

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ВІДНОВЛЕННЯ ВІДСУТНІХ ДАНИХ МОНІТОРИНГУ ЯКОСТІ ПОВІТРЯ НА
ОСНОВІ ДАНИХ КОНЦЕНТРАЦІЇ PM2.5
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS TO
RECOVER MISSING AIR QUALITY MONITORING DATA BASED ON PM2.5
CONCENTRATION DATA

Виконав(ла): студент(ка) 4 курсу, групи КНТ-221
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

КОЛОСОВ К.Є.

(ПРІЗВИЩЕ та ініціали)

Керівник ПОДКОВАЛІХІНА О.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент ГОЛУБ Т.В.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 122 Комп'ютерні науки
(код і найменування)
 Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КОЛОСОВА Кирила Євгеновича
(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів відновлення відсутніх даних моніторингу якості повітря на основі даних концентрації PM2.5 Research and Software Implementation of Methods to Recover Missing Air Quality Monitoring Data Based on PM2.5 Concentration Data
 керівник проєкту (роботи) к.ф.-м.н., доцент, ПОДКОВАЛІХІНА Олена Олександрівна,

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 02 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ПОДКОВАЛІХІНА О.О., доцент		
Нормоконтроль	БСЛОВА А.В., асистент		

7. Дата видачі завдання « 14 » квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки.		
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження	5 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	6 тиждень	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Кирило КОЛОСОВ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Олена ПОДКОВАЛІХІНА
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 81 с., 3 табл., 39 рис., 2 дод., 17 джерел.

МОНІТОРИНГ ЯКОСТІ ПОВІТРЯ, PM2.5, ЧАСОВІ РЯДИ, МЕТОДИ ВІДНОВЛЕННЯ ДАНИХ, ЕКОЛОГІЧНИЙ МОНІТОРИНГ, ЕКОЛОГІЧНИЙ ПРОГНОЗ.

Об'єкт дослідження – процес обробки та аналізу даних моніторингу якості повітря.

Предмет дослідження – методи та алгоритми машинного навчання та статистичного аналізу для відновлення пропущених даних у часових рядах концентрації PM2.5.

Мета роботи - дослідження та порівняльний аналіз методів відновлення пропущених даних у часових рядах концентрації PM2.5 з метою підвищення точності відновлення.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, поліноміальна регресія, інтерполяція сплайнами (PCHIP та згладжуючий сплайн), ARIMA, SARIMA, Random Forest, мова програмування Python, бібліотеки pandas, numpy, matplotlib, scikit-learn, statsmodels, scipy, tkinter, персональний комп'ютер з процесором Intel Core i7-12700F під управлінням операційної системи Microsoft Windows 10.

Результати. У межах дослідження було реалізовано й порівняно шість методів відновлення пропущених значень у часових рядах концентрацій PM2.5 на основі реальних даних моніторингу повітря у Запоріжжі, Дніпрі та Львові: поліноміальна регресія третього ступеня, інтерполяція сплайнами (PCHIP і згладжуючий сплайн), ARIMA, SARIMA, Random Forest. Якість відновлення оцінювалася за метрикою RMSE.

На основі проведеного аналізу створено програмний застосунок для Windows із графічним інтерфейсом. Він дозволяє завантажувати та обробляти часові ряди, обирати метод відновлення, візуалізувати результати (графіки, таблиці) та експортувати їх для подальшого аналізу. Реалізацію виконано на Python із використанням бібліотек `scikit-learn`, `statsmodels`, `tkinter` та `Matplotlib`.

Висновки. Було проведено систематичне порівняння ефективності різних методів відновлення. Для цього в даних штучно моделювали пропуски, що дозволило оцінити точність кожного підходу на основі RMSE. Найкращі результати в усіх наборах продемонструвала поліноміальна регресія третього ступеня, що дозволяє вважати її найбільш ефективною для досліджуваних даних. А саме відновлення пропущених значень має вирішальне значення для подальшого аналізу, зокрема:

- підвищення точності прогнозування — забезпечує безперервність ряду для достовірних моделей;
- коректне виявлення трендів і сезонності — сприяє точному аналізу динаміки концентрацій;
- краща візуалізація та інтерпретація — забезпечує цілісне уявлення про дані;
- можливість застосування складних моделей — є необхідною умовою для роботи більшості сучасних алгоритмів.

Галузь використання – сфера екологічного моніторингу, аналітики якості повітря, урбаністики, а також у наукових дослідженнях, пов’язаних із аналізом часових рядів та екологічним прогнозуванням.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 81 pages, 3 tables, 39 figures, 2 appendixes, 17 sources.

AIR QUALITY MONITORING, PM2.5, TIME SERIES, DATA RECOVERY METHODS, ENVIRONMENTAL MONITORING, ENVIRONMENTAL FORECASTING.

The object of research is the process of analysing and data collection for air quality monitoring.

The subject of the study is methods and algorithms of machine learning and statistical analysis for recovering missing data in time series of PM2.5 concentration.

The purpose of the study is to investigate and comparatively analyse methods for rebuilding missing data in PM2.5 concentration time series in order to improve the accuracy of rehabilitation.

Materials, methods and equipment: structural and object-oriented programming, polynomial regression, spline interpolation (PCHIP and smoothing spline), ARIMA, SARIMA, Random Forest, Python programming language, pandas, numpy, matplotlib, scikit-learn, statsmodels, scipy, tkinter libraries, a personal computer with an Intel Core i7-12700F processor running the Microsoft Windows 10 operating system.

Results. Six methods for recovering missing values in the time series of PM2.5 concentrations based on real air monitoring data in Zaporizhzhia, Dnipro, and Lviv were implemented and compared: third-degree polynomial regression, spline interpolation (PCHIP and smoothing spline), ARIMA, SARIMA, and Random Forest. The quality of recovery was evaluated by the RMSE metric.

Based on the analysis, a Windows software application with a graphical interface was created. It allows you to load and process time series, select a

recovery method, visualise the results (graphs, tables) and export them for further analysis. The implementation was done in Python using the scikit-learn, statsmodels, tkinter, and Matplotlib libraries.

