема, вплив на розвиток дітей, ведучи до виникнення проблем із навчанням та затримок у розвитку. Також

свинець може викликати проблеми із серцево-судинною системою та погіршення якості кровообігу. Забруднення повітря свинцем особливо небезпечне для вагітних жінок та маленьких дітей, оскільки воно може впливати на нормальний розвиток нервової системи у плоду. Тому важливо вживати заходи для зменшення викидів свинцю та вживати заходів безпеки для мінімізації впливу цього токсичного металу на здоров'я людей та навколишнє середовище.

Бензапірен ($C_{20}H_{12}$) – це один із токсичних ароматичних вуглеводнів, який може потрапляти в повітря та викликати серйозні проблеми для здоров'я та довкілля. Виникнення бензапірену пов'язане із згорянням вуглеводневих сполук, особливо при неповному згорянні палива, такого як тверді вугілля, деревина та інші органічні матеріали. При вдиханні бензапірен може викликати серйозні наслідки для здоров'я. Ця сполука може бути канцерогенною та сприяти розвитку ракових захворювань, зокрема легень. Крім того, бензапірен може впливати на роботу респіраторної системи, викликаючи алергічні реакції, астматичні стани та загальне погіршення функції легень. Зменшення викидів бензапірену в повітря важливо для забезпечення безпеки здоров'я людей та збереження екологічної рівноваги. Впровадження технологій, що зменшують викиди при спалюванні палива, та стратегії управління викидами можуть сприяти зменшенню наслідків бензапіренового забруднення повітря [6-7].

У Вінниці моніторинг якості повітря здійснюється не лише державними організаціями , а і різними громадськими організаціями, які створюють свої проекти з покращення благоустрою міста. Серед таких проектів:

- Eco City;
- SaveEcoBot;

Мережа Eco City об’єднує понад 500 станцій моніторингу, що працюють в єдиній мережі та надають результати спостережень якості атмосферного повітря та хімічної та радіаційної безпеки. З 2019 року команда Eco City накопичувала та обробляла дані, що дало їм змогу створити масивну базу даних результатів

спостережень. Щоб мати змогу використовувати зібрані дані, необхідно отримати доступ до кабінету дослідника Eco City. Дослідник може встановити власну станцію і отримувати дані з неї чи отримати доступ до даних з тих станцій, які необхідні для проведення досліджень, проте для цього потрібно обговорити умови використання даних з командою Eco City. В залежності від типу станції користувач сервісу може отримувати дані за наступними показниками:

- PM 1.0;
- PM 2.5;
- PM 10;
- Температура;
- Вологість;
- Атмосферний тиск;
- CO2;
- ЛОС (Формальдегід);
- O3;
- Монооксид вуглецю;
- тощо.

Сервіс Eco City має інтерактивну карту зі шкалою, яка дозволяє побачити як змінювався рівень забрудненості повітря за певний час (рис 1.3, 1.4). Завдяки цьому можна відслідковувати динаміку змін якості повітря у часі [8].

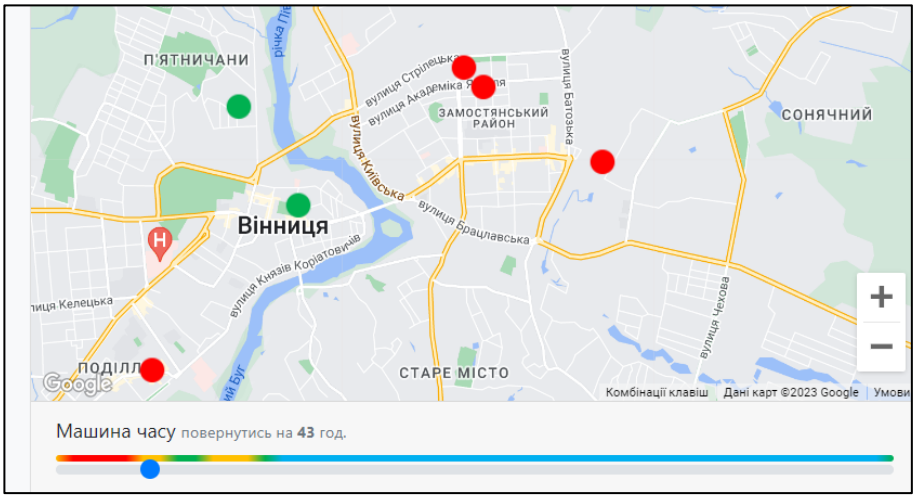


Рисунок 1.3 – Приклад інтерфейсу Eco City

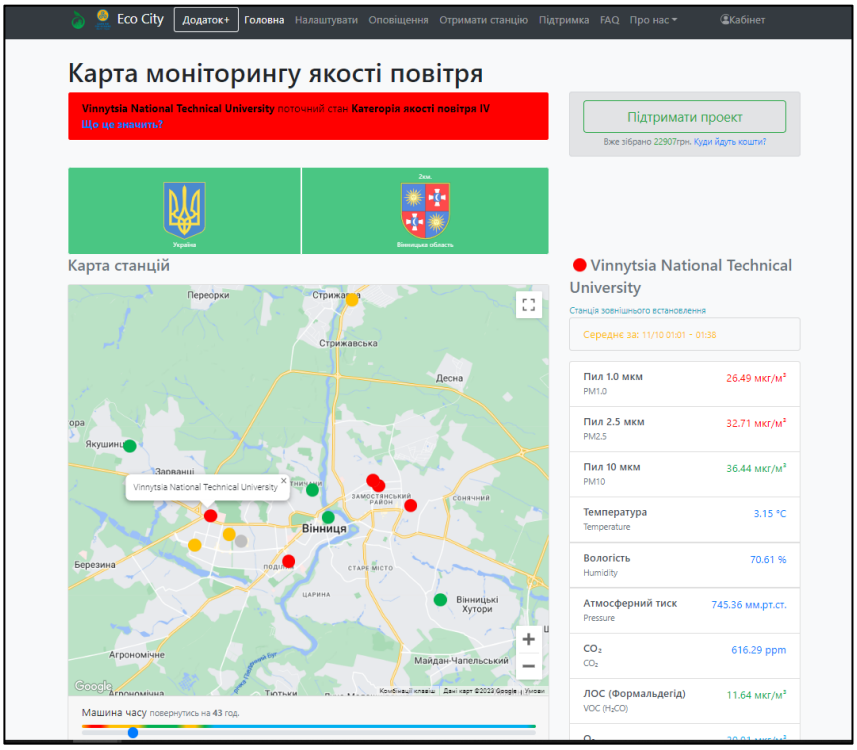
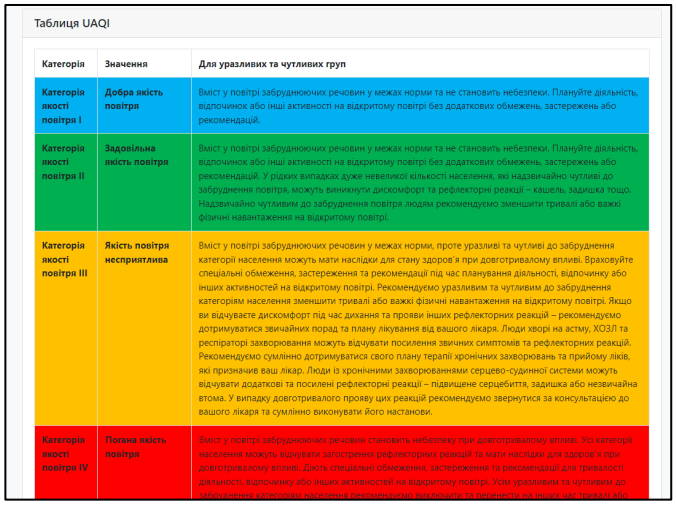


Рисунок 1.4 – Приклад інтерфейсу Eco City

Також сервіс Eco City надає змогу дізнатися індекс якості повітря (AQI). Індекс якості повітря це шкала, що демонструє рівень забруднення повітря. Розрахунок індексу якості повітря вимагає середніх значень концентрації забруднювальних речовин за певний період, які отримані під час моніторингу

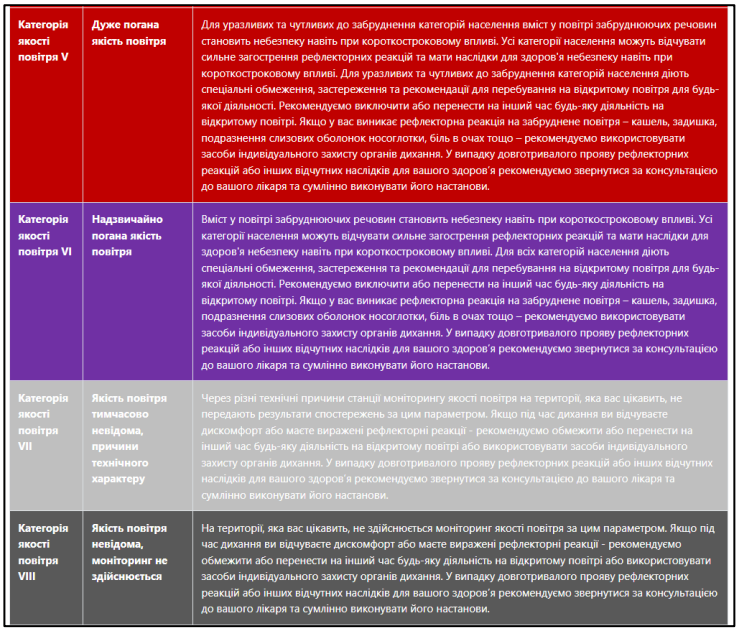
атмосферного повітря. Індекс якості повітря зазвичай об’єднують в діапазони. Кожному діапазону надається певний ідентифікатор, код, колір та рекомендації щодо охорони здоров’я (рис. 1.5, 1.6).



Таблиця UAQI

Категорія	Значення	Для уразливих та чутливих груп
Категорія якості повітря I	Добра якість повітря	Вміст у повітрі забруднюючих речовин у межах норми та не становить небезпеки. Плануйте діяльність, відпочинок або інші активності на відкритому повітрі без додаткових обмежень, застережень або рекомендацій.
Категорія якості повітря II	Задовільна якість повітря	Вміст у повітрі забруднюючих речовин у межах норми та не становить небезпеки. Плануйте діяльність, відпочинок або інші активності на відкритому повітрі без додаткових обмежень, застережень або рекомендацій. У рідких випадках дуже невеликої кількості населення, які надзвичайно чутливі до забруднення повітря, можуть виникнути дискомфорт та рефлекторні реакції – кашель, задишка тощо. Надзвичайно чутливим до забруднення повітря людям рекомендуємо зменшити тривалі або важкі фізичні навантаження на відкритому повітрі.
Категорія якості повітря III	Якість повітря несприятлива	Вміст у повітрі забруднюючих речовин у межах норми, проте уразливі та чутливі до забруднення категорії населення можуть мати наслідки для стану здоров’я при довготривалому впливі. Враховуйте спеціальні обмеження, застереження та рекомендації під час планування діяльності, відпочинку або інших активностей на відкритому повітрі. Рекомендуємо уразливим та чутливим до забруднення категоріям населення зменшити тривалі або важкі фізичні навантаження на відкритому повітрі. Якщо ви відчуваєте дискомфорт під час дихання та прояви інших рефлекторних реакцій – рекомендуємо дотримуватися звичайних порад та плану лікування від вашого лікаря. Люди хворі на астму, ХОЗЛ та респіраторі захворювання можуть відчувати посилення звичних симптомів та рефлекторних реакцій. Рекомендуємо сумлінно дотримуватися свого плану терапії хронічних захворювань та прийому ліків, які призначив ваш лікар. Люди із хронічними захворюваннями серцево-судинної системи можуть відчувати додаткові та посилені рефлекторні реакції – підвищене серцебиття, задишка або незвичайна втома. У випадку довготривалого прояву цих реакцій рекомендуємо звернутися за консультацією до вашого лікаря та сумлінно виконувати його настанови.
Категорія якості повітря IV	Погана якість повітря	Вміст у повітрі забруднюючих речовин становить небезпеку при довготривалому впливі. Усі категорії населення можуть відчувати загострення рефлекторних реакцій та мати наслідки для здоров’я при довготривалому впливі. Діють спеціальні обмеження, застереження та рекомендації для тривалості діяльності, відпочинку або інших активностей на відкритому повітрі. Усім уразливим та чутливим до забруднення категоріям населення рекомендуємо відкласти на пізніше застереження або

Рисунок 1.5 – Приклад шкали індексу якості повітря із сервісу Eco City



Категорія якості повітря V	Дуже погана якість повітря	Для уразливих та чутливих до забруднення категорій населення вміст у повітрі забруднюючих речовин становить небезпеку навіть при короткостроковому впливі. Усі категорії населення можуть відчувати сильне загострення рефлекторних реакцій та мати наслідки для здоров’я небезпеку навіть при короткостроковому впливі. Для уразливих та чутливих до забруднення категорій населення діють спеціальні обмеження, застереження та рекомендації для перебування на відкритому повітря для будь-якої діяльності. Рекомендуємо виключити або перенести на інший час будь-яку діяльність на відкритому повітрі. Якщо у вас виникає рефлекторна реакція на забруднене повітря – кашель, задишка, подразнення слизових оболонок носоглотки, біль в очах тощо – рекомендуємо використовувати засоби індивідуального захисту органів дихання. У випадку довготривалого прояву рефлекторних реакцій або інших відчутних наслідків для вашого здоров’я рекомендуємо звернутися за консультацією до вашого лікаря та сумлінно виконувати його настанови.
Категорія якості повітря VI	Надзвичайно погана якість повітря	Вміст у повітрі забруднюючих речовин становить небезпеку навіть при короткостроковому впливі. Усі категорії населення можуть відчувати сильне загострення рефлекторних реакцій та мати наслідки для здоров’я небезпеку навіть при короткостроковому впливі. Для всіх категорій населення діють спеціальні обмеження, застереження та рекомендації для перебування на відкритому повітря для будь-якої діяльності. Рекомендуємо виключити або перенести на інший час будь-яку діяльність на відкритому повітрі. Якщо у вас виникає рефлекторна реакція на забруднене повітря – кашель, задишка, подразнення слизових оболонок носоглотки, біль в очах тощо – рекомендуємо використовувати засоби індивідуального захисту органів дихання. У випадку довготривалого прояву рефлекторних реакцій або інших відчутних наслідків для вашого здоров’я рекомендуємо звернутися за консультацією до вашого лікаря та сумлінно виконувати його настанови.
Категорія якості повітря VII	Якість повітря тимчасово невідома, причини технічного характеру	Через різні технічні причини станції моніторингу якості повітря на території, яка вас цікавить, не передають результати спостережень за цим параметром. Якщо під час дихання ви відчуваєте дискомфорт або маєте виражені рефлекторні реакції - рекомендуємо обмежити або перенести на інший час будь-яку діяльність на відкритому повітрі або використовувати засоби індивідуального захисту органів дихання. У випадку довготривалого прояву рефлекторних реакцій або інших відчутних наслідків для вашого здоров’я рекомендуємо звернутися за консультацією до вашого лікаря та сумлінно виконувати його настанови.
Категорія якості повітря VIII	Якість повітря невідома, моніторинг не здійснюється	На території, яка вас цікавить, не здійснюється моніторинг якості повітря за цим параметром. Якщо під час дихання ви відчуваєте дискомфорт або маєте виражені рефлекторні реакції - рекомендуємо обмежити або перенести на інший час будь-яку діяльність на відкритому повітрі або використовувати засоби індивідуального захисту органів дихання. У випадку довготривалого прояву рефлекторних реакцій або інших відчутних наслідків для вашого здоров’я рекомендуємо звернутися за консультацією до вашого лікаря та сумлінно виконувати його настанови.

Рисунок 1.6 – Продовження шкали індексу якості повітря з сервісу Eco City

1.3 Огляд проблематики прогнозування якості повітря

Система моніторингу якості повітря закладає основу для прогнозування та відстеження якості повітря, а більш точне прогнозування якості повітря стає можливим завдяки покращенню даних моніторингу. Прогнозування якості повітря є ефективним підходом до захисту здоров'я людини шляхом надання попередніх попереджень про шкідливе забруднення повітря.

Було запропоновано численні моделі прогнозування, в основному для концентрації забруднюючих речовин. Існує багато моделей прогнозування, які розробляються на основі аналізу часових рядів для прогнозування концентрації забруднювачів повітря. Підходи до моделювання можна розділити на три категорії: хімічні моделі, статистичні моделі та моделі штучного інтелекту.

Хімічні моделі прогнозують забруднювачі повітря на основі трансформації та хімічних властивостей забруднювачів. Найпоширенішими моделями для прогнозування якості повітря є Community Multiscale Air Quality (CMAQ), Комплексна модель якості повітря з розширеннями (CAMx), Система спостереження за Землею Годдарда (GEOS-Chem) та прогнозування погодних досліджень (WRF). Дані моделі можуть використовувати хімічні реакції для передбачення рівнів забруднювачів повітря, їх ефективність залежить від різноманітних даних щодо цих забруднювачів, а величезна обсягова інформація для точного прогнозування робить цей процес складним. Крім того, моделі базуються на обширних розрахунках, що може призвести до обмеження їх продуктивності [9-15].

Статистичні моделі, такі як ARIMA (Autoregressive Integrated Moving Average,), та регресійні моделі, розвивають статистичний зв'язок між історичними даними різних параметрів впливу на забруднювачі повітря. Однак статистичні моделі демонструють обмеження у вивченні великих багатовимірних і складних нелінійних даних часових рядів. Моделі також не в змозі передбачити

багатоступінчасті часові горизонти забруднювача повітря на основі численних змінних, що впливають.

Враховуючи обмеження хімічних транспортних і статистичних моделей у навчанні та прогнозуванні забруднювачів повітря на основі різних параметрів впливу, були створені технології на основі штучного інтелекту (ШІ), такі як машинне навчання та моделі глибокого навчання. Однак ці методи мають недоліки, які полягають у тому, що вони обмежені у вирішенні більших нелінійних наборів даних часових рядів і нездатні ефективно фіксувати особливості розподілу наборів даних про якість повітря.

Розуміння важливості точного прогнозування концентрації забруднювачів повітря призвело до все більшого розвитку досліджень і перспективних моделей прогнозування. В останні кілька років глибоке навчання стало популярною технікою в застосуванні прогнозування якості повітря і демонструє кращу продуктивність у порівнянні з традиційними нейронними мережами та іншими моделями машинного навчання. Методи глибокого навчання, такі як RNN (Recurrent neural network), LSTM (Long short-term memory), CNN (Convolutional neural network) і GRU (Gated recurrent unit), розробляються на основі архітектури нейронної мережі, що складається з безлічі обчислювальних шарів. Дані методи здатні мінімізувати недоліки традиційних нейронних мереж у задачах часових рядів якості повітря та забезпечити чудові показники прогнозування [9-15].

1.4 Висновки

У даному розділі проведено огляд проблематики забруднення якості атмосферного повітря у світі та в Україні. Визначено, які основні показники вимірюються на даний час в Україні та розглянуто чим вони є небезпечними для здоров’я людини. Розглянуто основні недоліки моніторингу атмосферного повітря в Україні та як їх можна виправити. Розглянуто стан моніторингу атмосферного

повітря у місті Вінниця, які показники вимірюються тощо. Досліджено проблематику прогнозування якості атмосферного повітря з використанням штучного інтелекту та машинного навчання. Визначено, що для прогнозування стану атмосферного повітря міста Вінниця, моделі на основі рекурентних нейронних мереж, наприклад LSTM, є ефективнішими за інші.

2 ВИБІР ОПТИМАЛЬНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Вибір мови програмування та середовища розробки

На сьогоднішній день існує багато способів розв'язання проблеми прогнозування якості атмосферного повітря. У ході виконання даної роботи було використано мову програмування Python. Python – це високорівнева мова програмування, яка була створена 1991 році. Вона має чистий і читабельний синтаксис, що дозволяє програмістам писати зрозумілий і підтримуваний код.

Одним з головних переваг Python є його широкий спектр застосувань. Він використовується для веб-розробки, наукових обчислень, аналізу даних, штучного інтелекту, автоматизації завдань та багатьох інших галузях. Ця універсальність робить його однією з найпопулярніших мов програмування у світі.

Для Python написано безліч бібліотек та модулів, що значно спрощую роботу не лише програмістів а і науковців та дослідників які користуються даною мовою програмування у свої цілях. Під час виконання даної роботи було використано наступні бібліотеки:

Numpy. Бібліотека NumPy є важливим інструментом у світі наукового програмування та обробки числових даних на мові програмування Python. Її основною метою є надання потужних структур даних, високоефективних функцій та інструментів для роботи з масивами та матрицями [15].

Pandas. Бібліотека Pandas є ключовим інструментом для обробки та аналізу даних на мові програмування Python. Створена для ефективної та зручної роботи з табличними даними, Pandas надає високорівневі структури даних, які роблять легким та зрозумілим маніпулювання інформацією. Pandas використовується в різноманітних областях, включаючи фінанси, науку про дані, економіку та інші галузі, де обробка та аналіз даних є важливою частиною роботи. Взаємодія з іншими

бібліотеками, такими як NumPy та Matplotlib, дозволяє створювати повноцінні програми для візуалізації та розуміння великих обсягів інформації [16].

Datetime. Бібліотека DateTime використовується для роботи з датами та часом в мові програмування Python. Вона надає зручний та потужний інтерфейс для роботи з різноманітними аспектами часу, включаючи обробку дат, розрахунки різниць у часі, форматування та парсинг дат. DateTime також включає можливості для роботи зі стандартами часових поясів, обчислення різниці у часі між двома датами та перетворення дат в різні формати для зручності виведення та зчитування

Matplotlib. Matplotlib – це потужна бібліотека для створення високоякісних графіків та візуалізації даних в мові програмування Python. Вона надає широкі можливості для створення різноманітних типів графіків, включаючи лінійні графіки, стовпчасті діаграми, кругові діаграми, scatter plots, 3D графіки та багато інших. Matplotlib є надзвичайно гнучкою та конфігурується. Вона дозволяє контролювати практично кожен елемент графіку, включаючи осі, легенду, кольори та розміри. Це робить бібліотеку ідеальним інструментом для створення як простих графіків для швидкого аналізу даних, так і складних візуалізацій для наукових досліджень та презентацій.

Prophet. Prophet – це відкрита бібліотека для прогнозування часових рядів, розроблена командою Facebook. Вона спеціально створена для простоти використання та ефективної роботи з рядами, які мають декілька років інтервалів спостережень та можуть містити пропущені дані та вихідні години. Prophet використовує адитивну модель, яка розбиває часовий ряд на складові, такі як тренд, сезонність та свята, і дозволяє здійснювати прогнози з високою точністю. Крім того, бібліотека надає засоби для візуалізації прогнозів та діагностики моделі. Prophet широко використовується в галузі фінансів, маркетингу, транспорту та інших областях для ефективного прогнозування часових рядів з мінімальними зусиллями з боку користувача [17].

Scikit-learn. Scikit-learn, часто відома як sklearn, – це відкрита бібліотека машинного навчання для мови програмування Python. Розроблена на базі бібліотек NumPy, SciPy та Matplotlib, scikit-learn надає простий та ефективний інтерфейс для виконання багатьох завдань машинного навчання, таких як класифікація, регресія, кластеризація, вимірювання якості моделей та вибір параметрів. Scikit-learn володіє вражаючою кількістю вбудованих алгоритмів для різних задач, включаючи метод опорних векторів (SVM), випадкові ліси, градієнтний бустінг, k-середні, наївний байесівський класифікатор та багато інших. Це робить бібліотеку ідеальним інструментом для навчання та використання в реальних проектах машинного навчання. Scikit-learn також активно використовується в наукових дослідженнях, конкурсах з машинного навчання та промислових застосуваннях, що свідчить про його широкий вплив на галузь машинного навчання [18].

Keras. Keras – це високорівневий інтерфейс для машинного навчання, який працює поверх бібліотек Tensorflow та Theano. Розроблений з фокусом на простоті використання та швидкості розробки моделей нейронних мереж, Keras став однією з найпопулярніших бібліотек для глибокого навчання в середовищі Python. Keras підтримує низку різноманітних типів моделей, включаючи звичайні нейронні мережі (DNN), згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та їх комбінації. Бібліотека також дозволяє використовувати передові технології глибокого навчання, такі як трансформери та генеративні моделі. Keras використовується як для досліджень в області штучного інтелекту, так і для створення практичних застосувань, таких як класифікація зображень, розпізнавання мови та обробка природної мови. Завдяки своїй гнучкості та ефективності, Keras залишається потужним інструментом для глибокого навчання в широкому спектрі задач [19].

Statsmodels. Statsmodels – це бібліотека для статистичного моделювання та тестування гіпотез в мові програмування Python. Ця бібліотека дозволяє виконувати різноманітні статистичні аналізи, включаючи лінійну та нелінійну регресію, аналіз

часових рядів, аналіз дисперсії (ANOVA) та багато інших. Однією з ключових переваг Statsmodels є надання широкого спектру статистичних моделей та інструментів для їхнього аналізу. Бібліотека включає в себе класичні методи статистики, такі як т-тести та аналіз залежності, що робить її важливим інструментом для дослідження та перевірки гіпотез у наукових дослідженнях та прикладних задачах. Statsmodels широко використовується в області соціальних та економічних наук, фінансів, медицини та інших галузях, де потрібно проводити статистичний аналіз [20].

У ході виконання даної роботи використовувався сервіс Kaggle. Kaggle це платформа для змагань з аналітики. У рамках змагань користувачі платформи створюють власні моделі для вирішення певної проблеми чи задачі, запропонованої компаніями або іншими користувачами. Спільнота Kaggle надає доступ до різноманітних наборів даних, завдань та конкурсів, де учасники можуть застосовувати свої навички та конкурувати за призи. Великим перевагою є можливість навчання на реальних завданнях з різних галузей, від медицини та фінансів до реклами та технологій.

Крім конкурсів, Kaggle також надає інші ресурси, такі як навчальні матеріали, форуми для обговорення та спілкування, а також можливості для співпраці та обміну знаннями серед учасників спільноти.