Conclusions. A systematic comparison of the effectiveness of different recovery methods was carried out. To do this, gaps were artificially modelled in the data, which allowed us to evaluate the accuracy of each approach based on RMSE. The best results in all sets were demonstrated by third-degree polynomial regression, which makes it the most effective for the data under study. The recovery of missing values is crucial for further analysis, in particular:

- improving forecasting accuracy - ensuring the continuity of the series for reliable models;
- correct detection of trends and seasonality - facilitates accurate analysis of concentration dynamics;
- better visualisation and interpretation - provides a holistic view of the data;
- the ability to use complex models is a prerequisite for most modern algorithms.

Applications - environmental monitoring, air quality analytics, urban studies, as well as research related to time series analysis and environmental forecasting.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	9
Вступ.....	10
1 Аналіз проблеми та постановка завдань.....	13
1.1 Забруднення повітря	13
1.2 Моніторинг якості повітря	15
1.3 Постановка задачі.....	19
2 Матеріали і методи.....	21
2.1 Огляд існуючих методів відновлення часових рядів	21
2.2 Огляд існуючих рішень	30
3 Опис програми.....	32
3.1 Призначення програми	32
3.2 Основний інтерфейс користувача	32
3.3 Основні функціональні можливості.....	34
4 Експлуатація, тестування та експериментальне дослідження.....	39
4.1 Умови запуску програми.....	39
4.2 Інструкція з використання.....	39
4.3 Тестування програми	40
4.4 Експериментальне дослідження	40
Висновки	57
Перелік джерел посилання	59
Додаток А Текст програми.....	62
Додаток Б Слайди презентації	73

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ANN	– Artificial Neural Network;
AR	– AutoRegressive;
ARIMA	– AutoRegressive Integrated Moving Average;
CSV	– Comma-Separated Values;
GUI	– Graphical User Interface;
I	– Integrated;
MA	– Moving Average;
PCHIP	– Piecewise Cubic Hermite Interpolating Polynomial;
PDF	– Portable Document Format;
PM	– Particulate Matter;
PNG	– Portable Network Graphics;
RMSE	– Root Mean Square Error;
SARIMA	– Seasonal AutoRegressive Integrated Moving Average;
XLSX	– Excel Spreadsheet;
ВООЗ	– Всесвітня організація охорони здоров'я;
ДГС	– Державна гідрометеорологічна служба;
СКП	– середньоквадратична помилка;
КМУ	– Кабінет Міністрів України.

ВСТУП

Забруднення атмосферного повітря належить до найсуттєвіших екологічних загроз, що негативно впливають на здоров'я населення та щорічно спричиняють мільйони випадків передчасної смерті й значні втрати років здорового життя [1, с. 7].

За даними ВООЗ, майже 99% світового населення проживає в умовах, де концентрація забруднювальних речовин у повітрі перевищує встановлені гігієнічні нормативи, при цьому найбільший вплив фіксується в країнах із низьким та середнім рівнем доходу [1, с. 7].

Тому у сучасних умовах стрімкого розвитку промисловості, транспорту та урбанізації якість атмосферного повітря дедалі частіше опиняється у центрі уваги наукової спільноти, державних структур та громадськості. Забруднення повітря має безпосередній вплив на здоров'я населення, стан довкілля та зміни клімату.

Одним із найнебезпечніших компонентів атмосферного забруднення для здоров'я людини є дрібнодисперсні тверді частинки, зокрема PM_{2.5} та PM₁₀. Завдяки своїм мікроскопічним розмірам — приблизно у 40 разів меншим за діаметр людської волосини — ці частинки є невидимими для неозброєного ока та здатні проникати глибоко в дихальні шляхи. Потрапляючи до альвеол легень, вони провокують запальні процеси, після чого, проникаючи в системний кровотік, можуть уражати серцево-судинну та центральну нервову системи, а також інші життєво важливі органи. Серед найбільш типових гострих реакцій організму на вплив твердих частинок фіксуються подразнення слизових оболонок очей, носа й горла, кашель, задишка та загальне погіршення дихальної функції [2].

У зв'язку з викликами, пов'язаними з якістю атмосферного повітря, особливої актуальності набуває забезпечення безперервного, достовірного та комплексного моніторингу рівня його забруднення, зокрема концентрації зважених частинок фракції PM_{2.5}. Проте в реальних умовах збирання даних

можуть виникати різноманітні труднощі: від технічних несправностей сенсорів до проблем із передачею даних. Це призводить до виникнення пропусків (втрат) у часових рядах вимірювань, що суттєво ускладнює подальший аналіз, прогнозування та прийняття управлінських рішень. Зважаючи на це, проблема відновлення відсутніх значень у часових рядах екологічного моніторингу набуває особливої актуальності.

Одним із найефективніших підходів до вирішення цієї проблеми є застосування методів математичного моделювання, статистичного аналізу та машинного навчання. Актуальні обчислювальні підходи, зокрема поліноміальна регресія, інтерполяційні методи на основі сплайнів, авторегресивні моделі типу ARIMA та SARIMA, а також ансамблеві алгоритми на основі дерев рішень, такі як Random Forest, засвідчують високу ефективність і точність у задачах відновлення пропущених значень у часових рядах. Вибір і застосування конкретного методу залежить від характеру даних, сезонності, наявності шумів та інших чинників.

У рамках дипломної роботи проведено дослідження ефективності згаданих методів у задачі відновлення відсутніх значень концентрації PM_{2.5} в даних екологічного моніторингу трьох міст. З метою практичної реалізації розроблено програмний застосунок з графічним інтерфейсом, який дає змогу обирати метод відновлення, візуалізувати результати та зберігати отримані дані. Програма реалізована мовою програмування Python із використанням таких бібліотек, як pandas, numpy, matplotlib, scikit-learn, statsmodels, scipy та tkinter, що забезпечує зручність користування та високий рівень модульності й гнучкості.

Розроблений застосунок може бути використаний в організаціях, що займаються екологічним моніторингом, для підвищення якості зібраних даних, що в свою чергу дозволяє здійснювати більш точні оцінки стану повітря та формувати обґрунтовані рекомендації щодо заходів із покращення екологічної ситуації.

Таким чином, дана робота є актуальною як з наукової, так і з прикладної точки зору, оскільки сприяє підвищенню достовірності екологічних даних та забезпечує інструментарій для їх ефективного аналізу та обробки.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ

1.1 Забруднення повітря

Забруднення атмосферного повітря в Україні є однією з найгостріших екологічних проблем, що має серйозні наслідки для здоров'я населення та довкілля. Ключовими антропогенними джерелами забруднення атмосферного повітря виступають промислові підприємства, транспортна інфраструктура, об'єкти енергетичного сектору, а також аграрна діяльність. Особливо високий рівень забруднення спостерігається в промислових регіонах, таких як Дніпропетровська, Донецька, Київська, Луганська та Запорізька області [3, с. 6].