У даному сервісі можна завантажити власний датасет, на основі якого буде проводитись дослідження. Дана платформа була вибрана, оскільки вона є зручною для виконання задачі прогнозування якості атмосферного повітря. Однією з переваг Kaggle є те, що дана платформа виконує обчислення не на боці користувача, а проводить обчислення в себе на серверах. Тобто навіть, якщо користувач не володіє потужним персональним комп’ютером, він зможе проводити дослідження на рівних з іншими учасниками. Приклад інтерфейсу наведено на рисунку 2.1

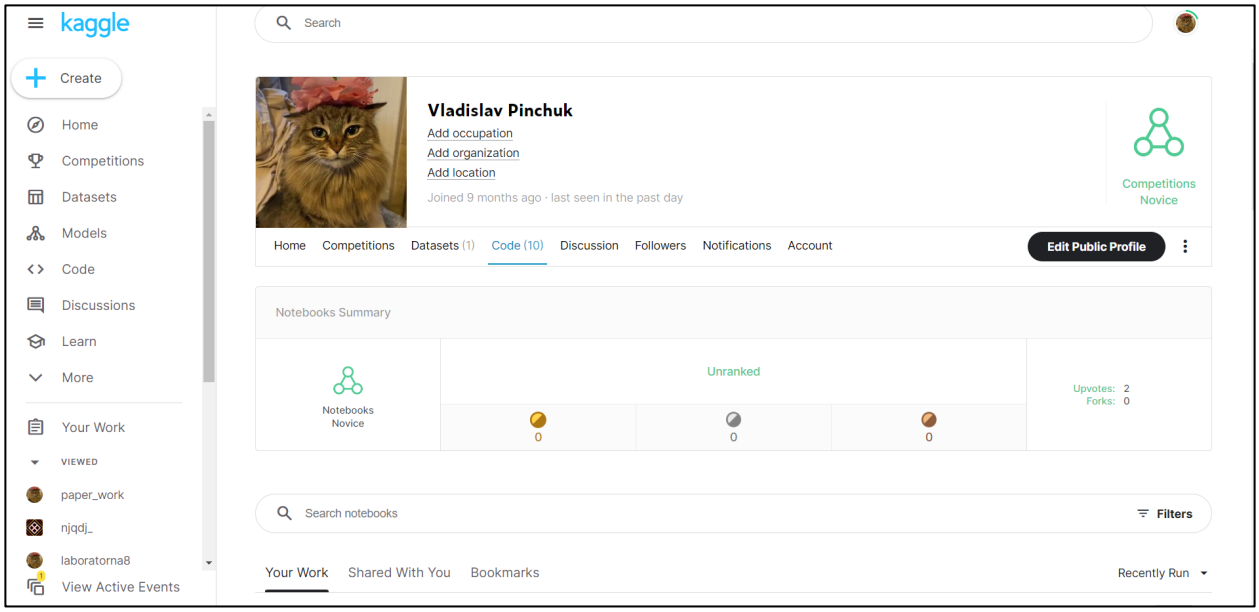


Рисунок 2.1 – Інтерфейс платформи Kaggle

2.2 Огляд моделей, що використовувались в дослідженні

Машинне навчання (Machine Learning, ML) - великий підрозділ штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися. Основна ідея полягає в тому, щоб система могла визначати закономірності та робити прогнози на основі вхідних даних.

До основних типів машинного навчання відносять:

– Навчання з вчителем (Supervised Learning). Модель навчається на підставі пари "вхід-вихід" (input-output), де вихід вже відомий. Задачі включають класифікацію (визначення категорії) та регресію (прогнозування числового значення);

– Навчання без вчителя (Unsupervised Learning). Модель навчається без конкретних пар "вхід-вихід". Головна задача - знаходження внутрішньої структури в даних, така як кластеризація (групування схожих об'єктів) та зменшення розмірності;

- Підготовка з нагородою (Reinforcement Learning). Модель навчається, взаємодіючи з оточенням і отримуючи від нього нагороду або покарання. Мета - вивчити оптимальні стратегії для максимізації нагород.

Основні компоненти процесу машинного навчання включають:

- Дані. Достатньо великий обсяг якісних та кількісних даних є основною основою для машинного навчання;
- Вибірка та підготовка даних. Дані часто потребують обробки та очищення перед тим, як їх можна буде використовувати для навчання моделей;
- Вибір моделі. Це визначається завданням та характером даних. Вибір відповідного алгоритму часто залежить від типу задачі, такої як класифікація, регресія чи кластеризація;
- Тренування моделі. Модель навчається за допомогою тренувальних даних, де вона виявляє закономірності та залежності;
- Оцінка та налаштування: Модель оцінюється за допомогою тестового набору, і якщо потрібно, проводяться корекції параметрів для поліпшення її продуктивності та уникнення перенавчання;
- Прогнозування або вирішення завдань: Після тренування та оцінки модель може використовуватися для роботи з новими, раніше невідомими даними, роблячи прогнози, класифікації або здійснюючи інші види прийняття рішень;

На рисунку 2.2 зображена «карта машинного навчання».

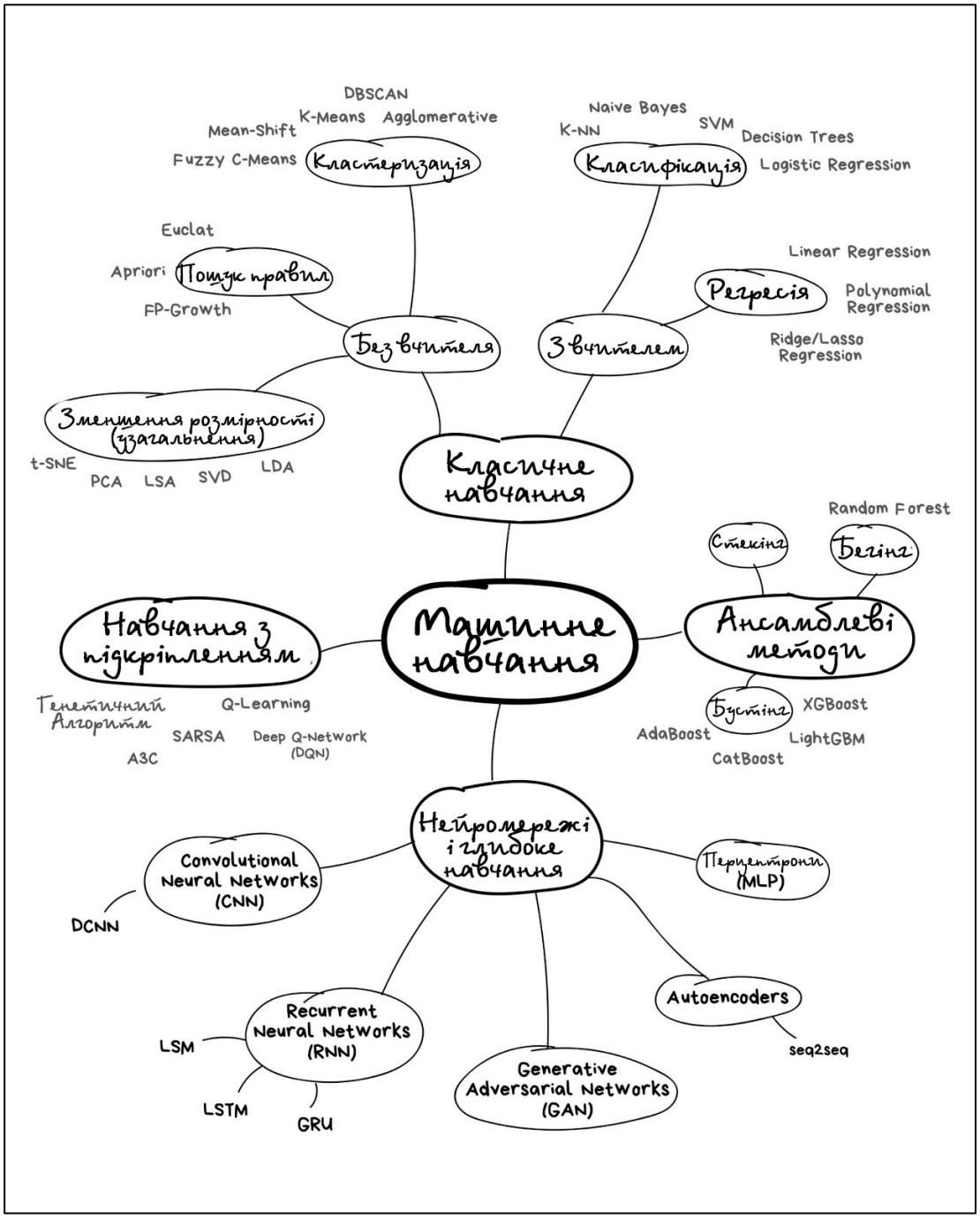


Рисунок 2.2 – Карта машинного навчання

Машинне навчання широко застосовується в різних галузях, включаючи медицину, фінанси, технології, автомобільну промисловість, науку про дані та інші, привносячи величезні можливості для автоматизації та оптимізації процесів.

У ході виконання магістерської кваліфікаційної роботи використовувались наступні моделі: Prophet, LSTM та ARIMA.

2.2.1 Модель Prophet

Модель Prophet – це інструмент для прогнозування часових рядів, розроблений компанією Facebook. Вона була спеціально створена з метою надання для широкої аудиторії простого та ефективного інструменту для прогнозування, який може враховувати різноманітні фактори, такі як сезонність, свята та змінні тренди.

Однією з ключових особливостей моделі Prophet є її здатність автоматично враховувати різні компоненти часового ряду. Модель може адаптуватися до щорічних та щотижневих сезонностей, враховувати вплив свят, а також автоматично виявляти та моделювати змінні тренди та важливі події в даних.

Prophet використовує аддитивну модель, що дозволяє розбити часовий ряд на компоненти, такі як тренд, сезонність та свята, і об'єднати їх для прогнозування майбутніх значень. Це робить модель ефективною для роботи з даними, де є явно виражені сезонні та трендові закономірності. Модель Prophet дозволяє легко налаштовувати параметри та проводити аналіз якості прогнозування. Крім того, вона надає інструменти для візуалізації результатів, що допомагає зрозуміти, як модель адаптується до вхідних даних та як точно вона передбачає майбутні значення. Приклад створення моделі Prophet наведено на рисунку 2.3.

Prophet широко використовується в різних галузях, включаючи економіку, фінанси, маркетинг, а також в будь-якому сценарії, де важливо точно передбачати часові ряди. Його простий інтерфейс та можливості зрозумілого моделювання роблять його доступним для великої кількості користувачів, від новачків до досвідчених аналітиків даних [21] .

```
pro_change= Prophet(n_changepoints=20, yearly_seasonality=True, changepoint_prior_scale=0.001)
forecast = pro_change.fit(df).predict(future_dates)
fig= pro_change.plot(forecast);
a = add_changepoints_to_plot(fig.gca(), pro_change, forecast)
```

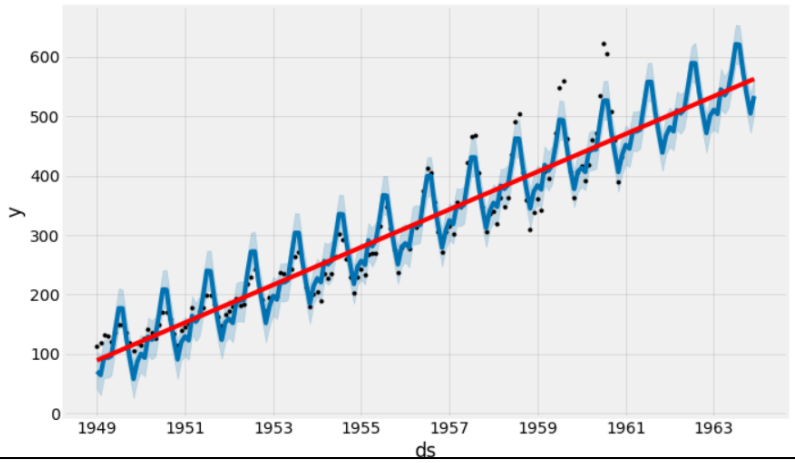


Рисунок 2.3 – Приклад створення Prophet

Загальний графік прогнозу можна розбити на компоненти. Це дозволяє ознайомитися з трендом та сезонністю. До прикладу, можна подивитися як поводять себе дані протягом дня, місяця, години тощо. Приклад розбиття на компоненти зображено на рисунку 2.4.

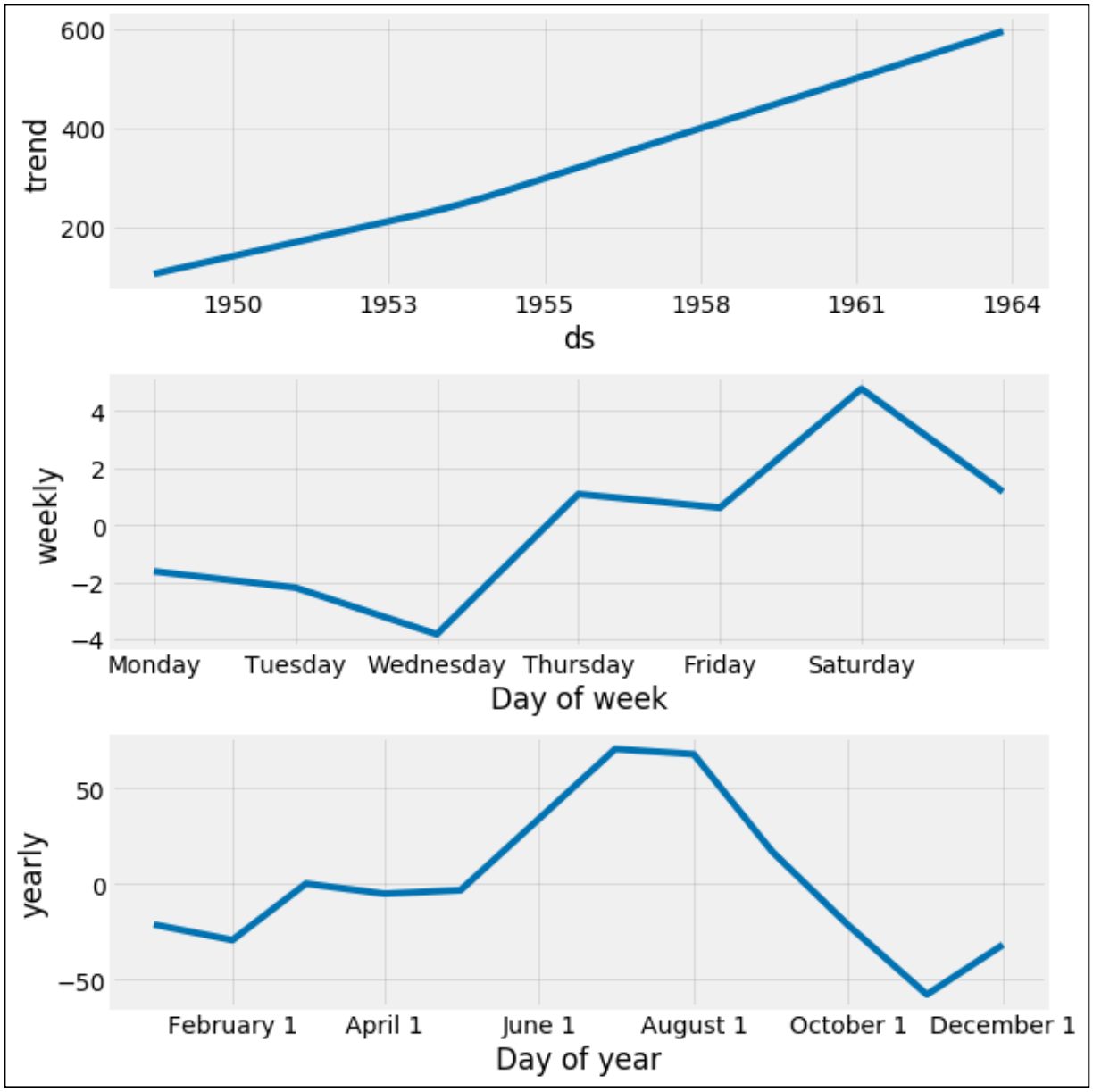


Рисунок 2.4 – Приклад компонентів пронозу

2.2.2 Модель LSTM

Long Short-Term Memory (LSTM) — це тип рекурентної нейронної мережі (RNN), яка спеціально розроблена для обробки послідовних даних, таких як часові ряди, мова та текст. Мережі LSTM здатні вивчати довгострокові залежності в послідовних даних, що робить їх добре придатними для таких завдань, як переклад

мови, розпізнавання мовлення та прогнозування часових рядів. LSTM є різновидом рекурентних нейронних мереж (RNN).

Рекурентні нейронні мережі (RNN) — це клас нейронних мереж, призначений для роботи з послідовністю даних. Основна особливість RNN полягає в їхній здатності враховувати попередній контекст та використовувати інформацію з попередніх кроків для обробки поточного вхідного сигналу. Кожен часовий крок RNN обробляє новий вхід та оновлює свій стан. Основне обмеження RNN — проблема зниклого градієнта, коли важко тренувати на довгострокових залежностях. Це призводить до втрати контексту для довгих послідовностей [22]. Схема RNN зображена на рисунку 2.5.

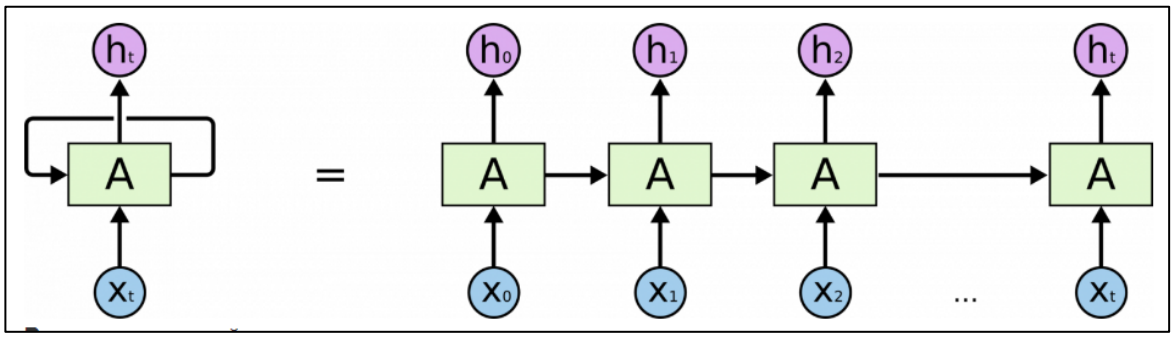


Рисунок 2.5 – Схема рекурентної нейронної мережі

Традиційний RNN має один прихований стан, який передається через час, що може ускладнити для мережі вивчення довгострокових залежностей. LSTM вирішують цю проблему, вводячи комірку пам’яті, яка є контейнером, який може зберігати інформацію протягом тривалого періоду часу. Комірка пам'яті керується трьома воротами: вхідним, забуття і вихідним. Ці ворота вирішують, яку інформацію до комірки пам’яті додати, видалити з неї та вивести з неї.

Ключовий елемент LSTM – це стан комірки, горизонтальна лінія, що проходить у верхній частині схеми (рис.2.6).

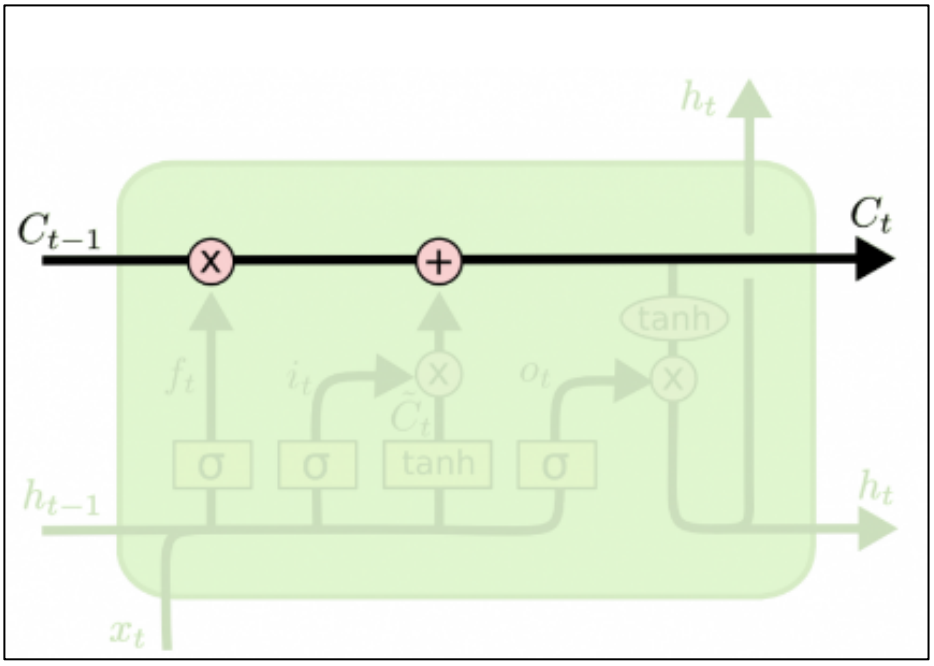


Рисунок 2.6 – Стан комірки LSTM

Вхідні ворота приймають рішення про те, які значення є важливими та повинні бути пропущені через модель. У вхідному шлюзі використовується сигмоїдна функція, яка визначає, які значення передавати через рекурентну мережу. Нуль відкидає значення, тоді як 1 зберігає його. Тут також використовується функція TanH, яка визначає, наскільки важливі для моделі вхідні значення в діапазоні від -1 до 1 (рис.2.7)

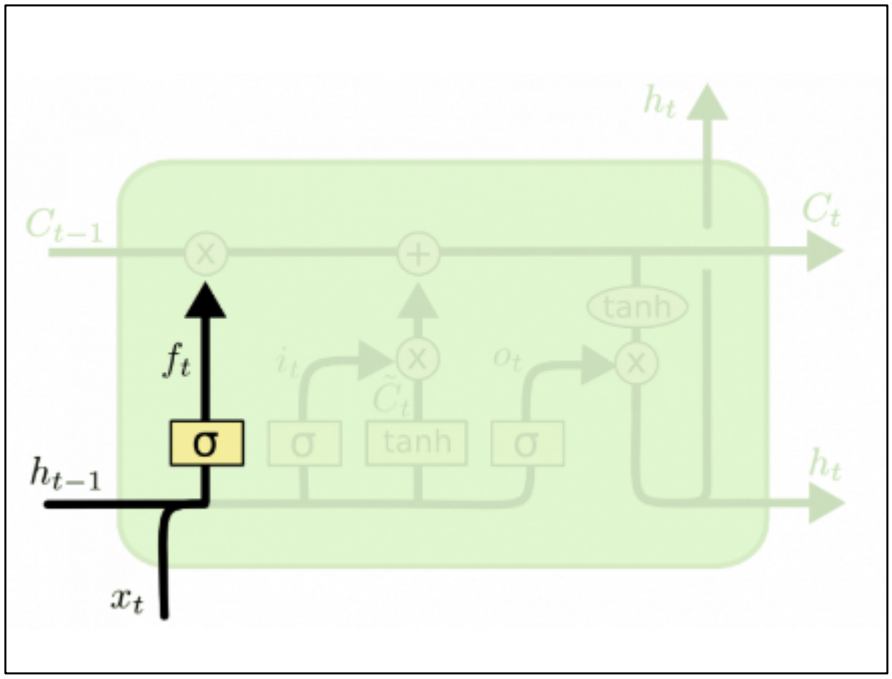


Рисунок 2.7 – Стан вхідних воріт

Після врахування поточних вхідних даних і стану пам’яті вихідний вентиль вирішує, які значення перенести на наступний часовий крок. У вихідному схемі значення аналізуються та призначаються важливість у діапазоні від -1 до 1. Це регулює дані перед тим, як вони передаються до наступного розрахунку за часом (рис.2.8).

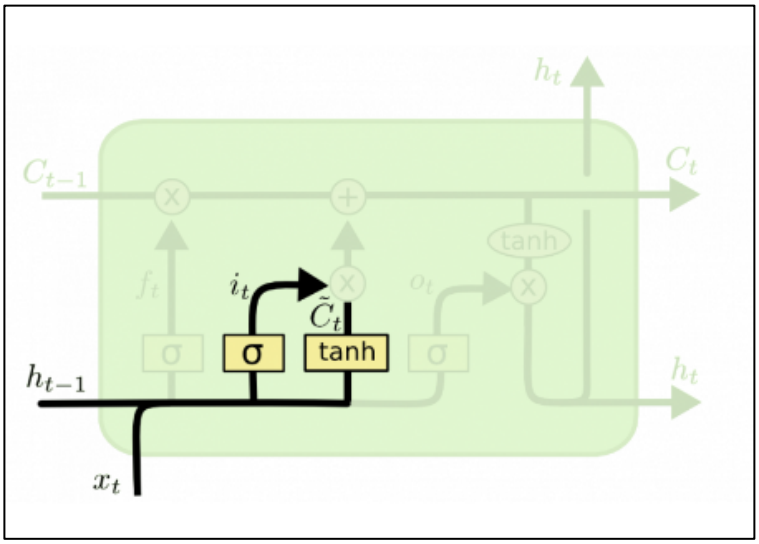


Рисунок 2.8 – Стан LSTM під час врахування даних

Нарешті, завдання шлюзу забуття полягає у видаленні інформації, яку модель вважає непотрібною для прийняття рішення щодо природи вхідних значень. Ворота забуття використовує сигмоїдну функцію для значень, виводячи числа від 0 (забути) до 1 (зберегти) (рис.2.8). Схема LSTM моделі зображена на рисунку 2.9 [22]. Приклад створення LSTM моделі зображено на рисунку 2.11.