Окрім очевидної шкоди для здоров'я населення, забруднення повітря чинить кумулятивний ефект, що призводить до погіршення якості життя, зниження продуктивності праці, зростання державних витрат на охорону здоров'я. Особливо вразливими до впливу шкідливих домішок у повітрі є діти, літні люди та особи з хронічними захворюваннями. В умовах воєнного часу в Україні ситуація ще більше ускладнюється через руйнування інфраструктури, пожежі, вибухи та зупинку природоохоронних систем на підприємствах [15, с. 132].

Атмосферне повітря містить значну кількість забруднювальних речовин, які можуть мати серйозний негативний вплив як на здоров'я людини, так і на стан довкілля. У таблиці 1.1 подано систематизовані відомості про основні види забруднювачів атмосферного повітря, типові джерела їх утворення, а також узагальнено їхній вплив на здоров'я людини та стан довкілля [10].

Таблиця 1.1 – Основні забруднювачі атмосферного повітря

Забруднювач	Хімічна формула / Позначення	Основні джерела	Вплив на здоров'я людини
Тверді частинки (PM2.5)	PM2.5	Автотранспорт, спалювання палива, промислові процеси, пилові бурі	Глибоке проникнення в легені, загострення астми, серцево-судинні та респіраторні захворювання
Тверді частинки (PM10)	PM10	Будівництво, сільське господарство, дорожній пил	Подразнення очей, носа, горла, ускладнене дихання
Діоксид азоту	NO ₂	Двигуни внутрішнього згоряння, ТЕС, металургія	Запалення легенів, зниження функції дихання, посилення симптомів астми
Озон (тропосферний)	O ₃	Фотохімічні реакції за участю NO _x та летких органічних сполук	Респіраторні порушення, подразнення очей, зниження імунітету Шкідливий для рослин,
Діоксид сірки	SO ₂	Спалювання вугілля, нафтопродуктів, промисловість	Подразнення дихальних шляхів, кашель, бронхіт
Оксид вуглецю	CO	Неповне згоряння палива, автомобільні вихлопи	Гіпоксія, головний біль, дезорієнтація, небезпечний для людей із серцевими хворобами
Леткі органічні сполуки	VOCs	Розчинники, фарби, нафтопереробка, вихлопи транспорту	Печіння в очах, головний біль, деякі з них — канцерогенні
Свинець	Pb	Стара автотехніка, виробництво акумуляторів, переробка металів	Токсичний для нервової системи, порушення розвитку у дітей

Одним із найпоширеніших та потенційно найнебезпечніших компонентів атмосферного забруднення є тверді частинки PM2.5 —

аеродисперсні частинки з аеродинамічним діаметром до 2,5 мкм, що формуються внаслідок процесів згоряння палива, промислової діяльності, а також під впливом як антропогенних, так і природних джерел [4, с. 49]. Завдяки своїм мікроскопічним розмірам, ці частинки можуть проникати глибоко в дихальні шляхи, досягаючи альвеол легенів, а звідти — потрапляти в кровообіг, що може спричиняти запальні процеси, порушення функцій серцево-судинної та нервової систем, а також інші серйозні захворювання.

В Україні проблема забруднення атмосферного повітря твердими частинками PM_{2.5} є надзвичайно актуальною, зокрема для урбанізованих зон та промислово розвинених регіонів. Згідно з результатами досліджень, концентрація PM_{2.5} демонструє виражену сезонну варіативність: у холодний період року фіксується її підвищення, що зумовлено інтенсивнішим використанням викопного палива для потреб опалення [4, с. 49]. Крім того, пожежі в природних системах також суттєво впливають на якість повітря.

1.2 Моніторинг якості повітря

В Україні якість атмосферного повітря контролюється через діяльність мережі спеціалізованих моніторингових станцій, які виконують вимірювання рівнів різних забруднювальних речовин, зокрема твердих частинок фракції PM_{2.5}. Однак, іноді через технічні несправності, обмеження в ресурсах або інші причини, моніторингові станції можуть пропускати вимірювання, що призводить до прогалин у даних. Це ускладнює точну оцінку екологічної ситуації та може впливати на ефективність прийняття управлінських рішень у сфері охорони довкілля [17, с.1034].

У даній роботі проведено дослідження методів відновлення пропущених значень у часових рядах, що характеризують динаміку концентрації частинок PM_{2.5} в атмосферному повітрі. Основним об'єктом дослідження виступив процес обробки та аналізу даних моніторингу якості повітря, зокрема задачі інтерполяції та апроксимації відсутніх вимірів.

Для отримання актуальної інформації про рівень забруднення повітря в Україні, зокрема концентрації PM2.5, можна скористатися онлайн-картами моніторингу, такими як Aqicn, які надають дані з більш ніж 500 станцій моніторингу з оновленням щогодини. На рисунку 1.1 можна побачити приклад даних моніторингу якості повітря з цього сайту [18].