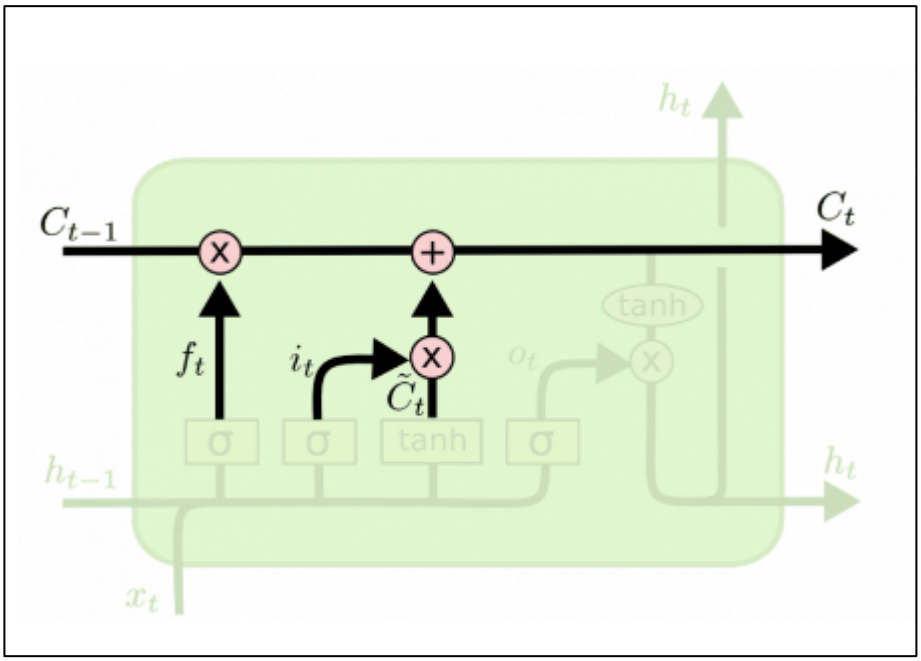


Рисунок 2.9 – Стан воріт забуття моделі LSTM

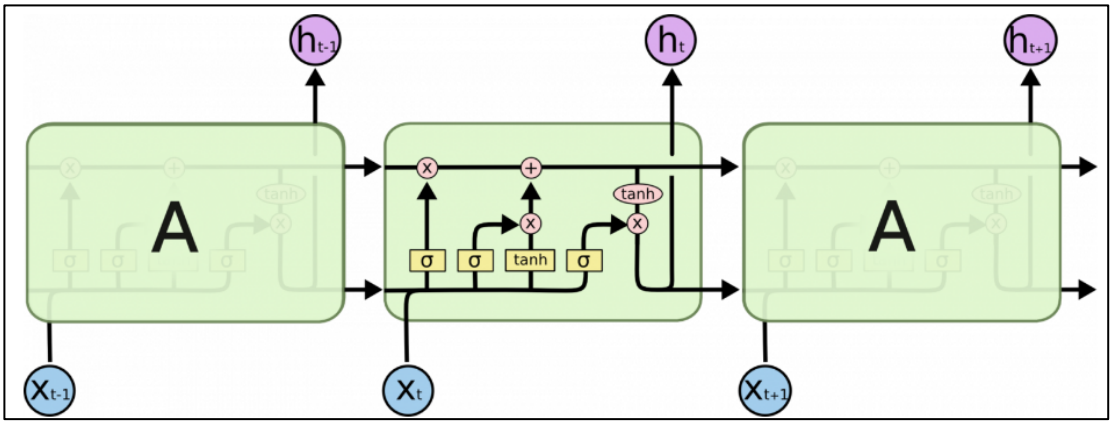


Рисунок 2.10 – Схема моделі LSTM

```python
# define model
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(n_steps, n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
# fit model
model.fit(X, y, epochs=200, verbose=0)
# demonstrate prediction
x_input = array([70, 80, 90])
x_input = x_input.reshape((1, n_steps, n_features))
yhat = model.predict(x_input, verbose=0)
print(yhat)
```

Рисунок 2.11 – Приклад створення LSTM моделі на Python

2.2.3 Модель ARIMA

ARIMA (Autoregressive Integrated Moving Average) — це статистична модель для аналізу та прогнозу часових рядів. Вона поєднує в собі три основні компоненти: авторегресію (AR), інтегрування (I), та ковзне середнє (MA). Модель ARIMA розроблена для роботи з послідовностями, де можна виявити тренди та сезонності. Авторегресія (AR) компонент враховує попередні значення часового ряду. Вказує, що поточне значення може залежати від попередніх значень з певним затримкою (lag).

Інтегрування (I) частина вказує на необхідність порядку диференціювання для зроблення часового ряду стаціонарним. Стаціонарний ряд – це ряд, в якому характеристики, такі як середнє значення та дисперсія, не залежать від часу.

Ковзне середнє (MA) компонент враховує попередні шоки або помилки прогнозу. Вказує, що поточне значення часового ряду залежить від попередніх помилок з певним затримкою (lag).

Формула моделі ARIMA представлена нижче

$$X_t = c + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \quad (2.1)$$

де Xt – значення часового ряду в час t;

c – константа;

ϕi – коефіцієнти авторегресії;

θi – коефіцієнти ковзного середнього;

εt – помилка прогнозу в час t;

p – порядок авторегресії (AR order);

d – ступінь диференціації;

q – порядок ковзного середнього (MA order).

Після тренування моделі важливо провести діагностику для перевірки, наскільки добре вона адаптується до даних. Для цього використовуються графіки залишків (помилок прогнозу), щоб впевнитися, що вони мають властивості білого шуму (незалежні та однорідні). Приклад створення моделі ARIMA зобажено на рисунку 2.12.

ARIMA вважається потужним інструментом для багатьох аналітичних завдань у галузі економіки, фінансів та інших галузей. Однак для деяких складних часових рядів може знадобитися використання більш складних моделей, таких як SARIMA (сезонна ARIMA), які розширюють базовий ARIMA для врахування сезонності [23].

```
]:
from statsmodels.tsa.arima_model import ARIMA

# 1,1,2 ARIMA Model
model = ARIMA(df.value, order=(1,1,2))
model_fit = model.fit(disp=0)
print(model_fit.summary())
```

Рисунок 2.12 – Приклад побудови моделі ARIMA на Python

2.3 Завантаження даних та підготовка до роботи

Використовуючи систему Kaggle, було створено новий ноутбук та підключено необхідні бібліотеки. Дані бібліотеки потрібні будуть для подальшої роботи. Короткий опис даних бібліотек можна знайти вище. Підключення бібліотек зображено на рисунку 2.13.

На рисунку 2.14 зображено створення додаткових змінних. Вони будуть використовуватись в подальшому. forecasting_period – це кількість годин для прогнозування. measuring_threshold_per_day – це гранична добова доза для параметра PM2.5. Саме цей параметр буде прогнозуватися. predicted_data та models_metrics це додаткові датафрейми для подальшої роботи.

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from matplotlib import pyplot as plt
plt.style.use('bmh')
from datetime import datetime
from scipy.stats import zscore
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error, mean_absolute_error
from prophet import Prophet
from prophet.plot import add_changepoints_to_plot
from statsmodels.tsa.arima.model import ARIMA
import statsmodels.api as sm
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import *
from keras.callbacks import ModelCheckpoint
from keras.optimizers import Adam
from keras.metrics import RootMeanSquaredError
```

Рисунок 2.13 – Підключення бібліотек

```
forecasting_period = 24
measuring_threshold_per_day = 15
predicted_data = pd.DataFrame(columns = [
    'Time',
    'Actual',
    'Prophet',
    'LSTM',
    'ARIMA',
])

models_metrics = pd.DataFrame(columns = ['name', 'r2_score_train', 'r2_score_test', 'rmse_train', 'rmse_test', 'mae_train', 'mae_te
t'])
```

Рисунок 2.14 – Створення додаткових змінних

Далі необхідно підключити дані. Підключення даних до ноутбука зображено на рисунку 2.15. Дані були надані у форматі csv. Використовуючи бібліотеку Pandas, можна легко підключити майже будь-який тип файлів для імпорту даних.

```
]:
data = pd.read_csv("/kaggle/input/air-quality-in-vinnytsia-2023-septe
mber/Air quality in September 2023.csv")
data

/tmp/ipykernel_20/3785616083.py:1: DtypeWarning: Columns (0,2,3,5,
12,13,15,16) have mixed types. Specify dtype option on import or s
et low_memory=False.
  data = pd.read_csv("/kaggle/input/air-quality-in-vinnytsia-2023-
september/Air quality in September 2023.csv")
```

Рисунок 2.15 – Підключення даних

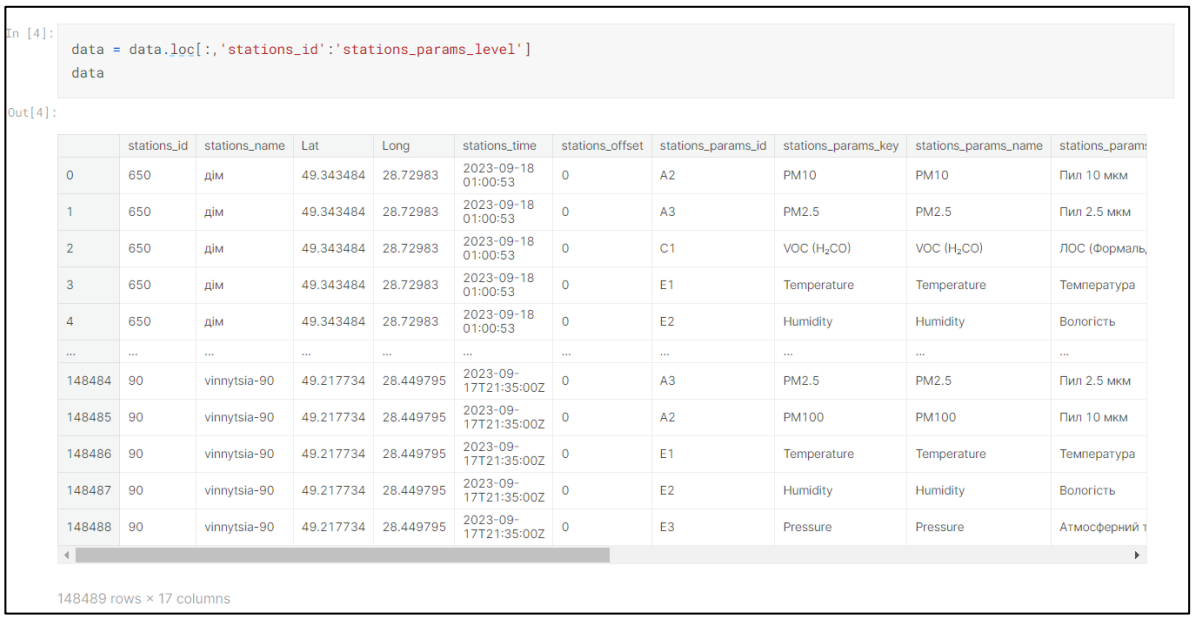
Після підключення є змога оглянути дані які містяться в датасеті. Фрагмент датасету зображено на рисунку 2.16.

	stations_id	stations_name	Lat	Long	stations_time	stations_offset	stations_params_id	stations_params_key	stations_params_name	stations_params
0	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	A2	PM10	PM10	Пил 10 мкм
1	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	A3	PM2.5	PM2.5	Пил 2.5 мкм
2	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	C1	VOC (H_2CO)	VOC (H_2CO)	ЛОС (Формаль
3	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	E1	Temperature	Temperature	Температура
4	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	E2	Humidity	Humidity	Вологість
...	...	...	...	...	...	...	...	...	...	...
148484	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	A3	PM2.5	PM2.5	Пил 2.5 мкм
148485	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	A2	PM100	PM100	Пил 10 мкм
148486	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	E1	Temperature	Temperature	Температура
148487	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	E2	Humidity	Humidity	Вологість
148488	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	E3	Pressure	Pressure	Атмосферний т

148489 rows × 1001 columns

Рисунок 2.16 – Датасет

Як видно з рисунку 2.117, доданий, до ноутбука датасет має 1001 колонку. Велика кількість колонок є пустими та зайвими, їх необхідно прибрати, адже для подальшої роботи вони не потрібні. На рисунку 2.18 зображено оновлений датасет



In [4]:
```
data = data.loc[:,'stations_id':'stations_params_level']
data
```

Out[4]:

	stations_id	stations_name	Lat	Long	stations_time	stations_offset	stations_params_id	stations_params_key	stations_params_name	stations_params
0	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	A2	PM10	PM10	Пил 10 мкм
1	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	A3	PM2.5	PM2.5	Пил 2.5 мкм
2	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	C1	VOC (H_2CO)	VOC (H_2CO)	ЛОС (Формаль,
3	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	E1	Temperature	Temperature	Температура
4	650	дім	49.343484	28.72983	2023-09-18 01:00:53	0	E2	Humidity	Humidity	Вологість
...	...	...	...	...	...	...	...	...	...	...
148484	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	A3	PM2.5	PM2.5	Пил 2.5 мкм
148485	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	A2	PM100	PM100	Пил 10 мкм
148486	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	E1	Temperature	Temperature	Температура
148487	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	E2	Humidity	Humidity	Вологість
148488	90	vinnytsia-90	49.217734	28.449795	2023-09-17T21:35:00Z	0	E3	Pressure	Pressure	Атмосферний т

148489 rows × 17 columns

Рисунок 2.17 – Датасет без зайвих колонок

Даний датасет містить наступні колонки:

- Station_id. Ідентифікатор станції, що виконувала виміри;
- Station_name. Назава станції;
- Lat. Координати розташування станції (ширина);
- Long. Координати розташування станції (довжина);
- Station_time. Час в який станція зробила вимірювання;
- Station_offset.
- Station_params_id. Ідентифікатор параметру, що вимірювався;
- Station_params_key. Ключ параметра, що вимірювався (назва показника PM10, PM2.5 тощо);
- Station_params_name. Ключ параметра, що вимірювався (назва показника PM10, PM2.5 тощо);
- station_params_localName. Ключ параметра, що вимірювався (назва показника PM10, PM2.5 тощо)українською;

– Station_params_unit. Одиниця вимірювання;
– Station_params_localUnit. Одиниця вимірювання українською;
– Station_params_valu. Виміряне значення;
– Station_params_cr.
– Station_params_time. Час коли було здійснено вимірювання.

Після огляду датасету, було визначено, що не всі колонки потрібні для виконання поставленої задачі, тому їх буде прибрано з датасету.

На рисунку 2.17 зображений оновлений датасет.

```
data = data = data[['stations_id', 'stations_time', 'stations_params_name', 'stations_params_value']]
data
```

	stations_id	stations_time	stations_params_name	stations_params_value
0	650	2023-09-18 01:00:53	PM10	4.663
1	650	2023-09-18 01:00:53	PM2.5	4.164
2	650	2023-09-18 01:00:53	VOC (H_2CO)	10.0
3	650	2023-09-18 01:00:53	Temperature	15.413
4	650	2023-09-18 01:00:53	Humidity	62.951
...	...	...	...	...
148484	90	2023-09-17T21:35:00Z	PM2.5	8.6295
148485	90	2023-09-17T21:35:00Z	PM100	14.80825
148486	90	2023-09-17T21:35:00Z	Temperature	17.80325
148487	90	2023-09-17T21:35:00Z	Humidity	56.13625
148488	90	2023-09-17T21:35:00Z	Pressure	99082.50175

148489 rows × 4 columns

Рисунок 2.18 – Оновлений датасет

Колонка stations_params_name містить іменування параметрі, повний список зображено на рисунку 2.19. Деякі з цих показників коротко описані вище.

```
print(data['stations_params_name'].unique())
```

```
['PM10' 'PM2.5' 'VOC (H₂CO)' 'Temperature' 'Humidity' 'Pressure'
'RAD'
 'PM1.0' 'PM100' 'CO' 'O₃' 'stations_params_name']
```

Рисунок 2.19 – Список параметрів.

Як видно з рисунку 2.18, колонка stations_time, має не однорідний формат даних. Далі подальшої роботи з цією колонкою необхідно привести дані у цій колонці до єдиного формату.

Мною було визначено, що певні станції, а саме, ті станції, що мають stations_id 246, 256, 271, 281, 315, 90 мають формат '%Y-%m-%dT%H:%M:%SZ', а інші – '%Y-%m-%d %H:%M:%S'. Приведення даних до єдиного формату представлено на рисунках 2.20 – 2.22.

```
data = data.copy()
```

```
def parse_date(date):
    parsed_date = datetime.strptime(date, '%Y-%m-%dT%H:%M:%SZ')
    return parsed_date.strftime('%Y-%m-%d %H:%M:%S')
```

Рисунок 2.20 – Функція для приведення до єдиного формату

```
data.loc[(data['stations_id'] == 246) | (data['stations_id'] == 256)
                        | (data['stations_id'] == 271) | (data['stat
ions_id'] == 281)
                        | (data['stations_id'] == 274) |(data['stati
ons_id'] == 315)
                        | (data['stations_id'] == 90), 'stations_tim
e'] = data.loc[(data['stations_id'] == 246) | (data['stations_id'] ==
256)
                        | (data['stations_id'] == 271) | (data['stat
ions_id'] == 281)
                        | (data['stations_id'] == 274) |(data['stati
ons_id'] == 315)
                        | (data['stations_id'] == 90), 'stations_tim
e'].apply(parse_date)
```

Рисунок 2.21 – Приведення до єдиного формату

```
data
```

	stations_id	stations_time	stations_params_name	stations_params_value
0	650	2023-09-18 01:00:53	PM10	4.663
1	650	2023-09-18 01:00:53	PM2.5	4.164
2	650	2023-09-18 01:00:53	VOC (H_2CO)	10.0
3	650	2023-09-18 01:00:53	Temperature	15.413
4	650	2023-09-18 01:00:53	Humidity	62.951
...	...	...	...	...
148484	90	2023-09-17 21:35:00	PM2.5	8.6295
148485	90	2023-09-17 21:35:00	PM100	14.80825
148486	90	2023-09-17 21:35:00	Temperature	17.80325
148487	90	2023-09-17 21:35:00	Humidity	56.13625
148488	90	2023-09-17 21:35:00	Pressure	99082.50175

Рисунок 2.22 – Оновлений датасет

Оскільки дане дослідження проводиться для показника PM2.5, то необхідно вибрати лише його. Далі необхідно привести датасет до формату, де колонка з датами називатиметься ds, а колонка із зібраними значеннями – y (рис. 2.23 – 2.25).

```
pm25 = data[data['stations_params_name'] == 'PM2.5']
```

Рисунок 2.23 – Вибір лише показників PM2.5

```
pm25 = pm25[['stations_time', 'stations_params_value']].reset_index(drop=True)
pm25 = pm25.rename(columns={'stations_time': 'ds', 'stations_params_value': 'y'})
pm25['ds'] = pd.to_datetime(pm25['ds'], format = '%Y-%m-%d %H:%M:%S', errors = 'i
gnore')
pm25
```

Рисунок 2.24 – Приведення датасету до формату зазначеного вище

	ds	y
0	2023-09-18 01:00:53	4.164
1	2023-09-18 01:00:22	3.39
2	2023-09-18 01:00:29	7.16
3	2023-09-18 01:00:10	6.395
4	2023-09-18 01:00:26	6.9
...	...	...
23307	2023-09-17 21:35:00	13.26375
23308	2023-09-17 21:35:00	11.91275
23309	2023-09-17 21:35:00	7.983
23310	2023-09-17 21:35:00	9.6665
23311	2023-09-17 21:35:00	8.6295

23312 rows × 2 columns

Рисунок 2.25 – Новий датасет

Після того, як дані приведені до необхідного формату, варто провести візуалізацію даних, щоб побачити як поводять себе дані в часі. На рисунку 2.27 зображено графік який показує динаміку частинок PM2.5 з усіх станцій погодинно. Код для виведення графіку представлено на рисунку 2.26.

```
pm25['y'] = pm25['y'].astype(float)
plt.figure(figsize=(12, 6))
plt.plot(pm25['ds'],pm25['y'])
plt.title('PM2,5')
plt.xlabel('Time')
plt.ylabel('Value')
plt.show()
```

Рисунок 2.26 – Код для виведення графіку

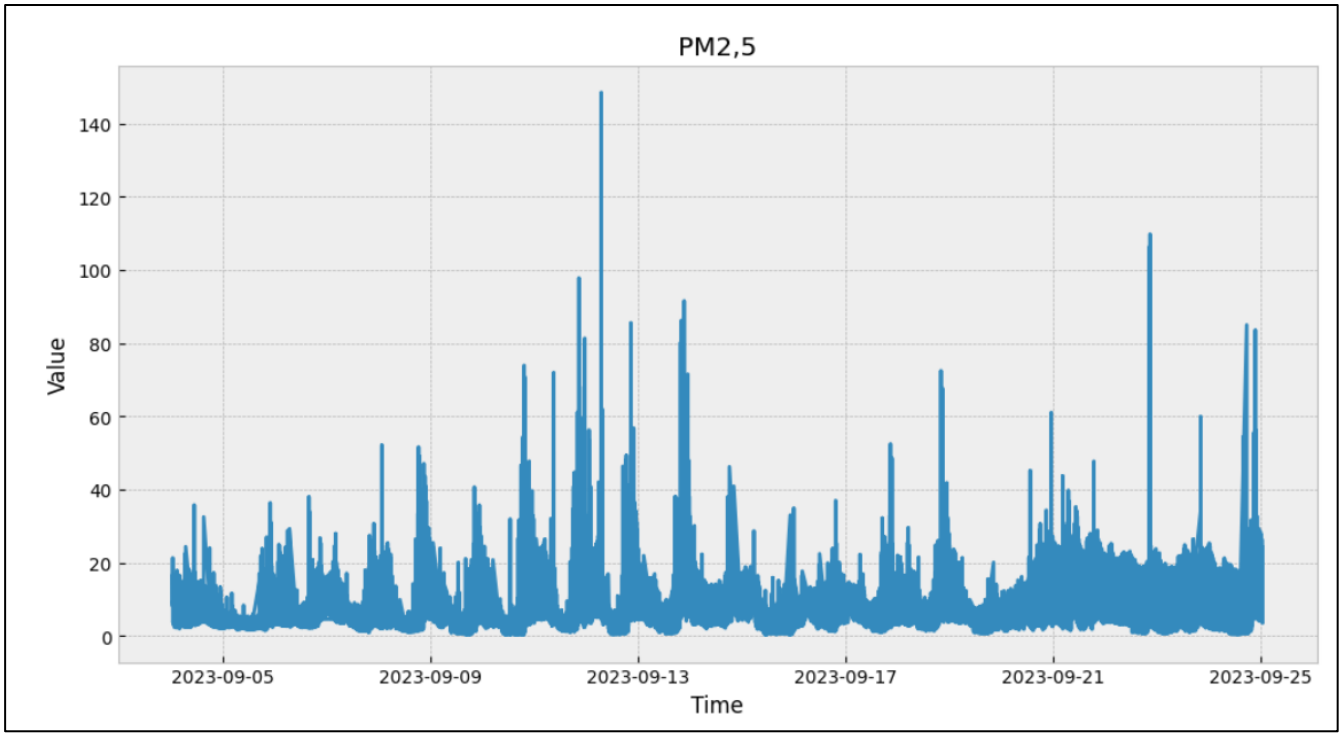


Рисунок 2.27 – Графік динаміки частинок PM2.5

Проведемо візуалізацію середніх показників за кожною станцією погодинно та за одну добу. Це дасть змогу краще зрозуміти динаміку даних у часі та дозволить краще налаштувати моделі для прогнозування. Код для виведення графіку динаміки середньодобових значень частинок PM2.5 зображено на рисунку 2.28. Графік представлено на рисунку 2.29.

```
avg_per_month = pm25_avg_1day['y'].mean()
plt.figure(figsize=(12, 6))
plt.plot(pm25_avg_1day['ds'], pm25_avg_1day['y'], label='Actual PM2.5', color='gr
een')
plt.axhline(y=measuring_threshold_per_day, color='red', label='Thershold PM2.5')
plt.axhline(y=avg_per_month, color='blue', label='Average PM2.5')
plt.xlabel('Time')
plt.ylabel('PM2.5 Values')
plt.title('Dust Pollution (PM2.5)')
plt.legend()
plt.show()
```

Рисунок 2.28 – Код для виведення графіку

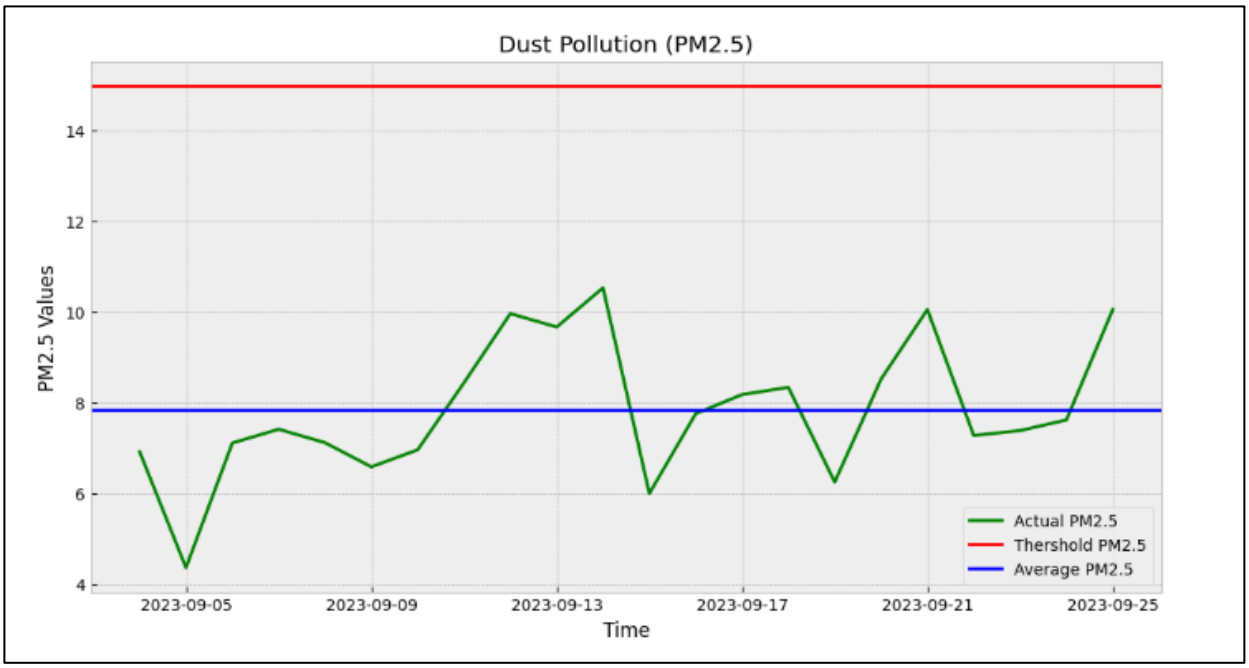


Рисунок 2.29 – Графік динаміки середньодобових значень частинок PM2.5

На рисунку 2.27 представлено графік на якому зображено середньо добові значення показнику PM2.5. Червоною лінією позначено граничне допустиме добове значення цього показника, синьою – середнє значення. Як видно з графіка середньо добові показники знаходяться в межах норми, та не перевищують гранично допустиме значення. Також видно, що пікове значення знаходиться орієнтовно 15 вересня 2023 року. Проведемо погодинну візуалізацію (рис.2.30-2.32).