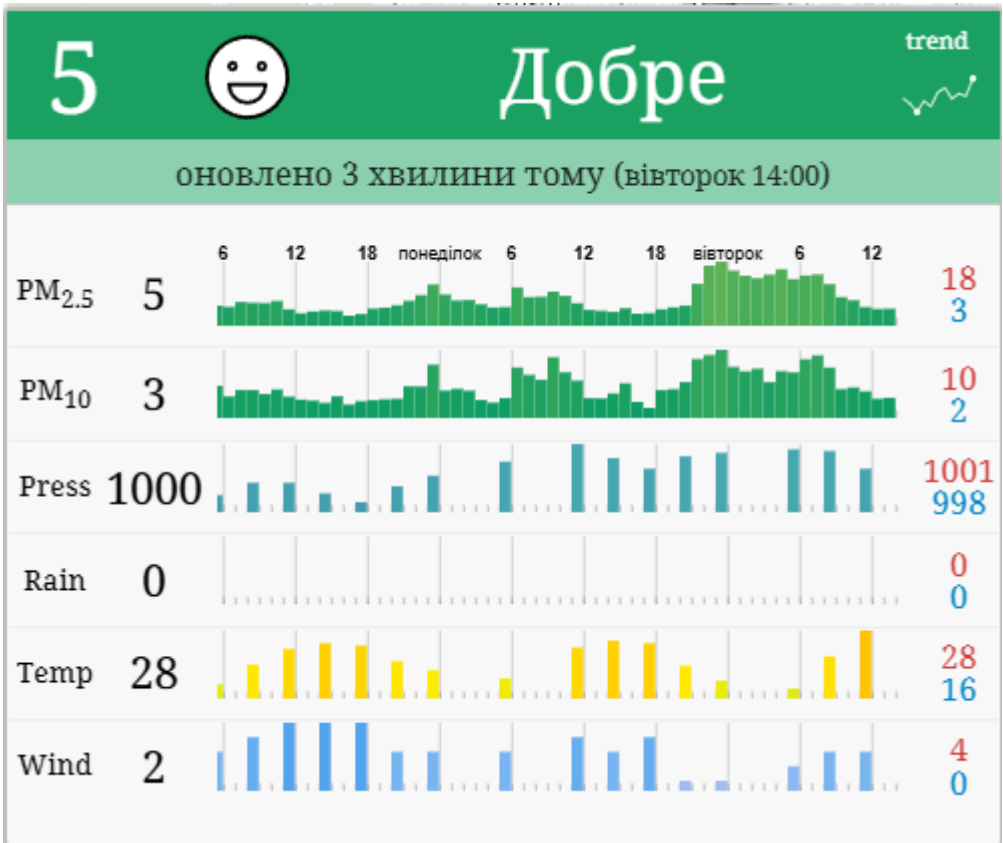


Рисунок 1.1 – Дані про забруднення повітря з сайту Aqicn

Аналіз рівнів забруднення повітря в Україні, зокрема концентрацій PM2.5, у різні часові періоди — до пандемії COVID-19, під час карантину та в умовах війни — має велике значення для розуміння впливу соціально-економічних і політичних факторів на якість атмосферного повітря. Таблиця 1.2 демонструє зміни середніх концентрацій PM2.5 у вибраних містах України протягом зазначених періодів [6].

Таблиця 1.2 – Середнє значення концентрації PM2.5 у різні періоди

Період	Запоріжжя ($\mu\text{g}/\text{m}^3$)	Дніпро ($\mu\text{g}/\text{m}^3$)	Львів ($\mu\text{g}/\text{m}^3$)	Примітки
До пандемії (2019)	20-21	21-22	17-19	Високі рівні PM2.5 були зафіксовані внаслідок інтенсивної промислової діяльності та транспорту.
Під час карантину (2020)	16-18	17-19	14-15	Під час карантину спостерігалось зниження концентрацій PM2.5 через обмеження руху транспорту та зменшення промислової активності.
Під час війни (2022-2025)	14-15	14-16	12-14	Під час війни рівні PM2.5 знизилися через зменшення економічної активності, але в окремих регіонах могли спостерігатися локальні підвищення через військові дії та пожежі.

Моніторинг якості повітря в Україні здійснюється за допомогою ДГС, яка проводить спостереження за станом забруднення в 39 містах на 129 стаціонарних постах. Проте, ефективність цього моніторингу обмежена через недостатнє фінансування та застаріле обладнання [12, с. 12].

З метою модернізації, Україна поступово впроваджує європейські стандарти моніторингу якості повітря. Зокрема, у 2019 році було прийнято постанову КМУ, яка передбачає оновлення нормативної бази та впровадження сучасних методів вимірювання.

У деяких містах України вже реалізовано сучасні системи моніторингу якості повітря. Наприклад, у Києві функціонує найбільша в Україні автоматична система моніторингу, яка включає 53 пункти спостережень, що забезпечують оперативне відображення даних у режимі реального часу через додаток «Київ Цифровий» [5].

Науково-технічний прогрес відкриває нові можливості для моніторингу якості повітря — зокрема, застосування дистанційних

спутникових спостережень, сенсорних мереж, алгоритмів машинного навчання та інтелектуального аналізу даних.

Моніторинг якості повітря в Україні є багаторівневою системою, що включає державні, муніципальні та громадські ініціативи. Попри наявні виклики, такі як застарілість обладнання та методів, країна активно працює над модернізацією системи моніторингу, впровадженням європейських стандартів та залученням громадськості до процесу контролю за якістю повітря [13]. Ці заходи є важливими для забезпечення екологічної безпеки та здоров'я населення.

1.2.1 Відновлення втрачених даних моніторингу

В умовах зростання уваги до проблеми забруднення атмосферного повітря особливого значення набуває забезпечення повноти та достовірності даних моніторингу, які є основою для подальшого аналізу, прогнозування екологічної ситуації та розробки управлінських рішень. Однак на практиці процес збору даних часто супроводжується технічними труднощами, що призводять до утворення прогалин у часових рядах вимірювань. Причинами таких втрат можуть бути як короткочасні збої в роботі обладнання моніторингових станцій, перебої з передачею даних або їх пошкодження під час зберігання, так і зовнішні обставини — зокрема, обмеження під час надзвичайних ситуацій, збройних конфліктів або пандемій.

Часові ряди є ключовим інструментом у дослідженні екологічних процесів, зокрема при аналізі забруднення атмосферного повітря. Вони представляють собою послідовність значень певного показника, зафіксованих у рівновіддалені моменти часу, що дозволяє виявляти тенденції, сезонні коливання та аномалії у динаміці концентрацій шкідливих речовин, таких як PM_{2.5}.

У контексті моніторингу якості повітря в Україні, дослідження часових рядів концентрацій PM_{2.5} дозволяє оцінити вплив різних факторів,

включаючи сезонні зміни, карантинні обмеження та воєнні дії, на рівень забруднення. Наприклад, дослідження даних з моніторингових станцій виявило, що максимальні рівні $PM_{2.5}$ припадають на холодний сезон, що зумовлено активним використанням викопного палива для опалення будинків [4, с. 52].

Наявність пропущених значень у наборах даних ускладнює проведення статистичного аналізу, знижує точність моделей прогнозування, а також спотворює результати довгострокового екологічного моніторингу. Особливу увагу слід приділяти аналізу концентрацій дрібних твердих частинок, зокрема $PM_{2.5}$, оскільки вони мають значний вплив як на здоров'я населення, так і на екологічну обстановку в регіонах.

У зв'язку з цим актуальним є розроблення і впровадження ефективних методів відновлення втрачених або пропущених значень у часових рядах даних моніторингу якості повітря. Такі методи мають враховувати специфіку екологічних даних, зокрема сезонність, автокореляцію, наявність пікових навантажень, а також можливі міжрегіональні залежності.

1.3 Постановка задачі

На основі проведеного аналізу предметної галузі основною метою даної роботи є дослідження та порівняння різних методів відновлення втрачених даних моніторингу якості атмосферного повітря за показником $PM_{2.5}$. З огляду на необхідність оцінки ефективності методів на даних з різними характеристиками було прийнято рішення виконати порівняльний аналіз на основі вимірювань, отриманих у трьох різних містах України: Дніпрі, Запоріжжі та Львові.

Одним із ключових завдань дослідження є апроксимація реальних даних моніторингу, зібраних із різних станцій, з метою якісного відновлення пропущених або втрачених значень. Для об'єктивного порівняння результатів застосування методів планується розрахунок СКП відновлених

значень, що дозволить кількісно оцінити точність апроксимації кожного з методів.

Результатом виконаної роботи стане комплексне порівняння ефективності методів апроксимації в різних умовах експлуатації, що дасть змогу визначити оптимальні підходи для відновлення даних моніторингу якості повітря.

Для реалізації поставленої мети буде розроблено програмне забезпечення, здатне здійснювати відновлення пропущених даних різними методами, обчислювати точність прогнозу за допомогою середньоквадратичної помилки та формувати графічні візуалізації результатів.

В межах дослідження передбачено вирішення наступних завдань:

- проведення детального аналізу характеристик даних моніторингу якості повітря;
- моделювання пропусків шляхом видалення окремих ділянок даних для подальшого тестування методів;
- імплементація вибраних методів відновлення даних;
- порівняння результатів апроксимації на основі обчислення показників точності.

2 МАТЕРІАЛИ І МЕТОДИ

Для проведення дослідження та порівняння ефективності відновлення даних було обрано шість методів. Детальний аналіз і характеристика кожного з них наведені нижче.

2.1 Огляд обраних методів відновлення часових рядів

Описані нижче методи є одними із найпопулярніших методів відновлення втрачених даних часових рядів.

2.1.1 Поліноміальна регресія

Поліноміальна регресія — це розвиток лінійної регресії, який дає змогу описувати більш складні взаємозв'язки між змінними шляхом застосування поліномів вищих ступенів. На відміну від простої лінійної регресії, яка передбачає лінійну (тобто прямолінійну) залежність між незалежною змінною x та залежною змінною y , поліноміальна регресія дозволяє апроксимувати нелінійні криві, що краще описують реальні процеси у природі та техніці [9, с. 100].

Поліноміальна регресія базується на побудові моделі виду. Рівняння для поліноміальної регресії наведено формулою (2.1):

$$y = \beta^0 + \beta^1 x + \beta^2 x^2 + \dots + \beta_n x^n + \varepsilon, \quad (2.1)$$

де y — залежна змінна (відгук);

x — незалежна змінна (предиктор);

$\beta^0, \beta^1, \dots, \beta_n$ — коефіцієнт моделі;

n — степінь полінома;

ε — похибка моделі.

Однією з основних переваг поліноміальної регресії є її здатність моделювати складні та нелінійні залежності, зокрема коли дані мають вигляд кривих, хвиль чи синусоїдальних коливань. Це робить її корисною у завданнях, де лінійна регресія дає недостатньо точні результати, зокрема у сфері екологічного моделювання, економіки, медицини, та прогнозуванні часових рядів.

У контексті задачі відновлення пропущених даних моніторингу повітря, поліноміальна регресія дозволяє з урахуванням наявної інформації побудувати плавну криву, яка апроксимує значення між наявними точками. Таким чином, можна заповнити прогалини у часових рядах значень PM2.5 досить ефективно, особливо коли дані мають чітко виражену тенденцію або сезонний компонент.

Попри свою гнучкість, метод має низку обмежень. Зокрема:

- перепідгонка (overfitting): зі збільшенням степеня полінома модель може надмірно точно "підганятися" під навчальні дані, втрачаючи здатність до узагальнення;
- чутливість до викидів: поліноміальна модель може сильно змінитися навіть при незначній зміні одного з вхідних значень;
- нестабільність при високих степенях: великі степені полінома можуть призвести до осциляцій кривої між точками, що погіршує точність відновлення.

2.1.2 Сплайн (RSHIP)

Метод RSHIP або сплайн на основі кубічного інтерполяційного полінома Ерміта є потужним інструментом для відновлення або апроксимації даних, особливо в тих випадках, коли необхідно зберегти монотонність та форму вихідного набору точок. На відміну від класичних кубічних сплайнів, які можуть створювати небажані осциляції або навіть перевищення між

заданими точками (ефект Гіббса), РСНІР спеціально розроблений для уникнення таких недоліків і забезпечує "гладку", але точну аппроксимацію.

РСНІР є кусково-поліноміальною інтерполяцією, де на кожному інтервалі між точками використовується кубічний поліном Ерміта, побудований так, щоби значення та похідні в точках з'єднання були узгоджені. Це дозволяє зберегти основну геометричну характеристику даних – зростання, спадання, або плоскостність функції.

Формально, якщо задано набір точок $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, метод будує кубічний поліном на кожному підінтервалі $[x_i, x_{i+1}]$, що проходить через відповідні значення y_i та y_{i+1} і має визначені значення похідних на обох кінцях інтервалу.

Перевагами цього методу є:

- збереження монотонності: якщо вхідні дані зростають або спадають, інтерполяційна крива також зберігає цю тенденцію. Це особливо важливо для часових рядів екологічних даних, де зміна концентрації забруднювача повинна бути плавною і фізично обґрунтованою;
- відсутність осциляцій: на відміну від звичайних кубічних сплайнів, які можуть створювати штучні коливання, РСНІР пригнічує "перерегулювання" між точками, зберігаючи стабільність наближення;
- простота реалізації: сучасні програмні пакети, такі як SciPy в Python, дозволяють реалізувати метод РСНІР з мінімальними зусиллями.