```
pm25_avg_1hour = pm25.copy()
pm25_avg_1hour['ds'] = pd.to_datetime(pm25_avg_1hour['ds'])
pm25_avg_1hour.set_index('ds', inplace=True)
pm25_avg_1hour = pm25_avg_1hour.resample('60min').mean()
pm25_avg_1hour = pm25_avg_1hour.dropna().reset_index(drop=False)
pm25_avg_1hour
```

Рисунок 2.30 – Код для створення погодинного датафрейму

```
pm25_avg_1hour['y'] = pm25_avg_1hour['y'].astype(float)
avg_per_hour = pm25_avg_1hour['y'].mean()
plt.figure(figsize=(12, 6))
plt.plot(pm25_avg_1hour['ds'],pm25_avg_1hour['y'], label='Actual PM2.5')
plt.axhline(y=avg_per_hour, color='red', label='Average PM2.5')
plt.title('PM2,5')
plt.xlabel('Time')
plt.ylabel('Value')
plt.show()
```

Рисунок 2.31 – Код для виведення графіку з погодинними значеннями

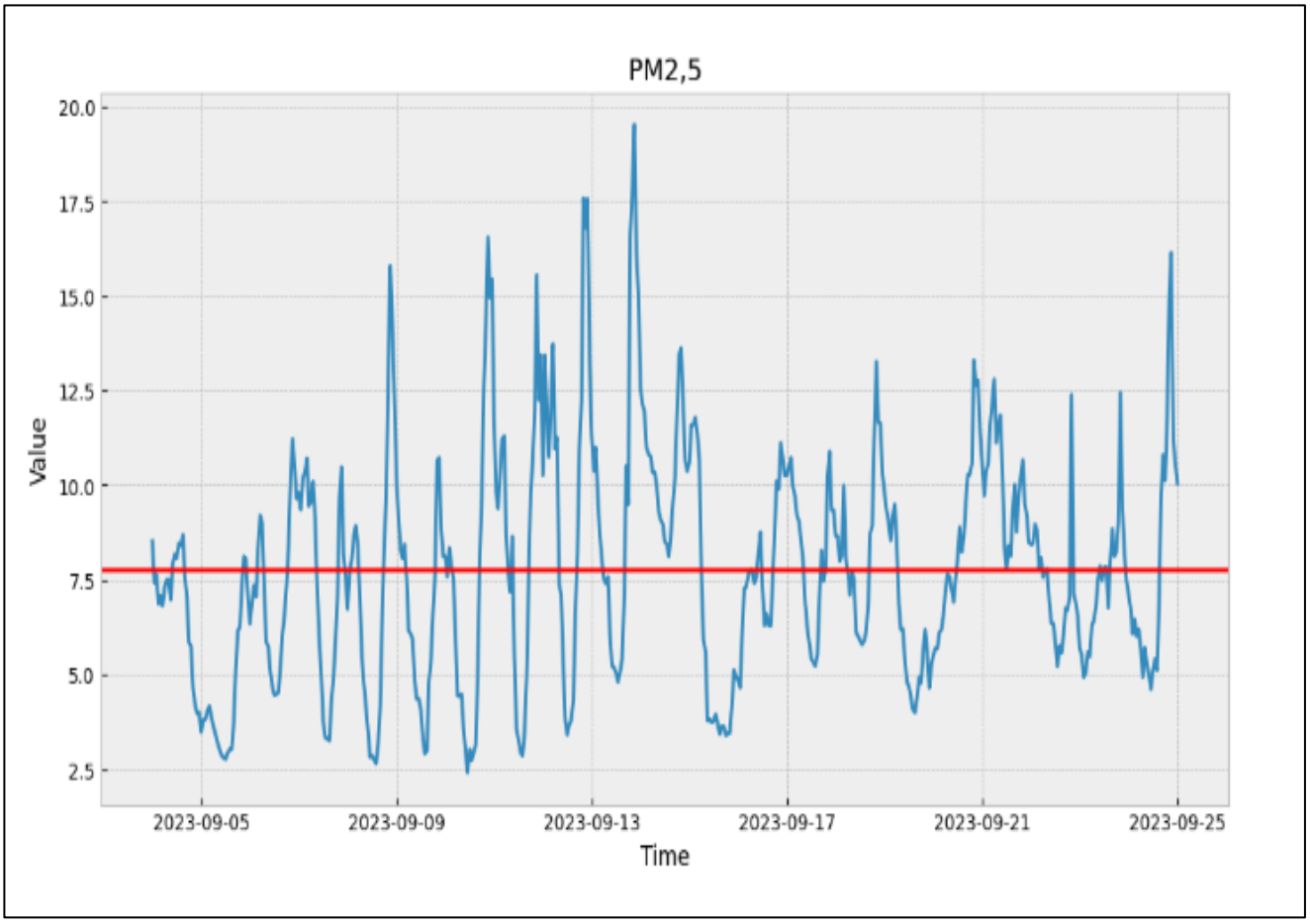


Рисунок 2.32 – Графік динаміки погодинних значень частинок PM2.5

На рисунку 2.32 представлено графік на якому зображено середньо добові значення показнику PM2.5. Червоною лінією позначено середнє значення для цього показника. Графік 2.29 та 2.32 є доволі схожими. Те, що видно одразу – це найбільше значення орієнтовно 15 вересня. Також можна помітити різке зниження показника орієнтовно 5 вересня після чого йде стрімке підвищення.

Наступним кроком необхідно визначити аномальні значення, які є в датасеті. Це необхідно виконати для того, щоб знайти дані які вибиваються з загального контексту, завдяки чому можна буде краще натренувати модель, що дозволить отримати кращі прогнозовані значення (рис.2.33-2.35).

Викиди або аномалії можуть бути результатом помилок вимірювань, невідомих подій або інших аномалій у даних. Фільтрація цих значень дозволяє вам працювати з більш чистим набором даних, що може поліпшити точність аналізу.

Виявлення аномалій може також служити індикатором якості даних. Якщо ви бачите значення, які відхиляються настільки далеко від середнього, це може свідчити про проблеми з якістю даних, і вимагати уваги та подальшого аналізу.

```
]:
mean = pm25_avg_1hour['y'].mean()
std = pm25_avg_1hour['y'].std()
print (mean)
print (std)

7.783744654011948
3.056541903970795

]:
anomalous= pm25_avg_1hour.copy()
anomalous['z_score'] = np.abs((anomalous['y'] - mean) / std)

anomalous = anomalous.loc[abs(anomalous['z_score']) > 1]
anomalous.head()
```

Рисунок 2.33 – Код для пошуку аномальних значень

Для початку необхідно знайти середнє значення та стандартне відхилення. Середнє значення погодинних даних дорівнює 7.78, а стандартне відхилення – 3.06. Аномальні значення обчислюється як абсолютне значення для кожного елемента – середнє і поділене на стандартне відхилення. Після чого відбувається фільтрація даних тих, що перевищують 1. Це зроблено для визначення і відокремлення аномальних або викидних значень від загального набору даних.

Значення z-score вказують на те, наскільки далеко конкретне значення знаходиться від середнього значення в одиницях стандартного відхилення. Зазвичай, якщо абсолютне значення z-оцінки більше 1 (або іншого порогового значення), то це може бути вважатися аномалією.

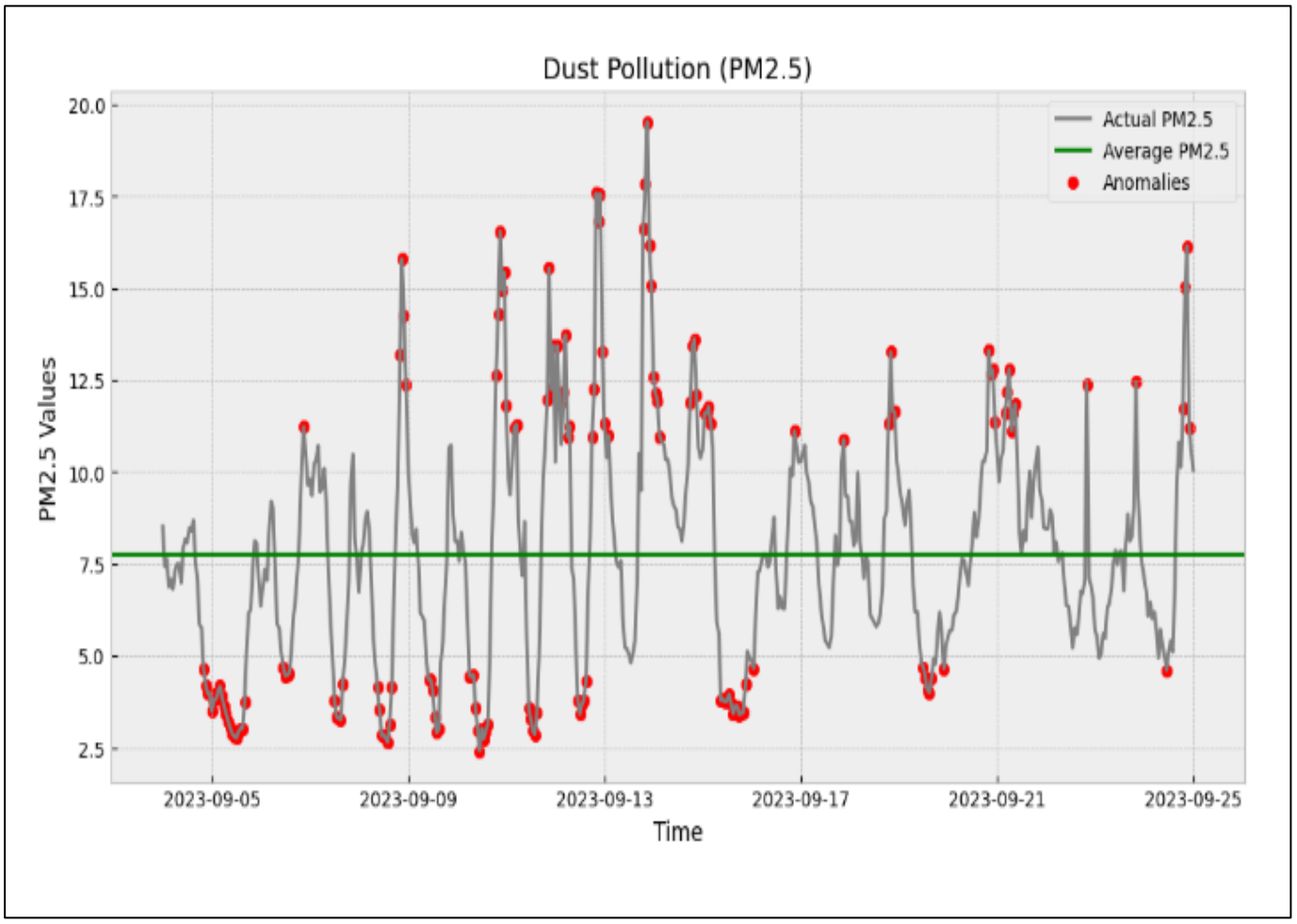


Рисунок 2.34 – Графік аномальних значень

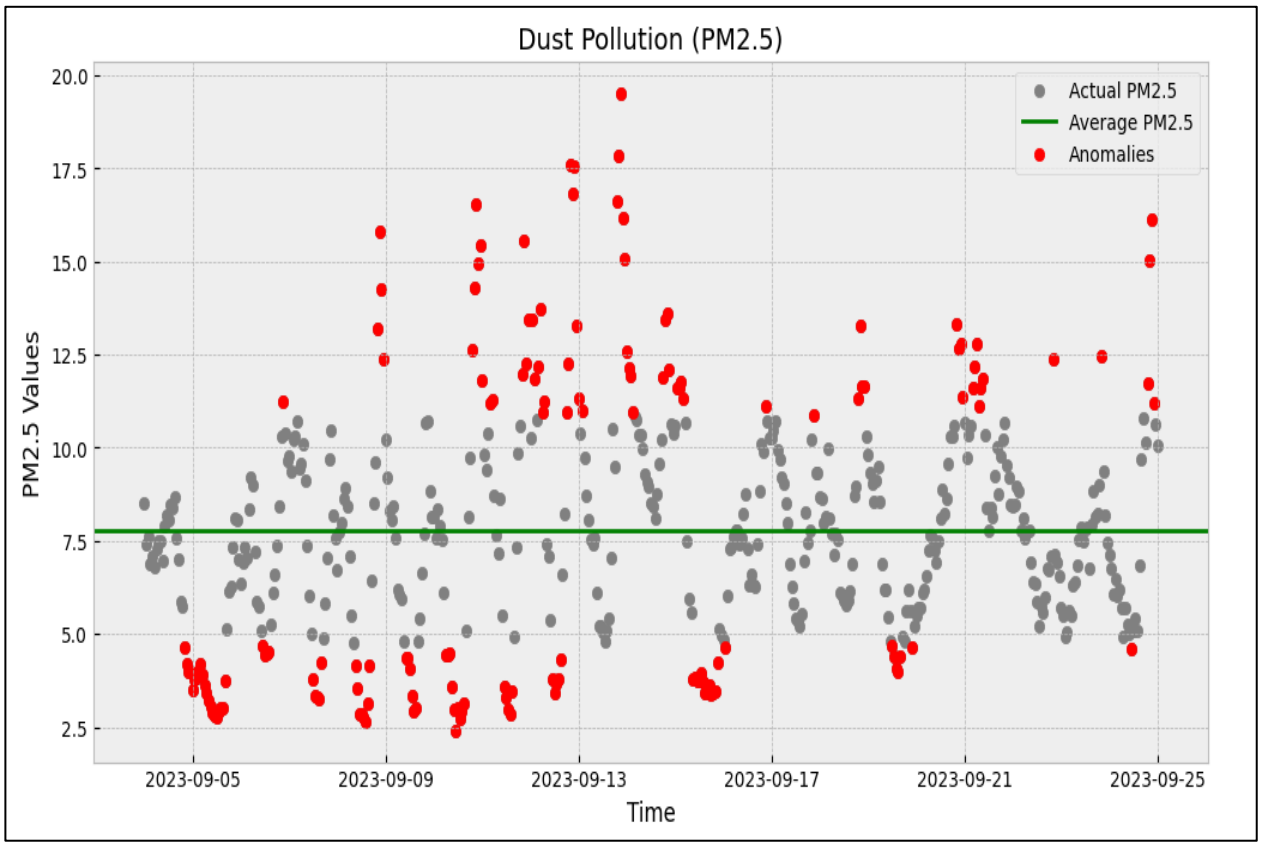


Рисунок 2.35 – Графік аномальних значень

Після проведення аналізу аномальних значень, можна почати побудову моделей для проведення прогнозу. Для цього, необхідно створити тренувальну та тестову вибірки. Розбиття погодинних даних показника PM2.5 на вибірки зображено на рисунку. 2.36-2.38).

```
pm25_test_data = pm25_avg_1hour[-forecasting_period:]
pm25_train_data = pm25_avg_1hour[:-forecasting_period]
predicted_data['Time'] = pm25_test_data['ds']
predicted_data['Actual'] = pm25_test_data['y']
```

Рисунок 2.36 – Розбиття погодинних даних показника PM2.5 на вибірки

Значення -forecasting_period дорівнює 24. Оскільки датафрейм містить середні погодинні значення з усіх станцій, то прогнозування буде здійснюватися на 24 години.

```
pm25_train_data.head(7)
```

	ds	y
0	2023-09-04 00:00:00	8.520163
1	2023-09-04 01:00:00	7.428431
2	2023-09-04 02:00:00	7.628699
3	2023-09-04 03:00:00	6.871228
4	2023-09-04 04:00:00	7.071322
5	2023-09-04 05:00:00	6.816912
6	2023-09-04 06:00:00	7.298957

Рисунок 2.37 – Фрагмент тренувальної вибірки

```
pm25_test_data.head(7)
```

	ds	y
481	2023-09-24 01:00:00	6.763430
482	2023-09-24 02:00:00	6.086140
483	2023-09-24 03:00:00	6.455436
484	2023-09-24 04:00:00	6.016409
485	2023-09-24 05:00:00	6.188614
486	2023-09-24 06:00:00	5.711490
487	2023-09-24 07:00:00	4.938551

Рисунок 2.38 – Фрагмент тестової вибірки

2.4 Висновки

У даному розділі проведено огляд технологій, що використовувалися під час дослідження та написання магістерської кваліфікаційної роботи. Обрано мову програмування, та середовище розробки в якому здійснювалось дослідження. Обрано та описано бібліотеки, що були використані під час роботи над кваліфікаційною роботою.

Проведено детальний огляд моделей, що використовувались для прогнозування якості атмосферного повітря в даній роботі. Створено новий ноутбук та підключено зібрані дані громадського моніторингу якості повітря.

Проведено детальний аналіз даних. Здійснено необхідні маніпуляції з даними аби привести їх до необхідного формату для подальшої роботи. Здійснено візуалізацію погодинних та середньо добових значень.

Проведено аналіз аномальних значень та виконано їх візуалізацію. Підготовлено тренувальний та тестові данні для подальшого прогнозування.

3 СТВОРЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЯКОСТІ ПОВІТРЯ

Перед початку побудови моделі, необхідно підключити бібліотеки для роботи ними. Підключення бібліотек можна знайти вище.

3.1 Побудова моделі Prophet

На рисунку 3.1 – зображено побудову моделі Prophet.

```
model = Prophet(daily_seasonality=True, weekly_seasonality=False, yearly_seasonality=False,
                n_changepoints = 30,
                changepoint_range=1, changepoint_prior_scale = 10,
                holidays=a, seasonality_mode = 'multiplicative')
model.add_seasonality(name='hour', period=1/24, fourier_order=1,
                      mode = 'additive', prior_scale = 10)

model.add_seasonality(name='week', period=7, fourier_order=3,
                      mode = 'additive', prior_scale = 3)
```

Рисунок 3.1 – Модель Prophet

Модель складається з наступних елементів:

- daily_seasonality=True. Додає денну сезонність (24-годинний цикл);
- weekly_seasonality=False. Відключає тижневу сезонність;
- yearly_seasonality=False. Відключає річну сезонність;
- n_changepoints=30. Кількість точок зміни, де модель може адаптуватися до тренду;

- changepoint_range=1. Розмах для визначення, де можуть відбуватися зміни. 1
- changepoint_prior_scale=10. Регулює гладкість тренду. Більше значення зробить тренд більш гнучким;
- holidays=a. Додає аномалії (якщо a – це DataFrame із аномаліями);
- seasonality_mode='multiplicative'. Встановлює мультиплікативний режим для сезонності;

Далі до моделі додається погодинна та щотижнева сезонності. Кожна сезонність має спільні параметри, а саме:

- fourier_order. Кількість компонент у розкладі Фур'є для апроксимації сезонності. Розклад Фур'є є математичним методом, який дозволяє представити будь-яку періодичну функцію або сигнал як суму окремих синусоїд та косинусоїд з різними частотами, амплітудами та фазами. У випадку апроксимації сезонності часового ряду, розклад Фур'є може бути використаний для розкладу циклічних коливань у вигляді суми синусоїд та косинусоїд;
- mode='additive'. Адитивний режим сезонності. В адитивному режимі сезонні компоненти додаються до тренду та інших частин моделі. В мультиплікативному режимі сезонні компоненти множаться на тренд та інші частини моделі. Якщо амплітуда сезонних змін виглядає сталим під час зростання або зменшення рівня часового ряду, то адитивний режим може бути більш підходящим. Якщо амплітуда сезонних змін здається пропорційною рівню часового ряду, то мультиплікативний режим може бути більш підходящим;
- prior_scale. Параметр, що впливає на гладкість сезонності тижнів.

Проведемо прогнозування з використанням моделі Prophet (рис.3.2 – 3.3).

```
model.fit(pm25_train_data)
future = model.make_future_dataframe(periods=forecasting_period, freq='H')
forecast = model.predict(future)
forecasted_values = forecast[['ds', 'yhat']]
```

Рисунок 3.2 – Код для виконання прогнозування

```
fig = model.plot(forecast, xlabel='Time', ylabel='PM2.5')
a = add_changepoints_to_plot(fig.gca(), model, forecast)
plt.show()
```

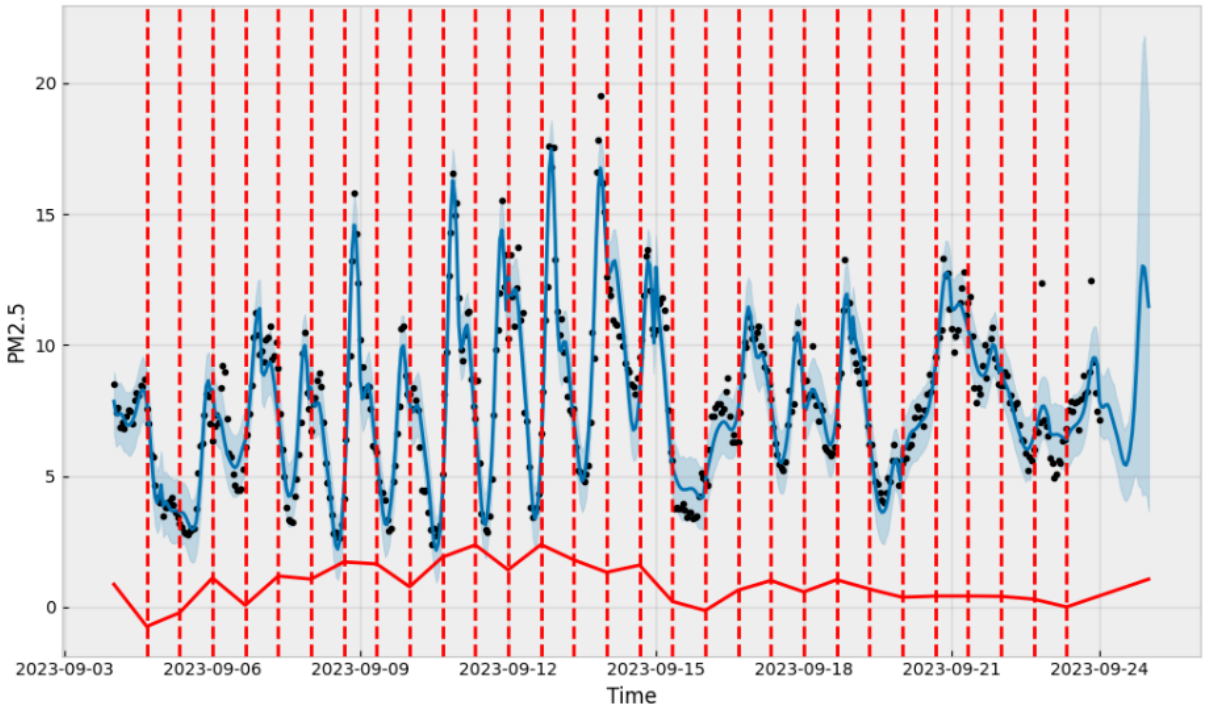


Рисунок 3.3 – Графік прогнозування

На рисунку 3.3 зображена візуалізація прогнозованих погодинних значень для показника PM2.5 з використанням моделі Prophet. Кінець графіку показує дані, що були прогнозовані моделлю. Подивимось на них ближче (рис. 3.4 – 3.6).

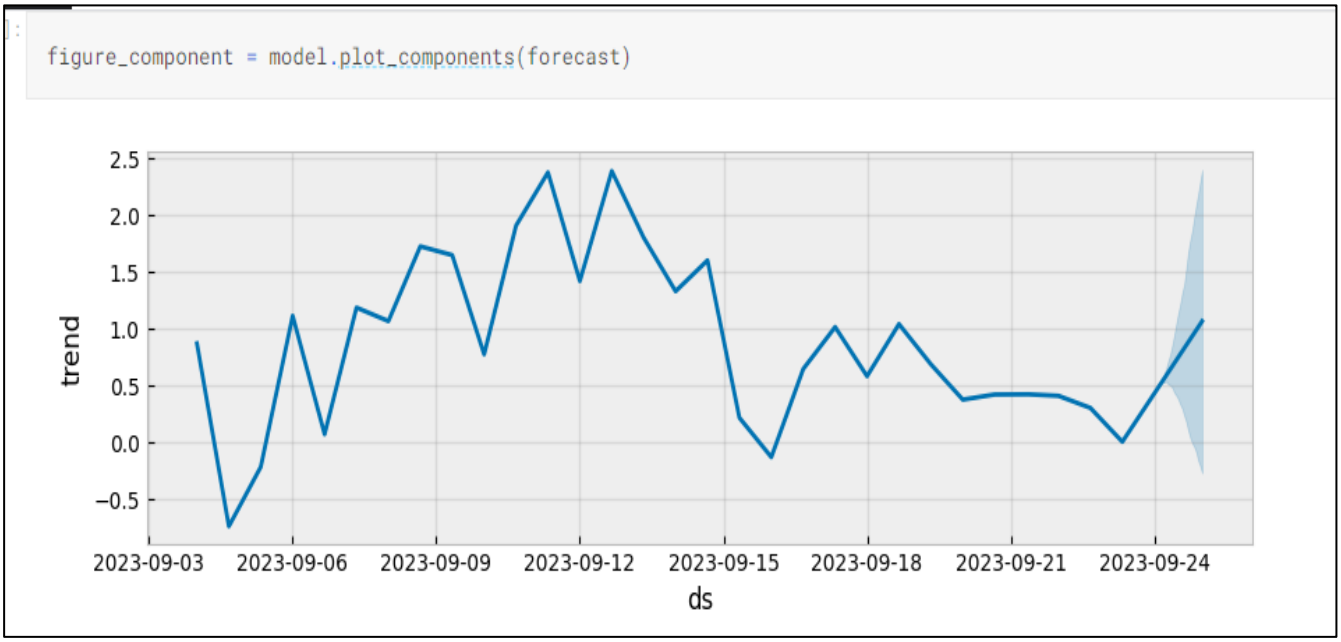


Рисунок 3.4 – Графік тренду

На рисунку 3.4 показано лінію тренду. Лінія тренду у часовому ряді показує загальну тенденцію часового ряду та вказує на його основну напрямок розвитку. Аналіз лінії тренду допомагає визначити основний рух або зміну у середньому значенні часового ряду протягом певного періоду часу. На графіку видно тренд того, як розташовані дані в часі.

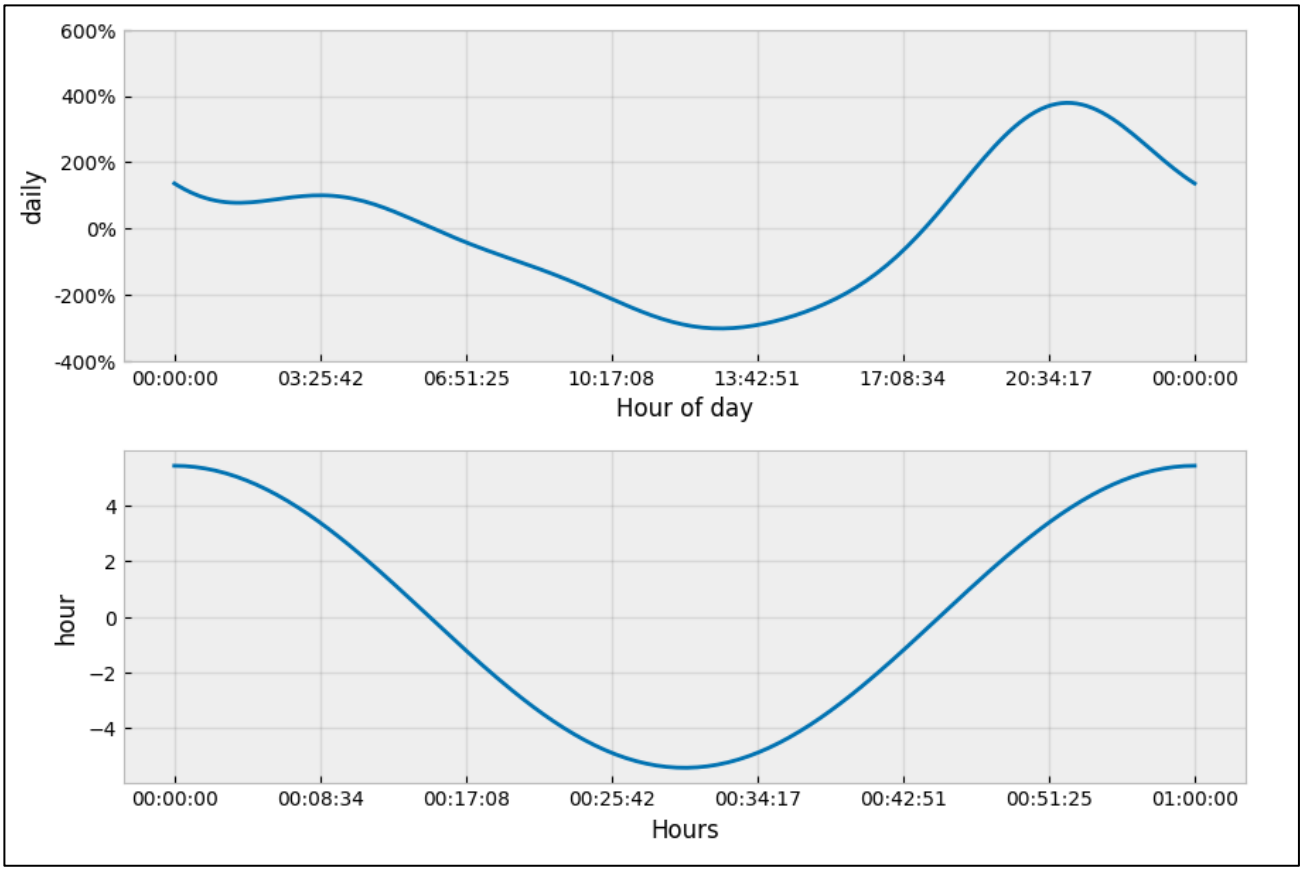


Рисунок 3.5 – Графік щоденної та щогодинної сезонності

Як видно з рисунку 3.5, а саме з щоденної сезонності, найбільші значення припадають на вечірній час. Це не дивно, оскільки багато людей, їдуть з роботи на автомобілях чим підвищують вміст шкідливих речовин у повітрі. Дивним є те, що з ранку такого піку не було, оскільки, на мою думку, повинна бути та сама ситуація.

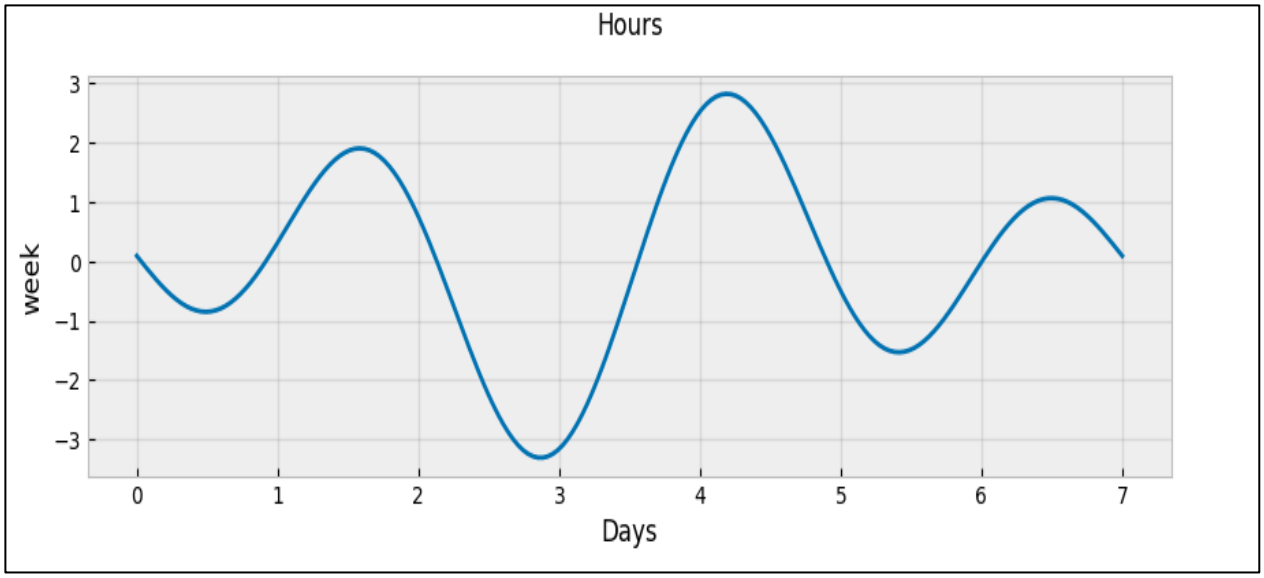


Рисунок 3. 6 – щотижнева сезонність

На рисунку 3.6 зображена щотижнева сезонність даних. З графіку видно, що до середини тижня показники значно падають, проте одразу під кінець тижня вони збільшуються, а потім знову падають.

Необхідно провести візуалізацію отриманих результатів після прогнозування. Код для візуалізації представлений на рисунку 3.7. Графік прогнозування зображено на рисунку 3.8.

```
plt.figure(figsize=(12, 6))
plt.plot(pm25_train_data['ds'], pm25_train_data['y'], label='Actual PM2.5', color='green')
plt.plot(b['ds'], b['yhat'], label='Predicted PM2.5', color='red')
plt.title('Train Data PM2.5')
plt.legend()
plt.show()
```

Рисунок 3.7 – Код для виведення графіку

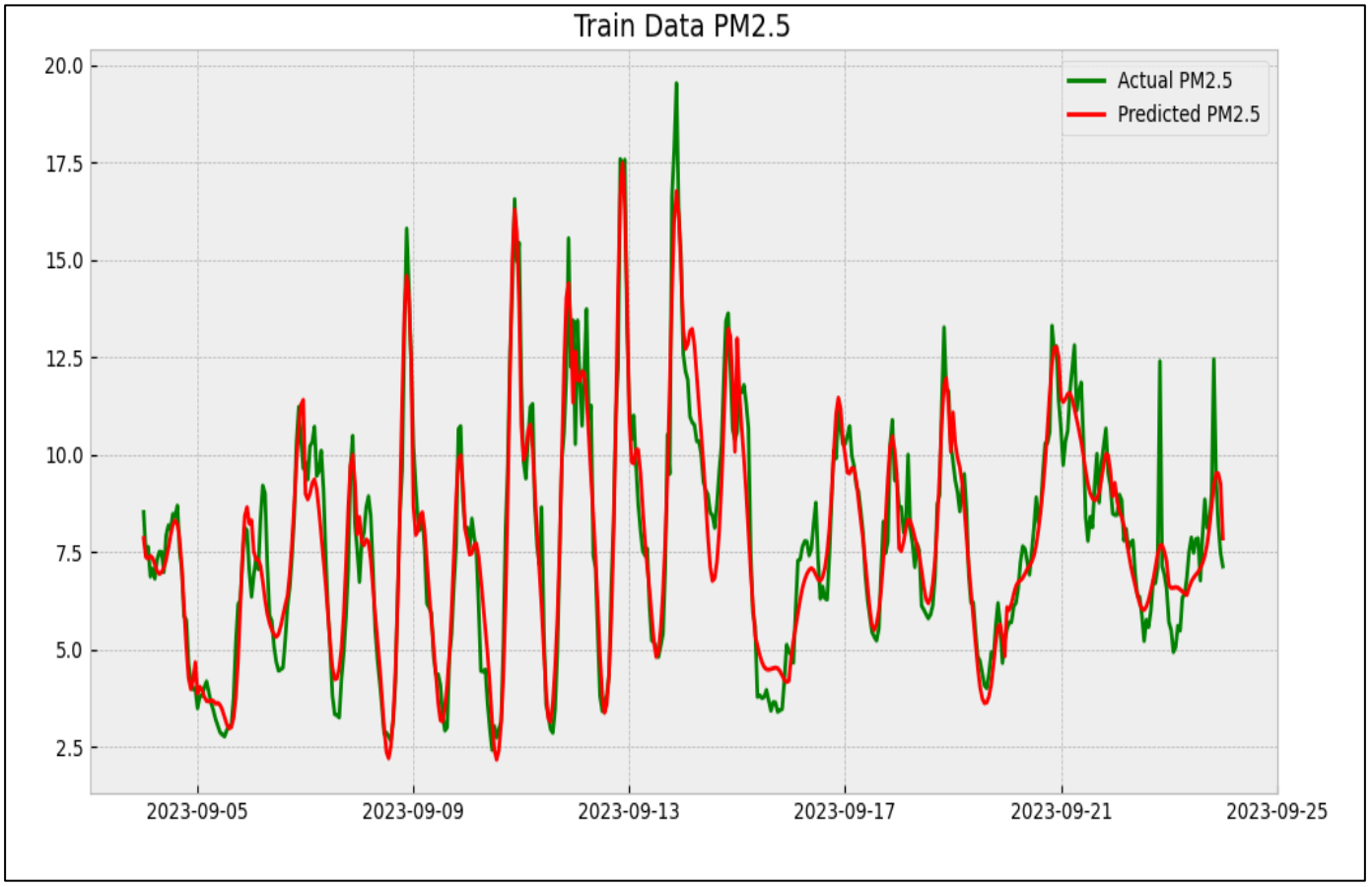


Рисунок 3.8 – Графік прогнозованих тренувальних значень у порівнянні з актуальними

З рисунку 3.8 видно динаміку тренувальних та прогнозованих даних. З графіку видно, що модель не до кінця розуміє деякі пікові точки. До прикладу останні два сплески, модель не змогла їх розпізнати в повній мірі, через, що прогнозований результат виявився не зовсім точним.

Код для візуалізації зображено на рисунку 3.9. Візуалізація тестових даних зображена на рисунку 3.10

```
plt.figure(figsize=(12, 6))
plt.plot(pm25_test_data['ds'], pm25_test_data['y'], label='Actual PM2.5', color='green')
plt.plot(prophet_predicted_values['ds'], prophet_predicted_values['yhat'], label='Predicted PM2.5', color='red')
plt.title('Test Data PM2.5')
plt.legend()
plt.show()
```

Рисунок 3.9 – Код для візуалізації

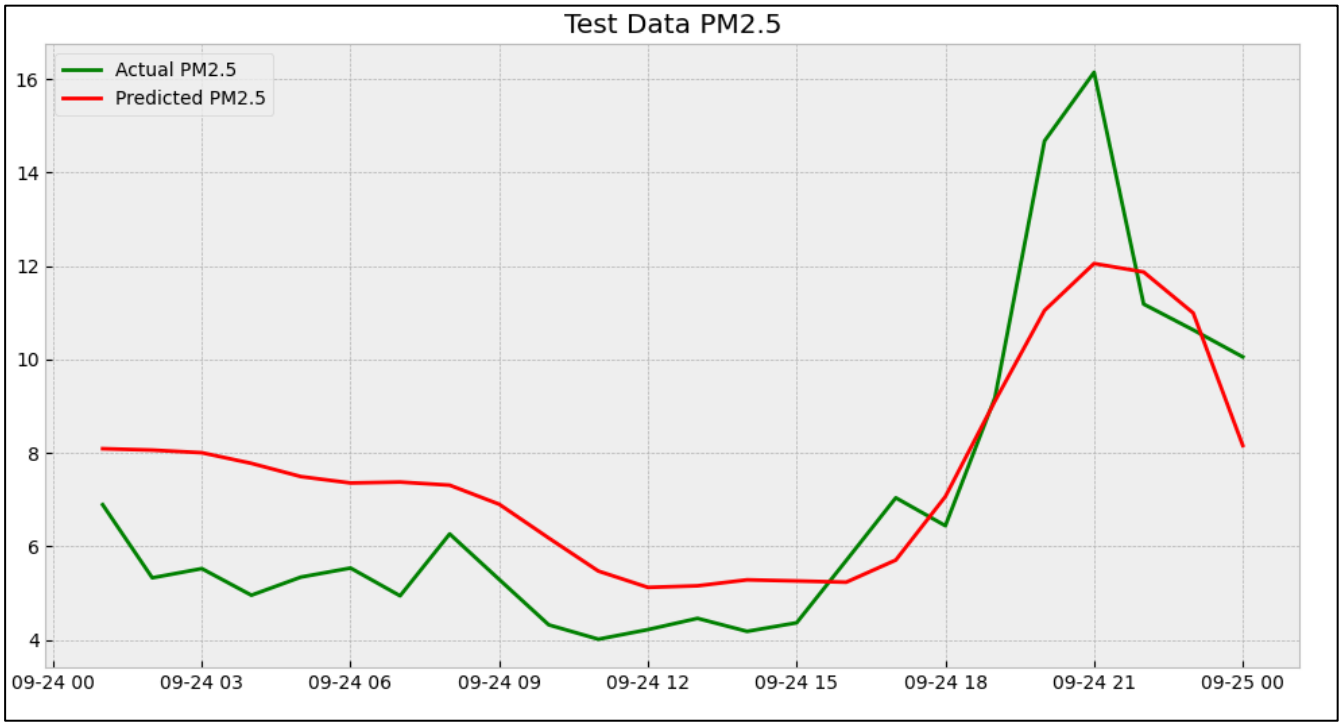


Рисунок 3.10 – Графік прогнозованих тестових значень у порівнянні з актуальними

З графіку, зображеному на рисунку 3.10 видно, що через те, що модель не змогла розпізнати деякі моменти на тренувальних даних – це вилилося в те, що прогнозовані значення в порівнянні з актуальними тестовими виявились не зовсім точними.

Необхідно провести оцінку роботи моделі за певним метриками (рис.3.11). Значення доданих метрик зображено на рисунку 3.12.

```
models_metrics.loc[0, 'name'] = 'Prophet'

models_metrics.loc[0,'r2_score_train'] = round(r2_score(pm25_train_data['y'], b['yhat']),2)
models_metrics.loc[0,'r2_score_test'] = round(r2_score(pm25_test_data['y'], prophet_predicted_values['yhat']),2)

models_metrics.loc[0,'rmse_train'] =  round(np.sqrt(mean_squared_error(pm25_train_data['y'], b['yhat'])),2)
models_metrics.loc[0,'rmse_test'] = round(np.sqrt(mean_squared_error(pm25_test_data['y'], prophet_predicted_values['yhat'])),2)

models_metrics.loc[0,'mae_train'] = round(mean_absolute_error(pm25_train_data['y'], b['yhat']),2)
models_metrics.loc[0,'mae_test'] = round(mean_absolute_error(pm25_test_data['y'], prophet_predicted_values['yhat']),2)

models_metrics
```

Рисунок 3.11 – Приклад додавання метрик до датафрейму

41]:

	name	r2_score_train	r2_score_test	rmse_train	rmse_test	mae_train	mae_test
0	Prophet	0.91	0.73	0.93	1.69	0.71	1.47

Рисунок 3.12 – Значення метрик

Для оцінки усіх моделей використовувались наступні метрики.

R2_Score. r2_score, або коефіцієнт детермінації який є метрикою для оцінки якості моделей у машинному навчанні. r2_score визначає ступінь варіації у цільових значеннях, який може бути пояснений моделлю. Це числова оцінка, яка варіюється від 0 до 1, де 1 вказує на ідеальне узгодження між передбаченими та спостережуваними значеннями, а 0 вказує на відсутність узгодження. Якщо r2_score дорівнює 1, це вказує на ідеальне узгодження між передбаченими та фактичними значеннями. Якщо r2_score дорівнює 0, це означає, що модель не вчиться нічого корисного з даних. Якщо r2_score від'ємний, це означає, що модель гірше працює, ніж просто використання середнього значення фактичних даних для прогнозування.

Вираховується за наступною формулою:

$$R^2 = 1 - \frac{D[y[x]]}{D[y]} = 1 - \frac{\sigma^2}{\sigma_y^2}, \quad (3.1)$$

де D[y] – дисперсія випадкової велични;

D[y[x]] – умовна дисперсія залежної змінної;

RMSE (Root Mean Squared Error) є ще однією популярною метрикою для оцінки якості прогнозів у задачах регресії в машинному навчанні. Вона вимірює ступінь відхилення між спостережуваними і передбаченими значеннями, а також враховує абсолютні розбіжності між ними. Основна ідея RMSE полягає в тому, щоб взяти квадрат кожної абсолютної розбіжності між спостережуваним і передбаченим значеннями, знайти середнє значення цих квадратів, а потім взяти квадратний корінь цього середнього значення. Використання RMSE допомагає оцінити точність прогнозів моделі та порівняти різні моделі за їх ефективністю.

Вираховується за наступною формулою:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y_i})^2}, \quad (3.2)$$

де Y – вектор спостережуваних значень;

$\hat{Y}$ – передбачені значення;

n – величина вибірки.

MAE (Mean Absolute Error) є ще однією метрикою для вимірювання точності моделей у задачах регресії в машинному навчанні. Ця метрика визначає середнє абсолютне значення розбіжності між фактичними і передбаченими значеннями. MAE вимірює абсолютні відхилення між спостережуваними і передбаченими значеннями. MAE є простою метрикою, оскільки вона просто обчислює середнє арифметичне абсолютних відхилень. Вона вимірює середню величину похибки без залучення напрямку. Що означає, що величина помилки не залежить від того, чи є

вона перепрогнозом чи недопрогнозом. Використання MAE дозволяє отримати простий показник, який відображає середню величину похибки моделі.

Вираховується за наступною формулою:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_y - \widehat{Y}_\iota\right|, \tag{3.3}$$

де Y – вектор спостережуваних значень;

$\hat{Y}$ – передбачені значення;

n – величина вибірки.

На відміну від середньоквадратичних помилок, де використовується квадрат різниці, MAE є лінійною оцінкою, тому вага різниці однакова незалежно від діапазону. Наприклад, помилка між 2 і 4 буде в два рази менше, ніж між 4 і 8, тобто. 2 і 4 відповідно (у той час як при квадратичній оцінці 4 і 16, тобто в 4 рази менше).

3.2 Побудова моделі LSTM

Наступною моделлю буде модель LSTM. Для початку потрібно підготувати дані належним чином аби модель могла працювати з ними. Код для приведення даних необхідного формату для моделі LSTM зображено на рисунку 3.13

```
lstm_data = pm25_avg_1hour.copy()

values = lstm_data['y'].values.reshape(-1, 1)

scaler = MinMaxScaler(feature_range=(0, 1))
scaled_values = scaler.fit_transform(values)

train_size =  int(len(scaled_values) - forecasting_period)
train_data = scaled_values[:train_size, :]
test_data = scaled_values[train_size:, :]

def create_sequences(data, sequence_length):
    X = []
    y = []
    for i in range(len(data) - sequence_length):
        a = data[i:(i+sequence_length), 0]
        X.append(a)
        y.append(data[i + sequence_length, 0])
    return np.array(X), np.array(y)

sequence_length = 1

X_train, y_train = create_sequences(train_data, sequence_length)
X_test, y_test = create_sequences(test_data, sequence_length)
```

Рисунок 3.13 – Приведення даних для моделі LSTM

Спочатку створюється копія даних. Далі за допомогою MinMaxScaler проводиться масштабування значень до діапазону від 0 до 1. Далі відбувається розбиття на тестові та тренувальні дані. Після чого функція create_sequences створює послідовності для навчання та тестування. Використовуючи величину sequence_length, вона формує пари (X, y), де X – послідовність, а y – наступне значення після послідовності.

Отже, після виконання цього коду, X_train, y_train, X_test та y_test будуть готові для використання у навчанні моделі LSTM для прогнозування часового ряду.

Далі необхідно передати дані моделі для прогнозування. Будова моделі та прогнозування зображені на рисунках 3.14 – 3.15.

```
lstm_model = Sequential()

lstm_model.add(LSTM(128, 'relu', return_sequences=True, input_shape=(sequence_length, 1)))
lstm_model.add(LSTM(64))
lstm_model.add(Dense(32))
lstm_model.add(Dense(1))

cp = ModelCheckpoint('modem', save_best_only=True)
```

Рисунок 3.14 – Модель LSTM

Спочатку створюється порожня модель, яка визначатиме нейронну мережу з послідовними шарами. Додається перший шар LSTM з 128 нейронами, активацією 'relu', вхідною формою (sequence_length, 1). Додається другий шар LSTM з 64 нейронами. Додається Dense шар із 32 нейронами. Додається вихідний Dense шар із 1 нейроном, оскільки це задача регресії. Використовуючи ModelCheckpoint, зберігається найкраща модель під час навчання, вказуючи шлях до збереження ('modem').

Далі йде етап компіляції моделі та прогнозування (рис. 3.15).

```
lstm_model.compile(optimizer='adam', loss='mean_squared_error', metrics=[RootMeanSquaredError()])

lstm_model.fit(X_train, y_train, epochs=200, batch_size=32, callbacks=[cp], verbose=False)

predictions_train = lstm_model.predict(X_train)
predictions_test = lstm_model.predict(X_test)
lstm_model.summary()
```

```
15/15 [==============================] - 1s 2ms/step
1/1 [==============================] - 1s 700ms/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 1, 128)            66560

 lstm_1 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 32)                2080

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 118,081
Trainable params: 118,081
Non-trainable params: 0
_________________________________________________________________
```

Рисунок 3.15 – Компіляція моделі та прогнозування

На рисунку 3.15 визначається процес компіляції моделі з оптимізатором 'adam', функцією втрати 'mean_squared_error' та метрикою – середньою квадратичною помилкою (RootMeanSquaredError). Викликається метод fit для тренування моделі з тренувальними даними X_train та y_train. казано 200 епох тренування, розмір пакета (batch_size) рівний 32. Використовується ModelCheckpoint для збереження найкращої моделі. Викликаються методи predict для отримання прогнозів моделі на тренувальному та тестовому наборах.

Далі необхідно інвертувати дані аби мати можливість провести оцінку отриманих результатів та візуалізувати їх (рис. 3.16).

```
predictions_train = scaler.inverse_transform(predictions_train)
train_data = scaler.inverse_transform(train_data)

predictions_test = scaler.inverse_transform(predictions_test)
test_data = scaler.inverse_transform(test_data)

train_dates = lstm_data['ds'].values[:train_size-sequence_length]
test_dates = lstm_data['ds'].values[train_size+sequence_length:]
```

Рисунок 3.16 – Код для інвертування даних

Після того, як дані було переведено в потрібний формат, потрібно їх візуалізувати та провести оцінку за метриками, які були описані вище. Код для візуалізації роботи моделі зображено на рисунку 3.17. Графік зображено на рисунку 3.18.