Разом із перевагами метод має й деякі слабкі сторони:

- менш гладка друга похідна: РСНІР гарантує безперервність лише значень функції та її першої похідної. Друга похідна може бути розривною в точках стику, що може бути недоліком у задачах, де важлива гладкість вищих порядків;
- не застосовується для екстраполяції: метод не дає коректних результатів поза межами заданого інтервалу (наприклад, для прогнозування майбутніх значень), і використовується виключно для інтерполяції.

2.1.3 Згладжуючий сплайн

Метод згладжуючого сплайна є ключовим засобом інтерполяції та регресійного аналізу, що дає можливість відтворювати тренди в даних із шумом або пропущеними значеннями. Він особливо корисний у тих випадках, коли потрібно знайти баланс між точною аппроксимацією значень і загальною гладкістю функції. У контексті обробки даних екологічного моніторингу — наприклад, вимірювань рівня PM2.5 — цей метод дозволяє адекватно відновити пропущені або втрачені значення, враховуючи варіативність та нерівномірність вимірювань.

У задачах аналізу забруднення повітря, зокрема при роботі з концентраціями твердих частинок PM2.5, вимірювання часто є неповними або містять випадковий шум (через погодні умови, технічні збої, локальні піки забруднення тощо). Згладжуючий сплайн у цьому контексті дозволяє:

- відновити пропущені значення без різких стрибків;
- оцінити загальний екологічний тренд протягом тривалого часу;
- зменшити вплив одиничних пікових значень, які можуть бути викликані короткочасними подіями.

На відміну від звичайних інтерполяційних сплайнів, які проходять точно через усі надані точки, згладжуючий сплайн створює функцію, що не обов’язково проходить через усі спостереження, а натомість мінімізує функціонал, який поєднує міру похибки апроксимації та рівень гладкості кривої. Формально задача виглядає так (2.2):

$$\min \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int (f''(x))^2 dx, \quad (2.2)$$

де y_i — значення функції в точці x_i ;

$f(x_i)$ — згладжуюча функція;

$f''(x)$ — друга похідна функції, яка визначає ступінь вигину;

λ — параметр згладжування.

Переваги згладжуючого сплайна:

гнучкість: дозволяє адаптуватися до шумних даних, уникаючи переоснащення;

- гладкість: отримана функція має неперервні похідні до другого порядку, що робить її корисною для подальшого чисельного аналізу, наприклад, інтегрування чи диференціювання;

- можливість контролю: параметр згладжування дає змогу досліднику самостійно визначати необхідний рівень компромісу;

- ефективність при роботі з часовими рядами, особливо коли є окремі пропуски або флуктуації, які не слід інтерпретувати як справжні тренди.

Втім, цей метод не позбавлений певних недоліків:

- неоднозначність вибору параметра згладжування: визначення оптимального значення λ часто потребує емпіричної оцінки або крос-валідації;

- нестабільність при сильних піках: при дуже різкому зростанні або спаданні значень можлива втрата точності;

- не використовується для екстраполяції: поза межами інтервалу інтерполяція може бути некоректною або нестійкою.

2.1.4 Метод ARIMA

Метод ARIMA є одним із найпоширеніших і найефективніших статистичних підходів для моделювання та прогнозування часових рядів. Він широко застосовується у різних сферах науки і техніки, зокрема в економіці, екології, метеорології та багатьох інших галузях, де дані мають часову залежність і трендові чи сезонні компоненти [10, с. 132]. У контексті моніторингу якості повітря, наприклад, для аналізу концентрації шкідливих частинок PM_{2.5}, ARIMA дозволяє моделювати поведінку забруднення в часі

та відновлювати пропущені значення, що є критично важливим для отримання повної та надійної картини екологічної ситуації.

Показники якості повітря, зокрема рівні $PM_{2.5}$, як правило, характеризуються сезонними змінами, наявністю загальних тенденцій та випадковими флуктуаціями. ARIMA дає змогу моделювати ці складні закономірності та відновлювати пропущені дані.

Модель ARIMA об'єднує три ключові компоненти:

- AR — означає, що поточне значення часового ряду залежить від кількох попередніх значень ряду (лагів). Кількість таких лагів позначається параметром p ;

- I — це ступінь різницювання ряду для досягнення стаціонарності (зняття тренду або сезонності). Параметр d вказує, скільки разів потрібно взяти різницю між послідовними значеннями, щоб отримати стаціонарний ряд;

- MA — моделює залежність поточного значення від помилок попередніх передбачень (лагів помилок). Кількість лагів помилок позначається параметром q .

Математично модель ARIMA записується як $ARIMA(p, d, q)$ і має вигляд (2.3):

$$\Phi(B)(1 - B)^d y_t = \theta(B)\varepsilon_t, \quad (2.3)$$

де y_t - спостережуване значення часового ряду у момент часу t ;

B - оператор зсуву (лаг-оператор);

$\Phi(B)$ - поліном авторегресії ступеня p ;

$\theta(B)$ - поліном ковзного середнього ступеня q ;

ε_t - випадкова помилка (білий шум).

Переваги цього методу:

- гнучкість у моделюванні: ARIMA може описувати широкий спектр часових рядів завдяки комбінації авторегресії, інтегрування і ковзного середнього;
- можливість прогнозування: метод дає змогу робити прогнози на майбутні періоди з урахуванням історичних даних;
- обробка нестабільних даних: параметр інтеграції d допомагає перетворити нестационарний часовий ряд у стаціонарний, що є умовою застосування багатьох статистичних методів;
- часова залежність: ARIMA враховує автокореляцію даних, що є важливим при роботі з екологічними даними, які часто мають внутрішню залежність у часі.

Обмеження методу:

- необхідність стаціонарності: якщо часовий ряд має складну сезонність, потрібні додаткові перетворення або застосування сезонної ARIMA (SARIMA);
- вразливість до сильних аномалій: різкі несподівані коливання можуть знизити якість відновлення;
- значна кількість параметрів у моделі ускладнює процес підбору оптимальних значень p , d , q і вимагає проведення додаткового аналізу.

2.1.5 Метод SARIMA

Метод SARIMA є розширенням класичної моделі ARIMA, спеціально адаптованим для аналізу та прогнозування часових рядів із яскраво вираженою сезонністю. Цей метод широко використовується у різних галузях науки і техніки, де спостерігаються періодичні коливання даних, зокрема в екології, економіці, метеорології, енергетиці та багатьох інших сферах. У контексті моніторингу якості повітря SARIMA дозволяє враховувати як трендові зміни, так і сезонні варіації концентрацій

забруднювачів, таких як частинки PM2.5, що суттєво підвищує точність відновлення пропущених даних та прогнозування.