```
trainPredictPlot = np.empty_like(scaled_values)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[sequence_length:len(predictions_train)+sequence_length, :] = predictions_train

testPredictPlot = np.empty_like(scaled_values)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(predictions_train)+(sequence_length*2):len(scaled_values), :] = predictions_test
plt.figure(figsize=(12, 6))

plt.plot(scaler.inverse_transform(scaled_values), label='Actual PM2.5', color='green')
plt.plot(trainPredictPlot, label='Predicted Train PM2.5', color='red')
plt.plot(testPredictPlot, label='Predicted Test PM2.5', color='yellow')
plt.legend()
plt.show()
```

Рисунок 3.17 – Код для візуалізації роботи моделі LSTM

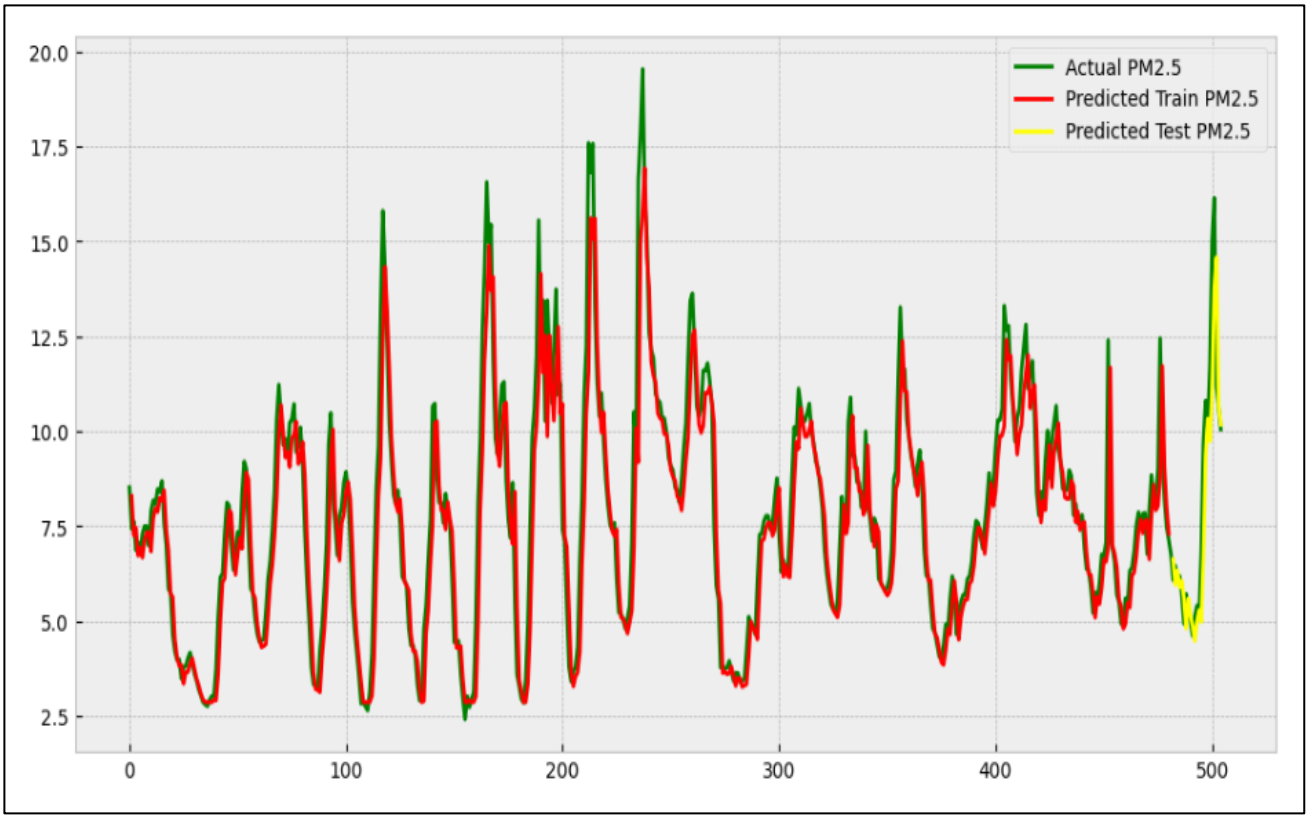


Рисунок 3.18 – Візуалізовані дані після прогнозу моделі LSTM

На рисунку 3.18 зображений графік візуалізації тренувальних та тестових даних у порівнянні з прогнозованими. Як видно з графіка модель LSTM краще впоралася з прогнозуванням. На рисунку 3.19 зображено прогнозовані тестові дані у порівнянні з дійсними. Це потрібно для кращого розуміння, наскільки точно модель виконала прогнозування.

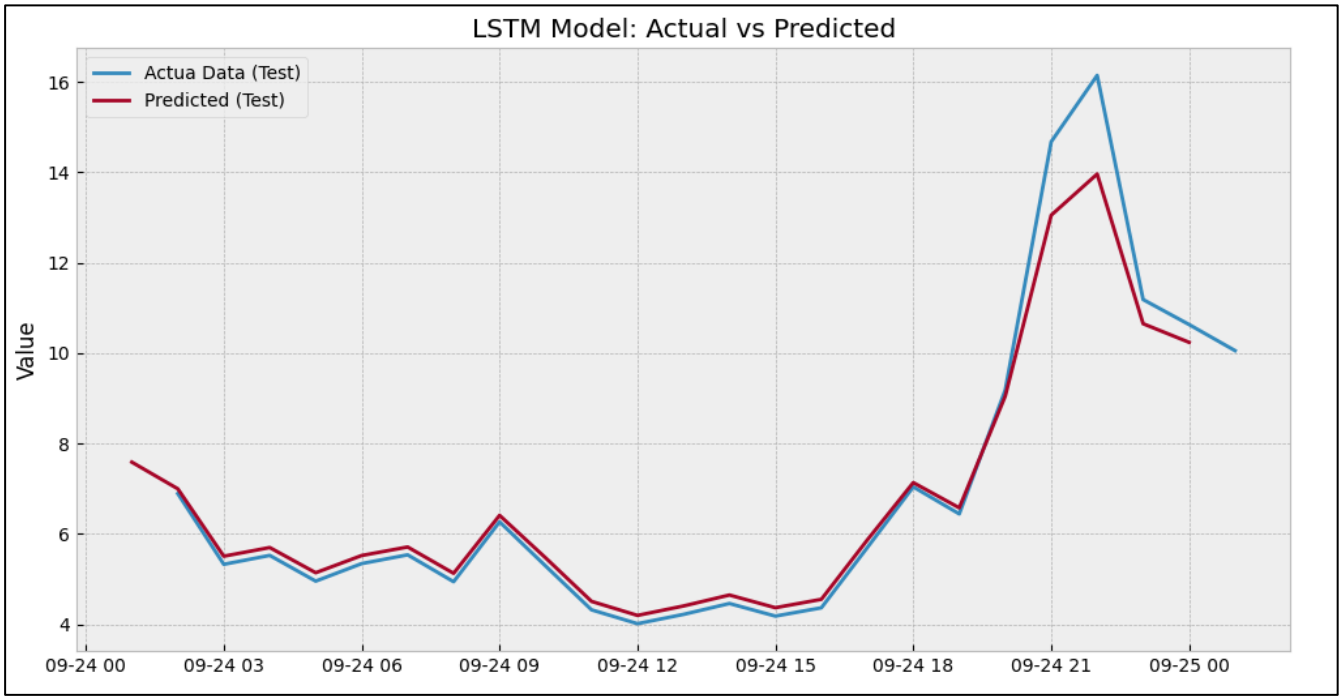


Рисунок 3.19 – Прогнозовані у порівнянні з дійсними тестовими

З графіку видно, що модель LSTM набагато краще впоралася з прогнозуванням даних аніж модель Prophet. Проте, все ще є невеличкий зсув вправо. Незважаючи на це, на мою думку, результат є доволі гарним. Перевіримо метрики. Метрики роботи моделі LSTM зображено на рисунку 3.20.

	name	r2_score_train	r2_score_test	rmse_train	rmse_test	mae_train	mae_test
0	Prophet	0.91	0.73	0.93	1.69	0.71	1.47
1	LSTM	0.99	0.89	0.426	1.502	0.295	1.015

Рисунок 3.20 – Метрики

Як видно з рисунку 3.20 на якому зображено оцінка роботи моделей LSTM та Prophet, модель LSTM справилась краще за свого попередника. Наступною буде модель ARIMA.

3.3 Побудова моделі ARIMA

Побудова моделі ARIMA зображена на рисунку 3.21.

```
arima_data = pm25_avg_1hour.copy()
arima_test_data = pd.Series(arima_data['y'][-forecasting_period:].values, index=range(forecasting_period))
arima_train_data = arima_data['y'][:-forecasting_period]


model = ARIMA(arima_train_data, order = (5, 1, 3))
model_fit = model.fit()
ARIMA_train_predictions = model_fit.predict()


history = [x for x in arima_train_data]
ARIMA_predicted = list()

for t in range(len(arima_test_data)):

    model = ARIMA(history, order = (5, 1, 3))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat = output[0]
    ARIMA_predicted.append(yhat)
    obs = arima_test_data[t]
    history.append(obs)
```

Рисунок 3.21 – модель ARIMA

Спочатку створюється копія вихідних даних, а також відокремлюється тестовий період для подальшого використання. Визначається модель ARIMA з параметрами (5, 1, 3).

Перший параметр вказує на кількість попередніх значень часового ряду, які використовуються для прогнозування наступного значення. Зазвичай його називають "порядок авторегресії".

Другий параметр вказує на кількість різниць, які застосовуються до часового ряду для стабілізації його змінності.

Третій параметр вказує на кількість попередніх значень шуму (залишків) часового ряду, які враховуються для прогнозування наступного значення. Зазвичай його називають "порядок ковзного середнього".

Далі модель навчається (fit) на тренувальних даних, здійснюється прогноз для тренувального періоду. Створюється змінна history, яка містить дані тренувального періоду. У циклі для кожного моменту часу тестового періоду використовується модель ARIMA для прогнозування наступного значення. Значення прогнозів додаються до списку ARIMA_predicted. Фактичні значення з тестового періоду додаються до історії для подальшого використання при прогнозуванні.

Необхідно візуалізувати прогнозовані дані, аби порівняти з дійсними (3.22 - 3.24).

```
plt.figure(figsize=(12, 6))
plt.plot(arima_data[-forecasting_period:]['ds'], arima_test_data.values[:train_size-sequence_length], label='Actual (Test)')
plt.plot(arima_data[-forecasting_period:]['ds'], ARIMA_predicted, label='Predicted (Test)')
plt.title('ARIMA Model: Actual vs Predicted')
plt.legend()
plt.tight_layout()
plt.show()
```

Рисунок 3.22 – Код для візуалізації

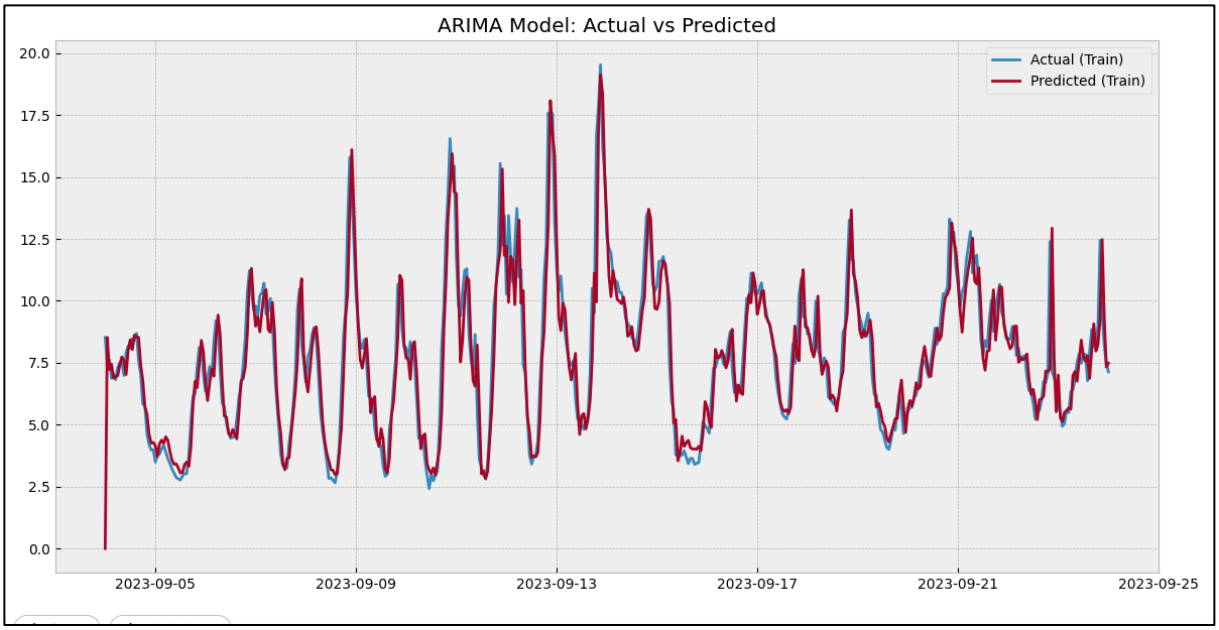


Рисунок 3.23 – Прогнозовані у порівнянні з дійсними тренувальними

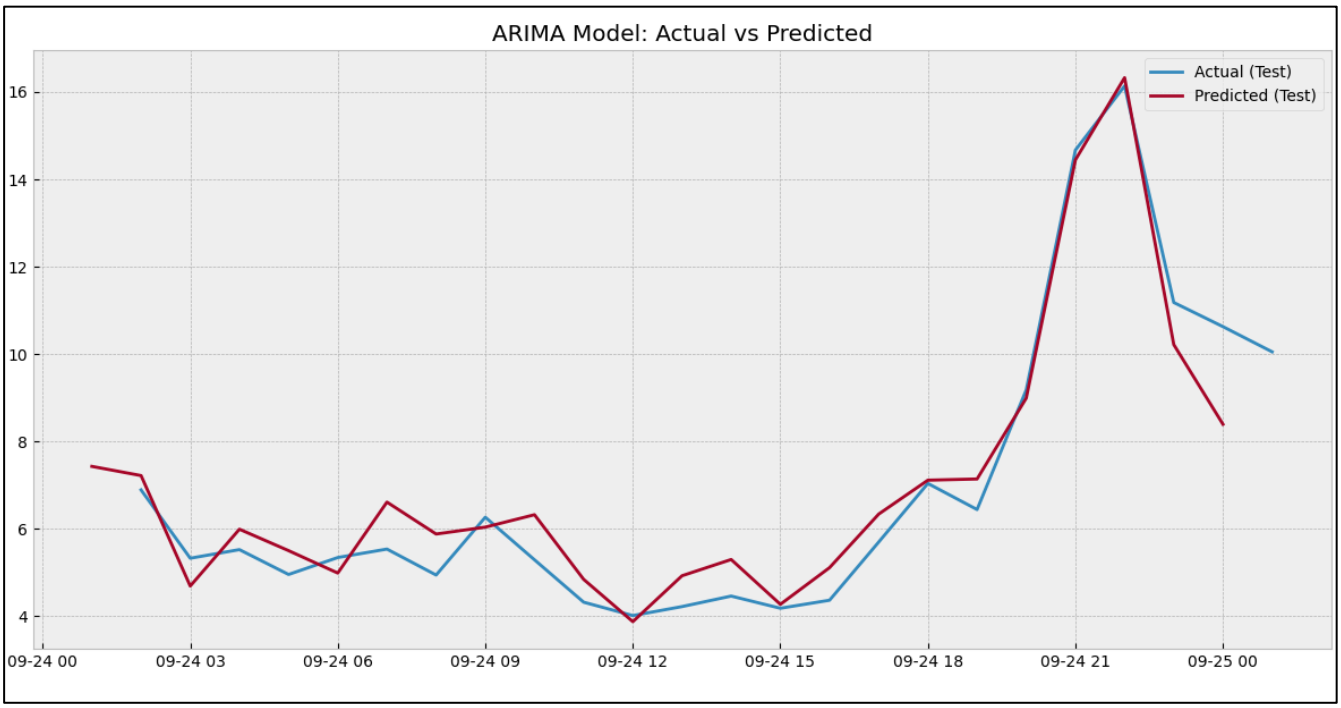


Рисунок 3.24 – Прогнозовані у порівнянні з дійсними тестовими

Як видно з графіку модель ARIMA виконала досить точний прогноз. Якщо порівнювати з LSTM, то деякі моменти модель ARIMA виконала краще, а деякі

гірше. Варто подивитись на метрики (рис. 3.25), а також на загальний графік для трьох моделей (рис. 3.26 – 3.27).

	name	r2_score_train	r2_score_test	rmse_train	rmse_test	mae_train	mae_test
0	Prophet	0.91	0.73	0.93	1.69	0.71	1.47
1	LSTM	0.99	0.89	0.426	1.502	0.295	1.015
2	ARIMA	0.85	0.82	1.17	1.38	0.78	0.91

Рисунок 3.25 – Метрики для трьох моделей

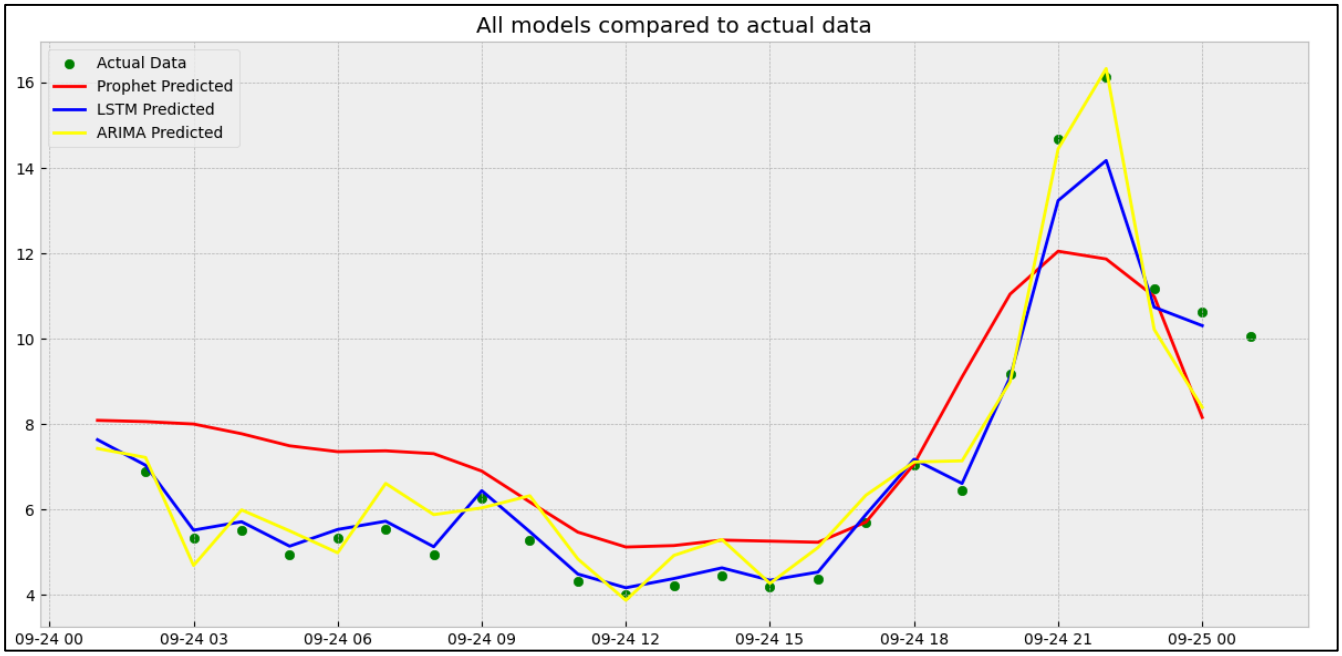


Рисунок 3.26 – Графік прогнозування усіх моделей у порівнянні з дійсними даними

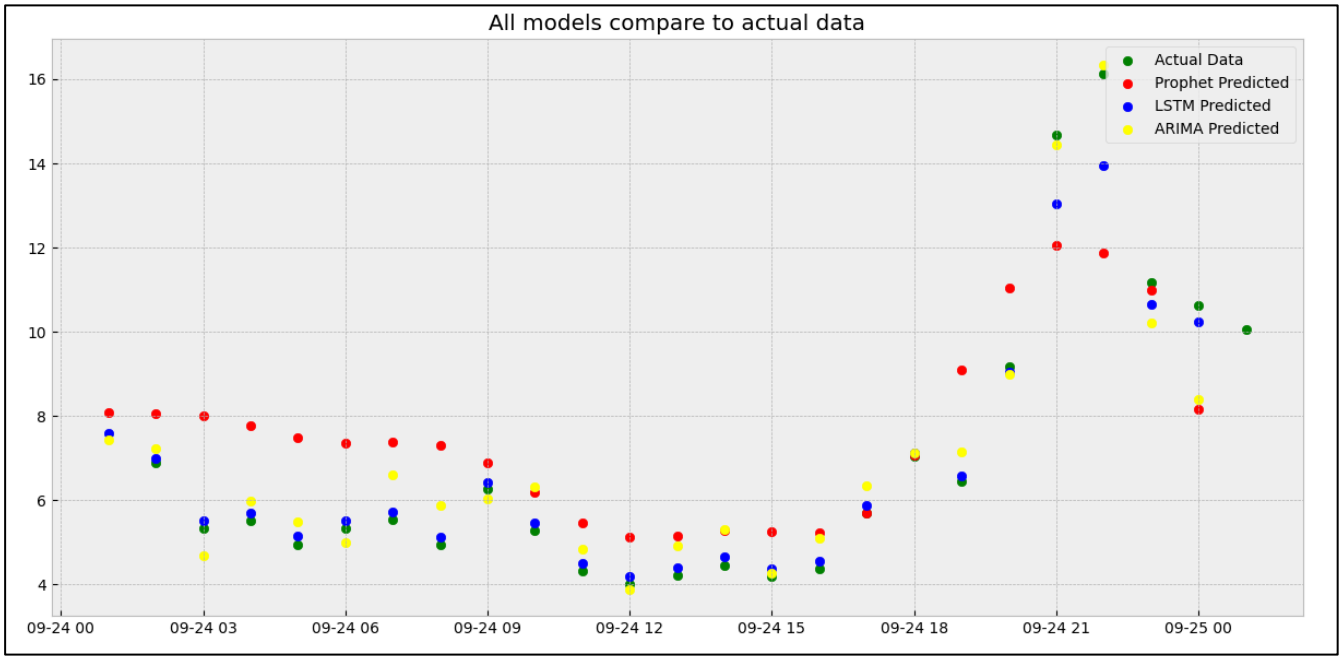


Рисунок 3.27 – Графік прогнозування усіх моделей у порівнянні з дійсними даними

Згідно з отриманими метриками, найкраще себе показала модель LSTM.

3.4 Висновки

У даному розділі проводилося прогнозування даних з використання трьох моделей машинного навчання, а саме: Prophet, ARIMA та LSTM. Перед побудовою моделей за необхідністю було виконано переведення даних до необхідного формату. Далі відбувалась побудова моделей. Після чого з використанням тренувального набору даних, відбувалось навчання моделі. Після навчання модель перевіряли на тестовому наборі даних.

Проведено візуалізацію отриманих результатів, та оцінку їх за метриками зазначеними вище. Найкращою виявилася модель LSTM, за метрикою r2_score. Для тренувального датасету – 0.99, а для тестового – 0.89.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних» 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 [25]

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
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
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
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

Продовження таблиці 4.1

	0	1	2	3	4
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 витрачати 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
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються 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	4	3	4
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	2	2	2
4. Ринкові переваги (технічні властивості)	2	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	2	2	2
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	2	2	2
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	32	34	34
Середньоарифметична сума балів $СБ_с$	33,3		

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 [26].

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 стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних» становить 33,3 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [26]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i , \tag{4.1}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *i-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$;

β_i – відносне значення *i-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами: для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ні}}{I_{ai}}, \tag{4.2}$$

де $I_{ні}$ та $I_{на}$ – чисельні значення конкретного *i*-го технічного показника якості відповідно для нової розробки та аналога;

для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ні}}. \tag{4.3}$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.4.

Таблиця 4.4 – Порівняння показників

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
1. Кількість прогнозованих показників інформаційної системи	одиниць	1	1	1	0,3
2. Об’єм бази даних інформації	мб	90	170	1,95	0,2
3. Точність прогнозу	%	0,5	0,8	1,6	0,15
4. Кількість аналізованих показників інформаційної системи	одиниць	6	8	1,8	0,25
5. Кількість станцій дослідження	одиниць	4	10	2,5	0,1

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1 \cdot 0,3 + 1,95 \cdot 0,2 + 1,6 \cdot 0,15 + 1,8 \cdot 0,25 + 2,5 \cdot 0,1 = 1,63.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,63 рази.

4.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних», 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 ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [26]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.4)$$

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=22 дні.

$$З_о = 17800{,}00 \cdot 48 / 22 = 38836{,}36 \ грн.$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на заробітну плату дослідників

Продовження таблиці 4.7

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
1.Керівник проекту з прогнозування даних	17800,00	809,09	48	38836,36
2. Інженер-програміст	17500,00	795,45	48	38181,82
3.Консультант (провідний фахівець служби екологічного нагляду)	12860,00	584,55	5	2922,73
4. Технік	8650,00	393,18	44	17300,00
Всього				97240,91

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i, \tag{4.5}$$

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:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \tag{4.6}$$

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

$К_і$ – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду [26];

$К_с$ – 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.

$Т_р$ – середнє число робочих днів в місяці, приблизно $Т_р$ = 22 дн;

$t_{зм}$ – тривалість зміни, год.

$С_1 = 6700{,}00 \cdot 1{,}10 \cdot 1{,}35 / (22 \cdot 8) = 56{,}53$ *грн.*

$З_{р1} = 56{,}53 \cdot 8{,}00 = 452{,}25$ *грн.*

Розраховані значення наведені в таблиці 4.6.

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
1.Встановлення офісних меблів, техніки, обладнання	8,00	2	1,10	56,53	452,25
2.Інсталяція програмного забезпечення	5,00	3	1,35	69,38	346,90
3.Компіляція програмних модулів	4,50	4	1,50	77,09	346,90
4.Відлагодження програмних модулів	6,00	5	1,70	87,37	524,20
5.Формування бази даних	22,00	3	1,35	69,38	1526,34
6.Налаштування серверного обладнання	7,00	4	1,50	77,09	539,62
Всього					3736,20

Додаткова заробітна плата дослідників та робітників

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

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.7)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$З_{дод}$ = (97240,91 + 3736,20) · 10 / 100% = 10097,71 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:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%}, \quad (4.8)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (97240,91 + 3736,20 + 10097,71) · 22 / 100% = 24436,46 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 стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних».

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

$$M = \sum_{j=1}^{n} H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^{n} B_j \cdot Ц_{вj}, \qquad (4.9)$$

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, ($K_j = 1,1 \ldots 1,15$);

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

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

$M_1 = 3 \cdot 167,00 \cdot 1,1 - 0 \cdot 0 = 551,10$ *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
Офісний папір Gemix A4 500	167,00	3	0	0	551,10
Папір для записів MIX 65 A5	89,00	5	0	0	489,50
Органайзер офісний	138,00	2	0	0	303,60
Набір офісний Base OFF	212,00	3	0	0	699,60
Картридж для принтера	969,00	1	0	0	1065,90
Диск оптичний TIMA CD	22,00	3	0	0	72,60

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Flesh-пам'ять GOODRAM 32 GB	132,00	1	0	0	145,20
Всього					3327,50

4.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних» відсутні.

4.3.5 Спецустаткування для наукових (експериментальних) робіт

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

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^{k} Ц_i \cdot C_{пр.i} \cdot K_i \ , \qquad (4.10)$$

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

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

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

k – кількість найменувань устаткування.

$B_{спец} = 35680,00 \cdot 1 \cdot 1,04 = 37107,20$ *грн.*

Отримані результати зведемо до таблиці 4.8.

Таблиця 4.8 – 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	37107,20
Сервер для зберігання БД (Сервер DELL R420 (4x3.5) LFF	1	28899,00	30054,96
Всього			67162,16

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:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.11)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

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

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

k – кількість найменувань програмних засобів.

$B_{прг} = 8459,00 \cdot 1 \cdot 1,01 = 8543,59$ *грн.*

Отримані результати зведемо до таблиці 4.9.

Таблиця 4.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Система керування реляційними базами даних MySQL	1	8459,00	8543,59
Система Kaggle	1	7599,00	7674,99
Середовище програмування Python та її бібліотеки	1	8520,00	8605,20
Емулятор серверної інтернет-платформи для моделювання поведінки інформаційного ресурсу	1	5800,00	5858,00
Доступ до мережі Internet (високошвидкісний) грн/місяць	2	239,00	482,78
Всього			31164,56

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:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \tag{4.12}$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

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

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл} = (25125{,}00 \cdot 3) / (2 \cdot 12) = 3140{,}63$ грн.

Проведені розрахунки зведемо до таблиці 4.10.

Таблиця 4.10 – 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	25125,00	2	3	3140,63
Обчислювально-графічна система програмної розробки	38599,00	2	3	4824,88
Робоче місце розробника програмного забезпечення	9580,00	5	3	479,00
Пристрій виводу інформації HP-5500	8765,00	4	3	547,81
Оргтехніка	8925,00	4	3	557,81
Приміщення лабораторії	255500,00	25	3	2555,00
ОС Windows 11	400,00	2	3	925,00
Прикладний пакет Microsoft Office 2021	6700,00	2	3	837,50
Всього				13867,63

4.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (4.13)$$

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

t_i – тривалість роботи обладнання на етапі дослідження, год;

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

$К_{вп}$ – коефіцієнт, що враховує використання потужності, $К_{вп} < 1$;

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

$В_е = 0,20 \cdot 360,0 \cdot 7,50 \cdot 0,95 / 0,97 = 540,00$ грн.

Проведені розрахунки зведемо до таблиці 4.11.

Таблиця 4.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер	0,20	360,0	540,00
Обчислювально-графічна система програмної розробки	0,40	360,0	1080,00
Робоче місце розробника програмного забезпечення	0,07	360,0	189,00
Пристрій виводу інформації HP-5500	0,22	4,0	6,60
Оргтехніка	0,45	2,0	6,75
Сервер – Компютер Expert PC Balance (I91F8H1S115E429)	0,22	360,0	594,00
Сервер для зберігання БД (Сервер DELL R420 (4x3.5) LFF	0,10	360,0	270,00
Всього			2686,35

4.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних» 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:

$$B_{св} = (З_о + З_р) \cdot \frac{H_{св}}{100\%}, \quad (4.14)$$

де $H_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$B_{св}$ = (97240,91 + 3736,20) · 20 / 100% = 20195,42 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:

$$B_{сп} = (З_о + З_р) \cdot \frac{H_{сп}}{100\%}, \quad (4.15)$$

де $Н_{сп}$ – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», приймемо $Н_{сп}$= 30%.

$B_{сп}$ = (97240,91 + 3736,20) · 30 / 100% = 30293,13 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:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.16)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів}$ = 50%.

$I_в$ = (97240,91 + 3736,20) · 50 / 100% = 50488,56 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:

$$В_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.17)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв}$ = 100%.

$В_{нзв}$ = (97240,91 + 3736,20) · 100 / 100% = 100977,11 грн.

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$В_{заг} = З_{о} + З_{р} + З_{дод} + З_{н} + М + К_{в} + В_{спец} + В_{прг} + А_{обл} + В_{е} + В_{св} + В_{сп} + І_{в} + В_{нзв}. \quad (4.18)$$

$В_{заг}$= 97240,91 +3736,20 +10097,71 +24436,46 +3327,50 +0,00 +67162,16 + 31164,56 + 13867,63 +2686,35 +20195,42 +30293,13 +50488,56 +100977,11 = 455673,70 грн.

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

$$ЗВ = \frac{В_{заг}}{\eta}, \quad (4.19)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,95.

ЗВ = 455673,70 / 0,95 = 479656,53 грн.

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

Робота складається з двох взаємопов’язаних частин «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці.

Частина 1. Формування датасету та системний аналіз даних» та «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних» які досліджувались окремими групами науковців. Сумарні витрати на проведення відповідних досліджень складають:

$$ЗВ = 384423{,}00 + 479656{,}00 = 864079{,}00 \text{ грн.}$$

Результати дослідження проведені за темою «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються, як відношення кількості років до збільшення кількості споживачів. 1-й рік – 6000, 2-1 рік – 8500, 3-й – 9200, 4-й – 6800.

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 33000 осіб;

$Ц_б$ – вартість послуги у році до впровадження інформаційної системи, приймемо 250,00 грн;

$\pm\Delta Ц_о$ – зміна вартості послуги від впровадження результатів, приймемо 57,50 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [26]:

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \quad (4.20)$$

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

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо ρ =45*%*;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році ϑ *=18%*;

Збільшення чистого прибутку 1-го року:

$\Delta П_1$ =(57,50·33000,00+307,50·6000)·0,83·0,45·(1-0,18/100%)=1146215,48 грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2$ =(57,50·33000,00+307,50·14500)·0,83·0,45·(1-0,18/100%)=1946728,69 грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3$ =(57,50·33000,00+307,50·23700)·0,83·0,45·(1-0,18/100%)=2813166,52 грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4$ =(57,50·33000,00+307,50·30500)·0,83·0,45·(1-0,18/100%)=3453577,09 грн.

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

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \quad (4.21)$$

де $\Delta П_i$ – 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;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ=0,12;

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

$$ПП = 1146215,48/(1+0,12)^1+1946728,69/(1+0,12)^2+2813166,52/(1+0,12)^3+$$
$$+3453577,09/(1+0,12)^4=1023406,67+1551920,19+2002356,36+2194810,67=$$
$$=6772493,90 \text{ грн.}$$

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

$$PV = k_{інв} \cdot ЗВ, \tag{4.22}$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=1;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 864079,00 грн.

$PV = k_{інв} \cdot ЗВ$= 1 · 864079,00 = 864079,00 грн.

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

$$E_{абс} = ПП - PV, \tag{4.23}$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 6772493,90 грн;

PV – теперішня вартість початкових інвестицій, 864079,00 грн.

$E_{абс} = ПП - PV$ = 6772493,90 – 864079,00 = 5908414,90 грн.

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

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.24)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 5908414,90 грн;

PV – теперішня вартість початкових інвестицій, 864079,00 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1$ = (1+5908414,90/864079,00)$^{1/4}$= 0,67.

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

$$\tau_{мін} = d + f, \quad (4.25)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,12;

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

$\tau_{мін}$= 0,12+0,3 = 0,42 < 0,67 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, 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 стану атмосферного повітря міста Вінниці» доцільно.

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

$$T_{ок} = \frac{1}{E_в}, \tag{4.26}$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,67 = 1,49 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 до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці» є високим, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

Також термін окупності становить 1,49 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 науково-дослідної роботи за темою «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці».

ВИСНОВКИ

В магістерській кваліфікаційній розроблено інформаційну технологію аналізу та прогнозування стану атмосферного повітря міста Вінниці.

Проведено огляд проблематики забруднення якості атмосферного повітря у світі та в Україні. Описано, які основні показники вимірюються на даний час в Україні та розглянуто чим вони є небезпечними для здоров'я людини. Розглянуто основні недоліки моніторингу атмосферного повітря в Україні та як їх можна виправити. Розглянуто стан моніторингу атмосферного повітря у місті Вінниця, які показники вимірюються тощо. Досліджено проблематику прогнозування якості атмосферного повітря з використанням штучного інтелекту та машинного навчання. Проведено огляд технологій, що використовувалися під час дослідження та написання магістерської кваліфікаційної роботи. Обрано мову програмування, та середовище розробки в якому здійснювалось дослідження. Обрано та описано бібліотеки, що були використані під час роботи над кваліфікаційною роботою.

Проведено детальний огляд моделей, що використовувались для прогнозування якості атмосферного повітря в даній роботі. Створено новий ноутбук та підключено зібрані дані громадського моніторингу якості повітря. Проведено детальний аналіз даних. Здійснено необхідні маніпуляції з даними аби привести їх до необхідного формату для подальшої роботи. Здійснено візуалізацію погодинних та середньо добових значень. Проведено аналіз аномальних значень та виконано їх візуалізацію. Підготовлено тренувальний та тестові данні для подальшого прогнозування.

Проводилося прогнозування даних з використання трьох моделей машинного навчання, а саме: Prophet, ARIMA та LSTM. Перед побудовою моделей а необхідністю було виконано переведення даних до необхідного формату. Далі відбувалась побудова моделей. Після чого з використанням тренувального набору даних, її тренувала. Після навчання модель перевіряли на тестовому наборі даних.

Проведено візуалізацію отриманих результатів, та оцінку їх за метриками зазначеними вище. Найкращою виявилася модель LSTM за метрикою r2_score. Для тренувального датасету – 0.99, а для тестового – 0.89.

Було проведено оцінку комерційного потенціалу інформаційної технології аналізу та прогнозування якості атмосферного повітря у місті Вінниці за даними громадського моніторингу, який є вище середнього що, свідчить про комерційну важливість проведення даних досліджень. Також термін окупності становить 1,49 р., 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 успішне створення інформаційної технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці.

Результати роботи опубліковано в матеріалах LIII Всеукраїнської науково-технічної підрозділів Вінницького національного технічного університету (2023-2024) [1].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пінчук В.П., Жуков С.О. Інформаційна технологія аналізу та прогнозування PM2.5 в атмосферному повітрі міста Вінниці на основі моделей Prophet, LSTM та ARIMA. *LIII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024). Вінниця, 2024.* URL: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/19681/16295 (дата звернення 08.12.2023)

2. Деякі питання здійснення державного моніторингу в галузі охорони атмосферного повітря. URL: http://search.ligazakon.ua/l_doc2.nsf/link1/KP190827.html (дата звернення: 01.12.2023).

3. Ramirez R. Only 13 countries and territories had 'healthy' air quality in 2022 | CNN. CNN. URL: https://edition.cnn.com/2023/03/14/world/air-pollution-report-2022-climate (date of access: 07.10.2023).

4. Explore the Data | State of Global Air. Home | State of Global Air. URL: https://www.stateofglobalair.org/data/#/air/plot (date of access: 01.12.2023).

5. Моніторинг довкілля : підручник В. М. Боголюбов, М. О. Клименко, В. Б. Мокін [та ін.] ; за ред. В. М. Боголюбова. і Т. А. Сафранова. Херсон: 2011. 530 с.

6. ОСНОВНІ ЗАБРУДНЮВАЧІ АТМОСФЕРНОГО ПОВІТРЯ – ТОВ «Автоекоприлад». URL: https://eco.aep.kiev.ua/novini/osnovni-zabrudnyuvachi-atmosfernogo-povitrya-harakteritstiki-vpliv-na-organizm-lyudini/ (дата звернення: 01.12.2023).

7. Вінницький обласний центр контролю та профілактики хвороб МОЗ України. URL: http://cgz.vn.ua/problematika-gromadskogo-zdorovya/problematika-gromadskogo-zdorovya_455.html (дата звернення: 01.12.2023).

8. Eco-City. Eco-City Громадський моніторинг стану якості повітря. URL: https://eco-city.org.ua/ (дата звернення: 26.06.2023).

9. Мокін В. Б., Дзюняк Д. Ю., Бондалєтов К. О., і Олійник В. В., «МЕТОД І ТЕХНОЛОГІЯ МОНІТОРИНГУ СТАНУ АТМОСФЕРНОГО ПОВІТРЯ ЗА ДОПОМОГОЮ УНІВЕРСАЛЬНОЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ МОБІЛЬНИХ ПРИСТРОЇВ», *НаукПраці ВНТУ*, вип. 4, Січ 2016.

10. PM2.5 forecasting for an urban area based on deep learning and decomposition method N. Zaini et al. *Scientific Reports*. 2022. Vol. 12, no. 1. URL: https://doi.org/10.1038/s41598-022-21769-1 (date of access: 01.12.2023).

11. S. Subramaniam et al. Artificial intelligence technologies for forecasting air pollution and human health: a narrative review. *Sustainability*. 2022. Vol. 14, no. 16. P. 9951. URL: https://doi.org/10.3390/su14169951 (date of access: 15.10.2023).

12. Middya A. I., Roy S. Pollutant specific optimal deep learning and statistical model building for air quality forecasting. *Environmental pollution*. 2022. Vol. 301. P. 118972. URL: https://doi.org/10.1016/j.envpol.2022.118972 (date of access: 15.10.2023).

13. A. Al-Dahoud, M. Fezari, I. Jannoud, T. AL-Rawashdeh. Monitoring. Metropolitan City Air-quality Using Wireless Sensor Nodes based on ARDUINO and XBEE. URL: http://www.inase.org/library/2015/vienna/bypaper/CSSCC/CSSCC (дата звернення: 01.12.2023).

14. Deep neural networks for spatiotemporal PM2.5 forecasts based on atmospheric chemical transport model output and monitoring data / P.-Y. Kow et al. *Environmental Pollution*. 2022. P. 119348. URL: https://doi.org/10.1016/j.envpol.2022.119348 (date of access: 07.12.2023).

15. AI-based air quality PM2.5 forecasting models for developing countries: A case study of Ho Chi Minh City, Vietnam R. Rakholia et al. *Urban Climate*. 2022. Vol. 46. P. 101315. URL: https://doi.org/10.1016/j.uclim.2022.101315 (date of access: 07.12.2023).

16. NumPy. NumPy. URL: https://numpy.org/ (date of access: 01.12.2023).

17. pandas documentation – pandas 2.1.3 documentation. pandas – Python Data Analysis Library. URL: https://pandas.pydata.org/docs/index.html (date of access: 01.12.2023).

18. Quick Start. Prophet. URL: https://facebook.github.io/prophet/docs/quick_start.html (date of access: 01.12.2023).

19. scikit-learn: machine learning in Python – scikit-learn 1.3.2 documentation. scikit-learn: machine learning in Python – scikit-learn 0.16.1 documentation. URL: https://scikit-learn.org/stable/ (date of access: 01.12.2023)

20. Keras documentation: Developer guides. Keras: Deep Learning for humans. URL: https://keras.io/guides/ (date of access: 01.12.2023).

21. statsmodels 0.14.0. URL: https://www.statsmodels.org/stable/index.html (date of access: 01.12.2023).

22. Lewinson E. Facebook's Prophet + Deep Learning = NeuralProphet. Medium. URL: https://towardsdatascience.com/facebooks-prophet-deep-learning-neuralprophet-76796aed1d86 (date of access: 01.12.2023).

23. Deep Learning | Introduction to Long Short Term Memory – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/ (date of access: 01.12.2023).

24. Hayes A. Autoregressive Integrated Moving Average (ARIMA) Prediction Model. Investopedia. URL: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp (date of access: 07.12.2023).

25. Козловський В. О., Лесько О. Й., Кавецький В. В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт : Вінниця: ВНТУ, 2021. 42 с.

26. Кавецький В. В., Козловський В. О., Причепа 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 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на комплексну магістерську кваліфікаційну роботу
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ ЯКОСТІ АТМОСФЕРНОГО ПОВІТРЯ У МІСТІ ВІННИЦІ. ЧАСТИНА 2. ПРОГНОЗУВАННЯ ДАНИХ»
08-34.МКР.002.02.000.ТЗ

Керівник: к.т.н., доц. каф. САІТ
_____ Сергій ЖУКОВ
«___» _____ 2023 р.

Розробив: студент гр. 2ІСТ-22м
_____ Владислав ПІНЧУК
«___» _____ 2023 р.

Вінниця 2023

1. Підстава для проведення робіт

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

2. Джерела розробки:

– В. Б. Мокін, Д. Ю. Дзюняк, К. О. Бондалєтов, і В. В. Олійник, «МЕТОД І ТЕХНОЛОГІЯ МОНІТОРИНГУ СТАНУ АТМОСФЕРНОГО ПОВІТРЯ ЗА ДОПОМОГОЮ УНІВЕРСАЛЬНОЇ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ МОБІЛЬНИХ ПРИСТРОЇВ», *НаукПраці ВНТУ*, вип. 4, Січ 2016.

– PM2.5 forecasting for an urban area based on deep learning and decomposition method N. Zaini et al. *Scientific Reports*. 2022. Vol. 12, no. 1. URL: https://doi.org/10.1038/s41598-022-21769-1 (date of access: 01.12.2023).

3. Мета і призначення роботи:

Поліпшення точності побудови картограм реального чи прогнозованого стану забруднення атмосферного повітря у м. Вінниці за даними громадського моніторингу шляхом розроблення методики вибору місць для нових пунктів спостереження.

4.Вихідні дані для проведення робіт:

Kaggle датасет, на основі даних громадського моніторингу міста Вінниці: Air quality in Vinnytsia 2023 (September) (https://www.kaggle.com/datasets/pirogovskiy/air-quality-in-vinnytsia-2023-september.

5.Методи дослідження:

– Розвідувальний аналіз даних;
– Моделі машинного навчання: Prophet, LSTM, ARIMA;
– Прогнозування даних;

6. Етапи роботи і терміни їх виконання:

1. Аналіз предметної області ..... ____ – ____
2. Опис моделей, які будуть використовуватись для передбачення даних. ..... ____ – ____
3. Вибір мови програмування, бібліотек та середовища розробки ……...____ – ____
4.
5. Розроблення інформаційної технології та її застосування на реальних даних. ..... ____ – ____
6. Прогнозування даних ..... ____ – ____
7. Економічна частина..... ____ – ____

8. Оформлення пояснювальної записки. ..... ____ – ____

7. Очікувані результати та порядок реалізації:

Отримати інформаційну технологію прогнозування та аналізу якості повітря у місті Вінниця.

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 2ІСТ-22м _____ Владислав ПІНЧУК

Додаток Б.

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: «Інформаційна технологія аналізу та прогнозування стану атмосферного повітря міста Вінниці. Частина 2. Прогнозування даних»
Тип роботи: магістерська кваліфікаційна робота
Підрозділ: кафедра САІТ

Показники звіту подібності Unicheck

Оригінальність 91,67% Схожість 8,33%

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 _____ Сергій ЖУКОВ
(підпис)

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

Автор роботи _____ Владислав ПІНЧУК
(підпис)

Керівник роботи _____ Сергій ЖУКОВ

Додаток В.

Лістинг програми

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from matplotlib import pyplot as plt
plt.style.use('bmh')
from datetime import datetime
from scipy.stats import zscore
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error, mean_absolute_percentage_error,
mean_absolute_error
from prophet import Prophet
from prophet.plot import add_changepoints_to_plot
from statsmodels.tsa.arima.model import ARIMA
import statsmodels.api as sm
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import *
from keras.callbacks import ModelCheckpoint
from keras.optimizers import Adam
from keras.metrics import RootMeanSquaredError


%matplotlib inline

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
forecasting_period = 24
measuring_threshold_per_day = 15
predicted_data = pd.DataFrame(columns = [
    'Time',
    'Actual',
    'Prophet',
    'LSTM',
    'ARIMA',
])

models_metrics = pd.DataFrame(columns = ['name', 'r2_score_train', 'r2_score_te
st', 'rmse_train', 'rmse_test', 'mae_train', 'mae_test'])
data = pd.read_csv("/kaggle/input/air-quality-in-vinnytsia-2023-september/Air q
uality in September 2023.csv")
data
```

```
data = data = data[['stations_id', 'stations_time', 'stations_params_name', 'st
ations_params_value']]
data
def parse_date(date):
    parsed_date = datetime.strptime(date, '%Y-%m-%dT%H:%M:%SZ')
    return parsed_date.strftime('%Y-%m-%d %H:%M:%S')
data.loc[(data['stations_id'] == 246) | (data['stations_id'] == 256)
                        | (data['stations_id'] == 271) | (data['stations_id']
== 281)
                        | (data['stations_id'] == 274) |(data['stations_id'] =
= 315)
                        | (data['stations_id'] == 90), 'stations_time'] = data
.loc[(data['stations_id'] == 246) | (data['stations_id'] == 256)
                        | (data['stations_id'] == 271) | (data['stations_id']
== 281)
                        | (data['stations_id'] == 274) |(data['stations_id'] =
= 315)
                        | (data['stations_id'] == 90), 'stations_time'].apply(
parse_date)
pm25 = data[data['stations_params_name'] == 'PM2.5']
pm25 = pm25[['stations_time', 'stations_params_value']].reset_index(drop=True)
pm25 = pm25.rename(columns={'stations_time': 'ds', 'stations_params_value': 'y'
})
pm25['ds'] = pd.to_datetime(pm25['ds'], format = '%Y-%m-%d %H:%M:%S', errors =
'ignore')
pm25
pm25['y'] = pm25['y'].astype(float)
plt.figure(figsize=(12, 6))
plt.plot(pm25['ds'],pm25['y'])
plt.title('PM2,5')
plt.xlabel('Time')
plt.ylabel('Value')
plt.show()
pm25_avg_1day = pm25.copy()
pm25_avg_1day['ds'] = pd.to_datetime(pm25_avg_1day['ds'])
pm25_avg_1day.set_index('ds', inplace=True)
pm25_avg_1day = pm25_avg_1day.resample('1D').mean()
pm25_avg_1day = pm25_avg_1day.dropna().reset_index(drop=False)
pm25_avg_1day
pm25_avg_1day['y'] = pm25_avg_1day['y'].astype(float)
plt.figure(figsize=(12, 6))
plt.plot(pm25_avg_1day['ds'],pm25_avg_1day['y'])
plt.title('PM2,5')
plt.xlabel('Time')
plt.ylabel('Value')
plt.show()
avg_per_month = pm25_avg_1day['y'].mean()
plt.figure(figsize=(12, 6))
```

```
plt.plot(pm25_avg_1day['ds'], pm25_avg_1day['y'], label='Actual PM2.5', color='
green')
plt.axhline(y=measuring_threshold_per_day, color='red', label='Thershold PM2.5'
)
plt.axhline(y=avg_per_month, color='blue', label='Average PM2.5')
plt.xlabel('Time')
plt.ylabel('PM2.5 Values')
plt.title('Dust Pollution (PM2.5)')
plt.legend()
plt.show()
pm25_avg_1hour = pm25.copy()
pm25_avg_1hour['ds'] = pd.to_datetime(pm25_avg_1hour['ds'])
pm25_avg_1hour.set_index('ds', inplace=True)
pm25_avg_1hour = pm25_avg_1hour.resample('60min').mean()
pm25_avg_1hour = pm25_avg_1hour.dropna().reset_index(drop=False)
pm25_avg_1hour
pm25_avg_1hour['y'] = pm25_avg_1hour['y'].astype(float)
avg_per_hour = pm25_avg_1hour['y'].mean()
plt.figure(figsize=(12, 6))
plt.plot(pm25_avg_1hour['ds'],pm25_avg_1hour['y'], label='Actual PM2.5')
plt.axhline(y=avg_per_hour, color='red', label='Average PM2.5')
plt.title('PM2,5')
plt.xlabel('Time')
plt.ylabel('Value')
plt.show()
mean = pm25_avg_1hour['y'].mean()
std = pm25_avg_1hour['y'].std()
print (mean)
print (std)
plt.figure(figsize=(12, 6))
plt.plot(pm25_avg_1hour['ds'], pm25_avg_1hour['y'], label='Actual PM2.5', color
='gray')
plt.axhline(y=avg_per_hour, color='green', label='Average PM2.5')
plt.scatter(anomalous['ds'], anomalous['y'], color='red', label='Anomalies')
plt.xlabel('Time')
plt.ylabel('PM2.5 Values')
plt.title('Dust Pollution (PM2.5)')
plt.legend()
plt.show()
pm25_test_data = pm25_avg_1hour[-forecasting_period:]
pm25_train_data = pm25_avg_1hour[:-forecasting_period]
predicted_data['Time'] = pm25_test_data['ds']
predicted_data['Actual'] = pm25_test_data['y']
```