Метод SARIMA є потужним та гнучким інструментом для аналізу та прогнозування часових рядів із сезонними складовими. Використання цього методу в екологічному моніторингу, зокрема для заповнення пропущених даних і прогнозування концентрацій шкідливих частинок PM2.5, сприяє підвищенню точності та достовірності отриманих результатів. Це має велике значення для оцінки стану довкілля, розробки стратегій зменшення забруднення та захисту здоров'я населення.

Модель SARIMA є доповненням до ARIMA, яке включає сезонні складові для врахування періодичних коливань у часових рядах. Вона описується параметрами (2.4):

$$SARIMA(p, d, q)(P, D, Q)_m, \quad (2.4)$$

де p, d, q – параметри нерегулярної (нерегулярної, нет сезонної) частини моделі;

p – порядок авторегресії (AR);

d – ступінь інтегрування (диференціювання) для досягнення стаціонарності;

q – порядок ковзного середнього (MA);

P, D, Q – аналогічні параметри для сезонної частини моделі;

P – порядок сезонної авторегресії;

D – ступінь сезонного диференціювання;

Q – порядок сезонного ковзного середнього;

m – довжина сезонного періоду (наприклад, 12 для щомісячних даних із річною сезонністю).

Математично модель SARIMA комбінує нерегулярні та сезонні оператори для забезпечення адекватного моделювання складних часових рядів.

Переваги даного методу:

- з урахуванням сезонності — на відміну від стандартної ARIMA, SARIMA здатна моделювати сезонні закономірності, що робить її більш придатною для даних із вираженою періодичністю;
- гнучкість у побудові моделей — можливість комбінувати сезонні та нерегулярні компоненти;
- висока точність прогнозів — особливо у випадках, коли дані мають як трендові, так і сезонні зміни;
- застосовність у широкому спектрі задач, від економічного аналізу до екологічного моніторингу.

2.1.6 Random Forest

Метод Random Forest (випадковий ліс) є одним із найпопулярніших та ефективних алгоритмів машинного навчання, який широко застосовується для розв'язання задач класифікації та регресії, у тому числі в галузі обробки екологічних даних та відновлення пропущених значень у часових рядах. Його популярність обумовлена високою точністю, стійкістю до перенавчання (overfitting), а також здатністю працювати з великою кількістю вхідних змінних та складними нелінійними залежностями.

У контексті дослідження якості повітря та аналізу концентрації забруднювачів, таких як PM_{2.5}, Random Forest широко використовується для імпутації (відновлення) пропущених значень у часових рядах. Пропуски можуть виникати через технічні несправності моніторингових станцій, погодні умови або людський фактор. Імпутація є критичною для забезпечення цілісності даних і коректності подальших екологічних висновків.

Переваги Random Forest у цій задачі полягають у тому, що він:

- враховує складні нелінійні взаємозв'язки між концентраціями забруднювачів та допоміжними змінними (погодними показниками, часом доби, місцем розташування станції);
- може інтегрувати інформацію з багатьох джерел і враховувати багатовимірність екологічних даних;
- забезпечує більш точні та стабільні оцінки пропущених значень порівняно з простими статистичними методами (середнє, лінійна інтерполяція).

Random Forest є ефективним і стійким методом для моделювання складних екологічних явищ та заповнення пропущених значень у даних моніторингу якості повітря. Завдяки здатності опрацьовувати багатовимірні дані і нелінійні залежності, він суттєво покращує якість досліджень і дозволяє отримувати більш точні й надійні екологічні оцінки. Впровадження цього методу в екологічний аналіз сприяє підвищенню достовірності отриманих результатів і формуванню ефективних заходів з охорони навколишнього середовища.

2.2 Огляд існуючих рішень

Огляд існуючих методів відновлення та прогнозування даних часових рядів є важливим етапом у дослідженні та застосуванні ефективних підходів до аналізу складних динамічних систем. У науковій літературі представлено широкий спектр методів, які можна класифікувати за різними критеріями, такими як статистичні, машинного навчання та гібридні підходи.

Сучасні підходи до аналізу часових рядів включають використання методів машинного навчання, таких як метод опорних векторів (SVM), дерева рішень, k-найближчих сусідів (k-NN) та нейронні мережі різних архітектур. Зокрема, у роботі Гурєвої та ін. розглянуто застосування цих методів для прогнозування фінансових часових рядів, що демонструє їхню

здатність моделювати складні нелінійні залежності та враховувати додаткові фактори, такі як новини чи поведінка користувачів [7, с. 21-23].

Поєднання статистичних методів та методів машинного навчання дозволяє створювати гібридні моделі, які враховують переваги обох підходів. Наприклад, у роботі Тищенко та ін. представлено методологію, яка поєднує аналіз часових рядів та виявлення аномалій за допомогою алгоритмів машинного навчання для моделювання впливу цифрових загроз на фінансові ринки [8, с. 208-211].

Для відновлення пропущених значень у часових рядах також застосовуються прості методи, такі як заміна пропусків середнім, зваженим середнім або медіаною. У дослідженні Камінського та ін. розглянуто ефективність цих методів на прикладі штучних часових рядів із пропущеними значеннями, що демонструє їхню доцільність у випадках з невеликою кількістю пропусків [9, с. 99-101].

Огляд існуючих методів відновлення та прогнозування даних часових рядів свідчить про різноманіття підходів, кожен з яких має свої переваги та обмеження. Вибір конкретного методу залежить від характеру даних, наявності сезонності, обсягу пропущених значень та інших факторів. Комбінація різних методів та використання гібридних моделей може забезпечити більш точне та надійне відновлення даних у складних динамічних системах.

3 ОПИС ПРОГРАМИ

3.1 Призначення програми

Метою розробленої програми є автоматизація процесу відновлення (апроксимації) відсутніх значень у часових рядах, які містять інформацію про концентрацію шкідливих часток PM_{2.5} у повітрі. Така задача є актуальною в контексті екологічного моніторингу, оскільки в реальних умовах зчитування даних часто виникають пропуски через технічні несправності датчиків, погодні умови, перебої живлення тощо. Відсутність таких даних унеможливорює повноцінний аналіз динаміки зміни забруднення повітря, тому їх якісне відновлення є критично важливим.