In [27]:

```
model = Prophet(daily_seasonality=True, weekly_seasonality=False, yearly_season
ality=False,
                n_changepoints = 30,
                changepoint_range=1, changepoint_prior_scale = 10,
```

```
                holidays=a, seasonality_mode = 'multiplicative')
model.add_seasonality(name='hour', period=1/24, fourier_order=1,
                      mode = 'additive', prior_scale = 10)

model.add_seasonality(name='week', period=7, fourier_order=3,
                      mode = 'additive', prior_scale = 3)
model.fit(pm25_train_data)
future = model.make_future_dataframe(periods=forecasting_period, freq='H')
forecast = model.predict(future)
forecasted_values = forecast[['ds', 'yhat']]
fig = model.plot(forecast, xlabel='Time', ylabel='PM2.5')
a = add_changepoints_to_plot(fig.gca(), model, forecast)
plt.show()
plt.figure(figsize=(12, 6))
plt.plot(pm25_train_data['ds'], pm25_train_data['y'], label='Actual PM2.5', col
or='green')
plt.plot(b['ds'], b['yhat'], label='Predicted PM2.5', color='red')
plt.title('Train Data PM2.5')
plt.legend()
plt.show()
prophet_predicted_values = forecasted_values[-forecasting_period:]
prophet_predicted_values.head()
plt.figure(figsize=(12, 6))
plt.plot(pm25_test_data['ds'], pm25_test_data['y'], label='Actual PM2.5', color
='green')
plt.plot(prophet_predicted_values['ds'], prophet_predicted_values['yhat'], labe
l='Predicted PM2.5', color='red')
plt.title('Test Data PM2.5')
plt.legend()
plt.show()
models_metrics.loc[0, 'name'] = 'Prophet'

models_metrics.loc[0,'r2_score_train'] = round(r2_score(pm25_train_data['y'], b
['yhat']),2)
models_metrics.loc[0,'r2_score_test'] = round(r2_score(pm25_test_data['y'], pro
phet_predicted_values['yhat']),2)

models_metrics.loc[0,'rmse_train'] =  round(np.sqrt(mean_squared_error(pm25_tra
in_data['y'], b['yhat'])),2)
models_metrics.loc[0,'rmse_test'] = round(np.sqrt(mean_squared_error(pm25_test_
data['y'], prophet_predicted_values['yhat'])),2)

models_metrics.loc[0,'mae_train'] = round(mean_absolute_error(pm25_train_data['
y'], b['yhat']),2)
models_metrics.loc[0,'mae_test'] = round(mean_absolute_error(pm25_test_data['y'
], prophet_predicted_values['yhat']),2)

models_metrics
```

```
lstm_data = pm25_avg_1hour.copy()

values = lstm_data['y'].values.reshape(-1, 1)

scaler = MinMaxScaler(feature_range=(0, 1))
scaled_values = scaler.fit_transform(values)

train_size =  int(len(scaled_values) - forecasting_period)
train_data = scaled_values[:train_size, :]
test_data = scaled_values[train_size:, :]

def create_sequences(data, sequence_length):
    X = []
    y = []
    for i in range(len(data) - sequence_length):
        a = data[i:(i+sequence_length), 0]
        X.append(a)
        y.append(data[i + sequence_length, 0])
    return np.array(X), np.array(y)

sequence_length = 1

X_train, y_train = create_sequences(train_data, sequence_length)
X_test, y_test = create_sequences(test_data, sequence_length)
lstm_model = Sequential()

lstm_model.add(LSTM(128, 'relu', return_sequences=True, input_shape=(sequence_l
ength, 1)))
lstm_model.add(LSTM(64))
lstm_model.add(Dense(32))
lstm_model.add(Dense(1))

cp = ModelCheckpoint('modem', save_best_only=True)

lstm_model.compile(optimizer='adam', loss='mean_squared_error', metrics=[RootMe
anSquaredError()])

lstm_model.fit(X_train, y_train, epochs=200, batch_size=32, callbacks=[cp], ver
bose=False)

predictions_train = lstm_model.predict(X_train)
predictions_test = lstm_model.predict(X_test)
lstm_model.summary()
predictions_train = scaler.inverse_transform(predictions_train)
train_data = scaler.inverse_transform(train_data)

predictions_test = scaler.inverse_transform(predictions_test)
test_data = scaler.inverse_transform(test_data)
```

```
train_dates = lstm_data['ds'].values[:train_size-sequence_length]
test_dates = lstm_data['ds'].values[train_size+sequence_length:]
trainPredictPlot = np.empty_like(scaled_values)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[sequence_length:len(predictions_train)+sequence_length, :] = p
redictions_train

testPredictPlot = np.empty_like(scaled_values)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(predictions_train)+(sequence_length*2):len(scaled_values),
:] = predictions_test
plt.figure(figsize=(12, 6))

plt.plot(scaler.inverse_transform(scaled_values), label='Actual PM2.5', color='
green')
plt.plot(trainPredictPlot, label='Predicted Train PM2.5', color='red')
plt.plot(testPredictPlot, label='Predicted Test PM2.5', color='yellow')
plt.legend()
plt.show()
plt.figure(figsize=(12, 6))
plt.plot(train_dates, lstm_data['y'].values[:train_size-sequence_length], label
='Actual (Training)')
plt.plot(train_dates, predictions_train, label='Predicted (Training)')
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('LSTM Model: Actual vs Predicted')
plt.legend()
plt.show()
plt.figure(figsize=(12, 6))
plt.plot(test_dates, lstm_data['y'].values[train_size+sequence_length:], label=
'Actual (Test)')
plt.plot(test_dates, predictions_test, label='Predicted (Test)')
plt.xlabel('Date')
plt.ylabel('Value')
plt.title('LSTM Model: Actual vs Predicted')
plt.legend()
plt.show()
arima_data = pm25_avg_1hour.copy()
arima_test_data = pd.Series(arima_data['y'][-forecasting_period:].values, index
=range(forecasting_period))
arima_train_data = arima_data['y'][:-forecasting_period]


model = ARIMA(arima_train_data, order = (5, 1, 3))
model_fit = model.fit()
ARIMA_train_predictions = model_fit.predict()
```

```
history = [x for x in arima_train_data]
ARIMA_predicted = list()

for t in range(len(arima_test_data)):

    model = ARIMA(history, order = (5, 1, 3))
    model_fit = model.fit()
    output = model_fit.forecast()
    yhat = output[0]
    ARIMA_predicted.append(yhat)
    obs = arima_test_data[t]
    history.append(obs)
plt.figure(figsize=(12, 6))
plt.plot(arima_data[-forecasting_period:]['ds'], arima_test_data.values[:train_
size-sequence_length], label='Actual (Test)')
plt.plot(arima_data[-forecasting_period:]['ds'], ARIMA_predicted, label='Predic
ted (Test)')
plt.title('ARIMA Model: Actual vs Predicted')
plt.legend()
plt.tight_layout()
plt.show()
models_metrics.loc[2,'name'] = 'ARIMA'

models_metrics.loc[2,'r2_score_train'] = round(r2_score(arima_train_data, ARIMA
_train_predictions),2)
models_metrics.loc[2,'r2_score_test'] = round(r2_score(arima_test_data, ARIMA_p
redicted),2)

models_metrics.loc[2,'rmse_train'] =  round(np.sqrt(mean_squared_error(arima_tr
ain_data, ARIMA_train_predictions)),2)
models_metrics.loc[2,'rmse_test'] = round(np.sqrt(mean_squared_error(arima_test
_data, ARIMA_predicted)),2)

models_metrics.loc[2,'mae_train'] = round(mean_absolute_error(arima_train_data,
ARIMA_train_predictions),2)
models_metrics.loc[2,'mae_test'] = round(mean_absolute_error(arima_test_data, A
RIMA_predicted),2)

models_metrics
plt.figure(figsize=(12, 6))
plt.plot(predicted_data['Time'], predicted_data['Actual'], label='Actua Data',
color='green')
plt.plot(predicted_data['Time'], predicted_data['Prophet'], label='Prophet Pred
icted', color='red')
plt.plot(predicted_data['Time'], predicted_data['LSTM'], label='LSTM Predicted'
, color='blue')
```

```
plt.plot(predicted_data['Time'], predicted_data['ARIMA'], label='ARIMA Predicte
d', color='yellow')
plt.title('All models compare to actual data')
plt.legend()
plt.tight_layout()
plt.show()
```

Додаток Г.

Ілюстративна частина

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ СТАНУ АТМОСФЕРНОГО ПОВІТРЯ МІСТА ВІННИЦІ. ЧАСТИНА 2. ПРОГНОЗУВАННЯ ДАНИХ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2023 р.

```
data
```

	stations_id	stations_time	stations_params_name	stations_params_value
0	650	2023-09-18 01:00:53	PM10	4.663
1	650	2023-09-18 01:00:53	PM2.5	4.164
2	650	2023-09-18 01:00:53	VOC (H_2CO)	10.0
3	650	2023-09-18 01:00:53	Temperature	15.413
4	650	2023-09-18 01:00:53	Humidity	62.951
...	...	...	...	...
148484	90	2023-09-17 21:35:00	PM2.5	8.6295
148485	90	2023-09-17 21:35:00	PM100	14.80825
148486	90	2023-09-17 21:35:00	Temperature	17.80325
148487	90	2023-09-17 21:35:00	Humidity	56.13625
148488	90	2023-09-17 21:35:00	Pressure	99082.50175

Рисунок Г.1 – Фрагмент датасету з даними

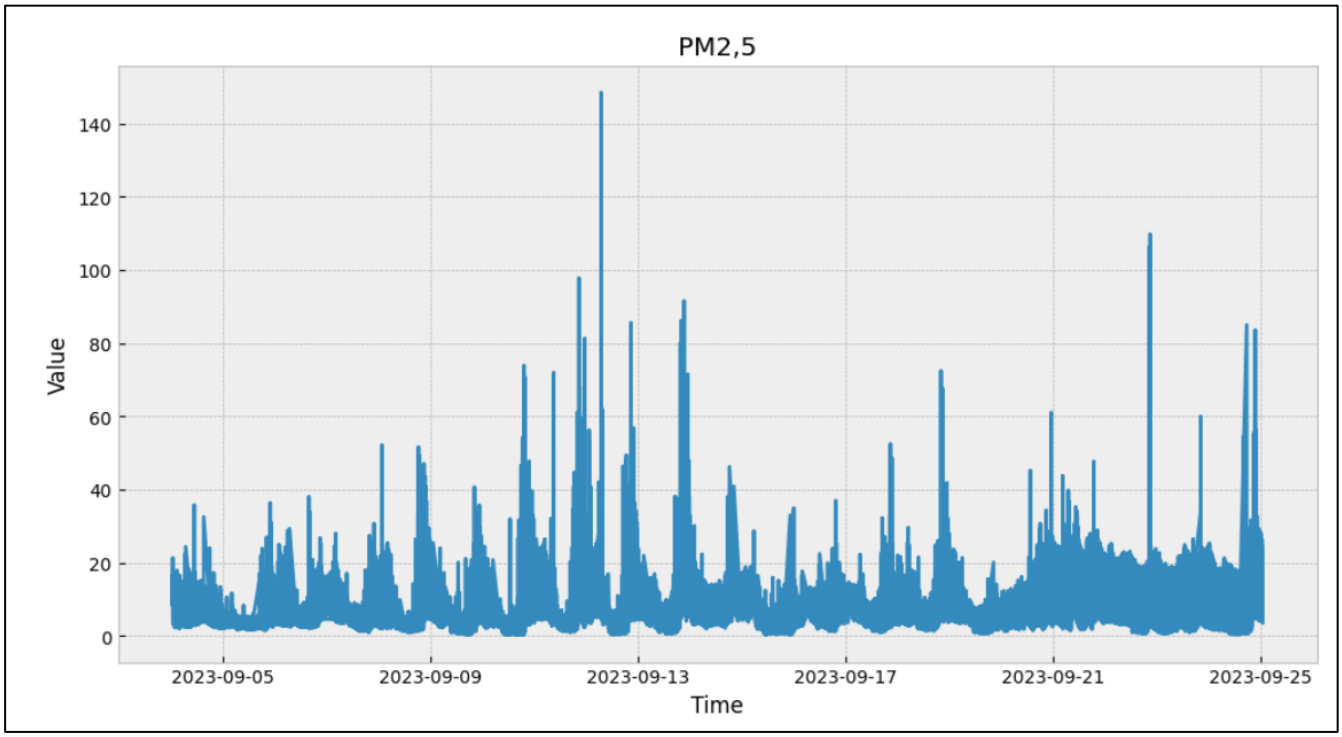


Рисунок Г.2 – Графік динаміки значень частинок PM2.5 з усіх станцій

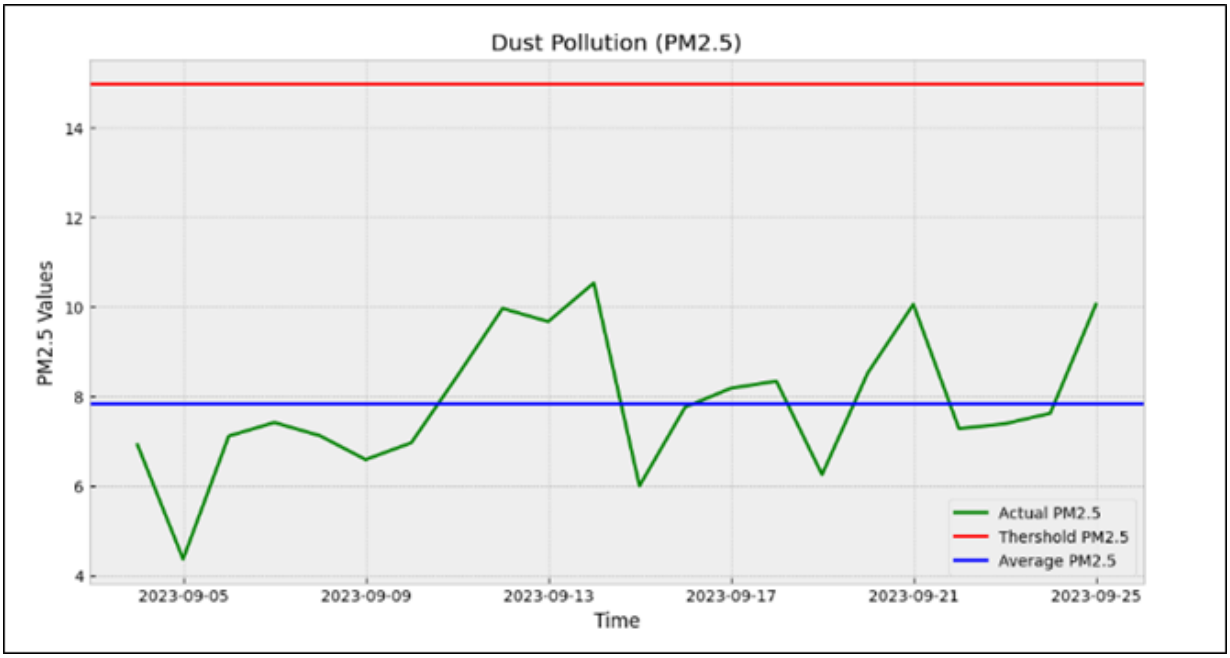


Рисунок Г.3 – Графік динаміки середньодобових значень частинок PM2.5

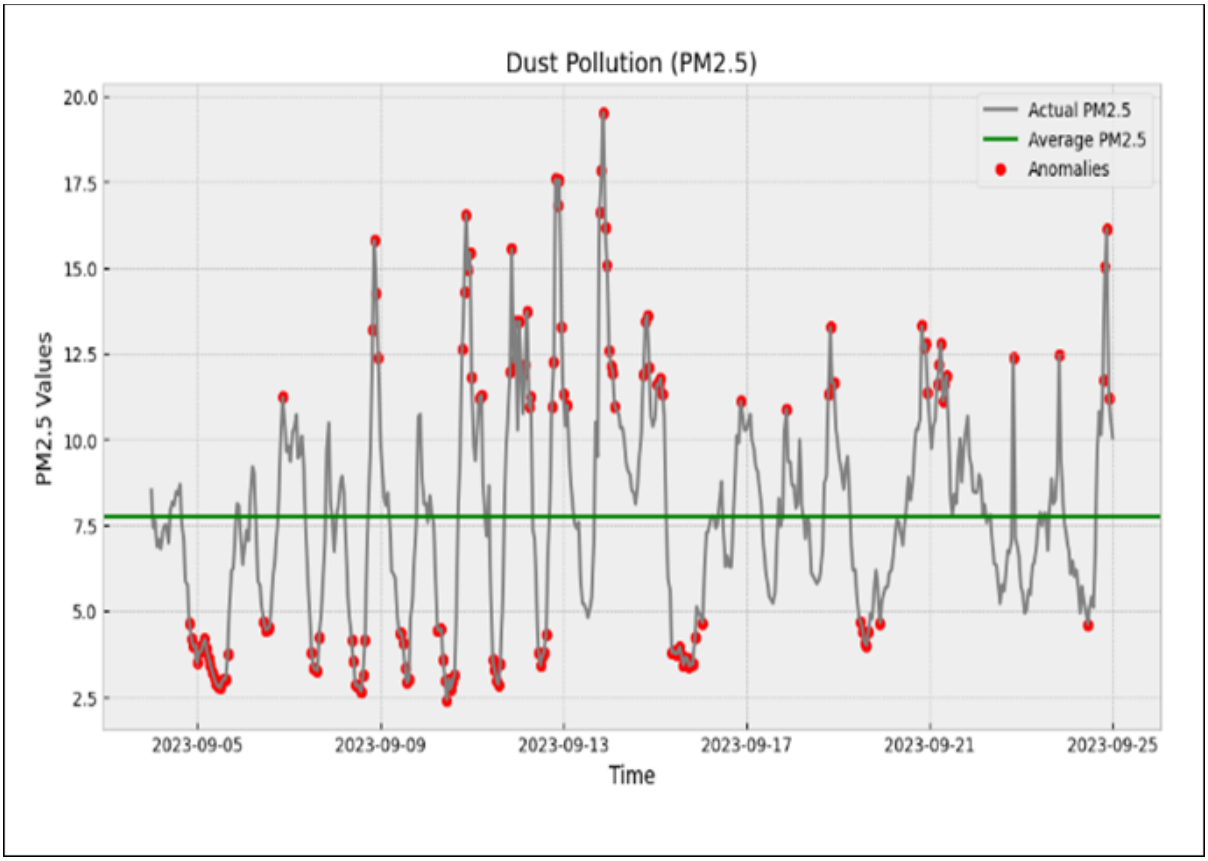


Рисунок Г.4 – Графік аномалій частинок PM2.5 в середніх погодинних значеннях

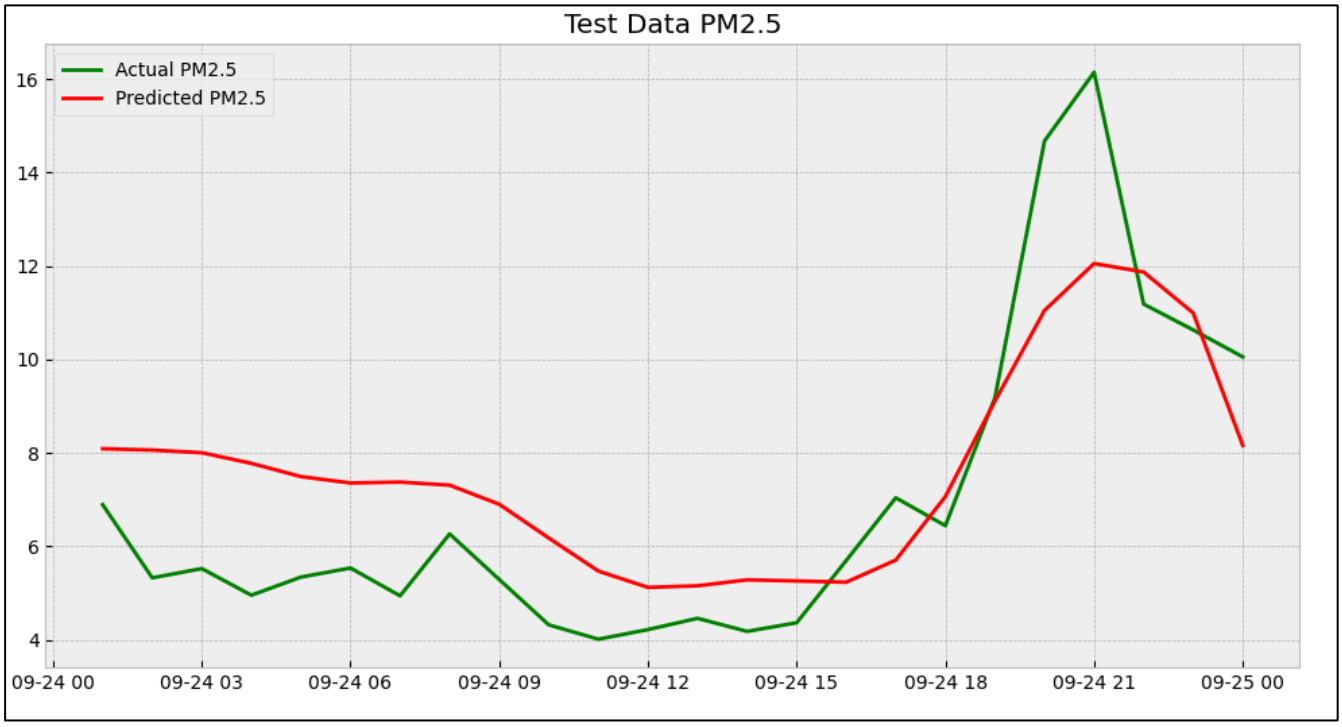


Рисунок Г.5 – Графік прогнозування моделі Prophet

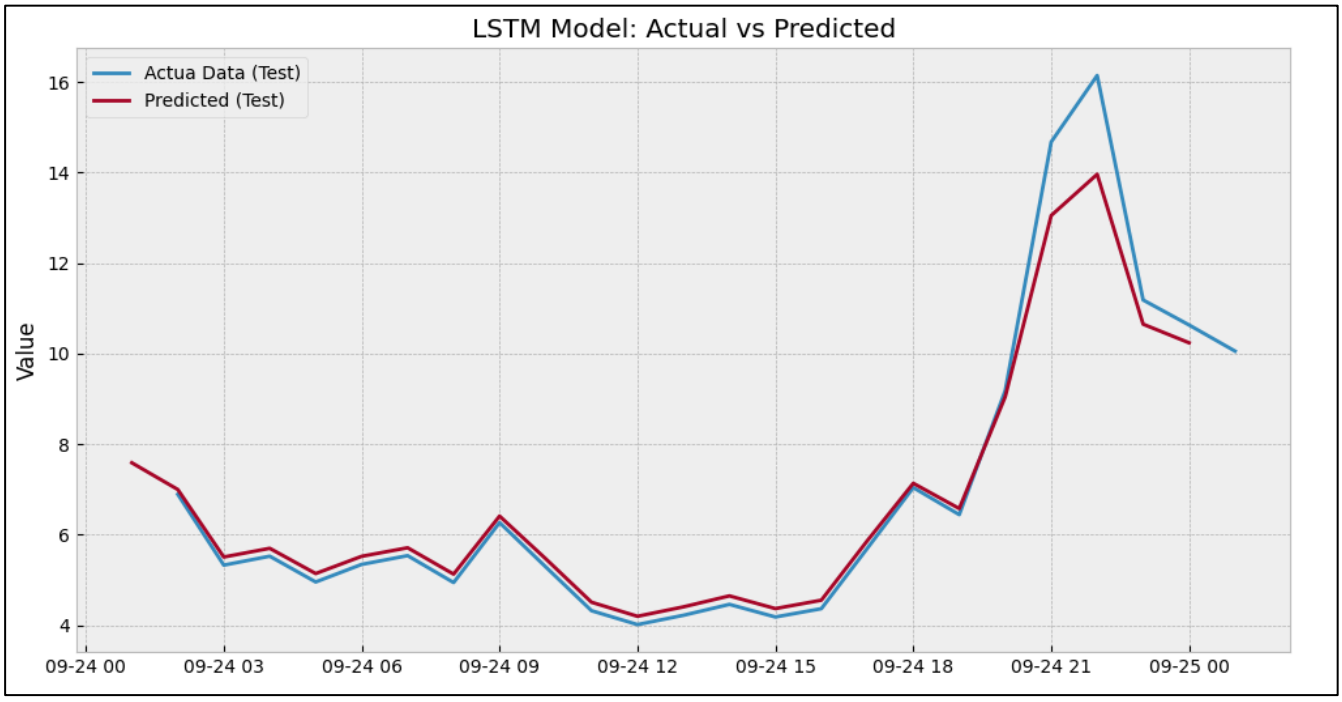


Рисунок Г.6 – Графік прогнозування моделі LSTM

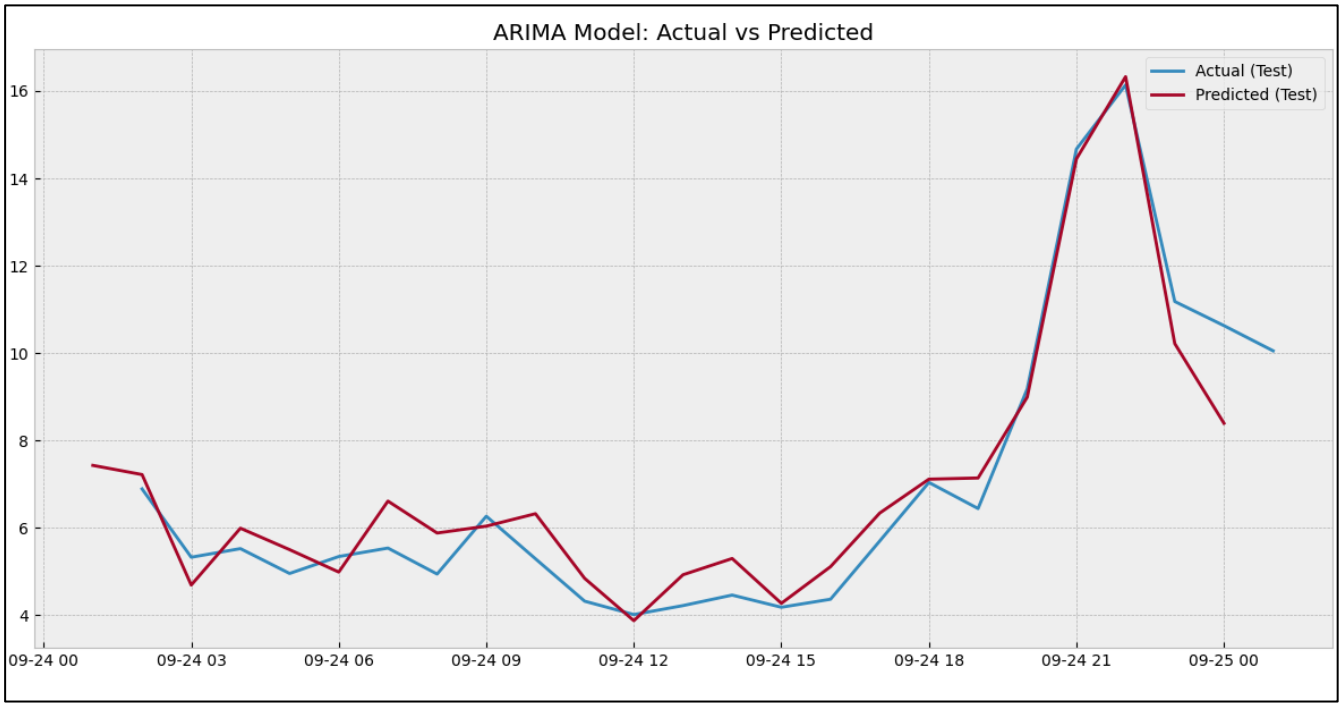


Рисунок Г.7 – Графік прогнозування моделі ARIMA

	name	r2_score_train	r2_score_test	rmse_train	rmse_test	mae_train	mae_test
0	Prophet	0.91	0.73	0.93	1.69	0.71	1.47
1	LSTM	0.99	0.89	0.426	1.502	0.295	1.015
2	ARIMA	0.85	0.82	1.17	1.38	0.78	0.91

Рисунок Г.8 – Результати роботи моделей