Програма розроблена як настільний додаток із графічним інтерфейсом, що забезпечує простоту використання навіть для користувачів, які не мають глибоких знань у галузі машинного навчання або статистики.

Завдяки використанню кількох різних алгоритмів (регресійних, інтерполяційних, статистичних та машинного навчання), програма дозволяє порівнювати ефективність підходів та обирати найбільш точний метод для конкретного типу даних. Це забезпечує високу гнучкість та практичну цінність створеного програмного продукту в умовах змінного та непередбачуваного середовища збору даних.

3.2 Основний інтерфейс користувача

У межах даного дослідження було реалізовано графічний інтерфейс користувача (GUI), який забезпечує зручний доступ до функціоналу аналізу, відновлення та візуалізації даних вимірювання концентрації дрібнодисперсного пилу PM_{2.5}. Основною метою створення інтерфейсу було забезпечення інтерактивної взаємодії користувача з алгоритмічними модулями, а також формування прозорого та інтуїтивно зрозумілого процесу обробки часових рядів.

Інтерфейс реалізовано з використанням мови програмування Python з залученням бібліотеки Tkinter, яка надає достатньо гнучкий інструментарій для побудови віконних застосунків. Структура інтерфейсу відповідає принципам модульності та логічного поділу функцій, що дозволяє легко орієнтуватися у доступному функціоналі та знижує імовірність помилкових дій з боку користувача.

Головне вікно застосунку містить:

- панель завантаження даних, де реалізовано можливість імпорту файлів у форматі CSV або XLSX з часовими рядами вимірювання. Користувач має можливість попередньо переглянути завантажений набір даних у вигляді таблиці;
- блок вибору методу відновлення, де представлено список доступних алгоритмів (Поліноміальна регресія, Сплайн РСНІР, Згладжуючий сплайн, ARIMA, SARIMA, Random Forest). Реалізовано механізм валідації вибору, що унеможливорює некоректну ініціалізацію обчислень;
- модуль візуалізації результатів, який забезпечує відображення графіків із відновленими та реальними значеннями в інтерактивному вигляді. Для цього було використано бібліотеки Matplotlib та Seaborn, що дозволяють якісно візуалізувати тренди, коливання та рівень відновлення для обраного методу;
- секція експорту результатів, де реалізована можливість збереження графіків та отриманих значень похибок (зокрема середньоквадратичної помилки) у форматах PNG, PDF або CSV. Це забезпечує подальшу інтеграцію результатів у звітну документацію або зовнішні аналітичні системи.

Візуальне представлення інтерфейсу користувача наведено на рисунку 3.1.

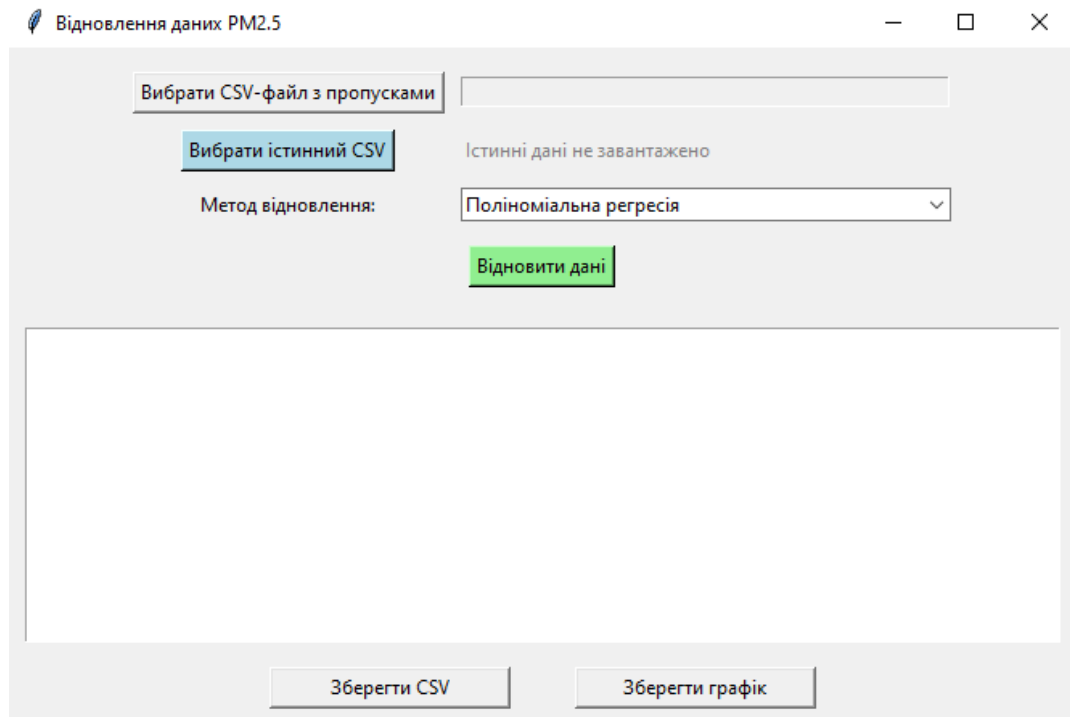


Рисунок 3.1 – Інтерфейс користувача

Також у рамках інтерфейсу було реалізовано базову обробку виключень, що дозволяє інформувати користувача про наявність помилок у введених даних або при виконанні обчислень. Повідомлення виводяться у зрозумілому форматі та не переривають виконання роботи програми.

Таким чином, реалізований інтерфейс користувача не лише виконує роль зручного інструменту взаємодії з обчислювальними модулями, але й виступає як повноцінна аналітична платформа для дослідження методів реконструкції пропущених значень у часових рядах, забезпечуючи зручність, функціональність та наочність на кожному з етапів роботи.

3.3 Основні функціональні можливості

Розроблена програма забезпечує комплексне відновлення відсутніх значень у часових рядах, що стосуються концентрацій часток PM2.5 у повітрі. Основна увага приділяється гнучкості, точності та зручності використання. Функціональні можливості програми охоплюють повний цикл

обробки даних — від їхнього завантаження до аналізу якості відновлення та збереження результатів. Нижче розглянуто ключові функції докладніше.

3.3.1 Завантаження та перевірка вхідних даних

Зчитування CSV-файлу з пропущеними значеннями.

Користувач завантажує файл із даними, який повинен містити два обов'язкові стовпці:

- date — календарна дата у форматі YYYY-MM-DD;
- median — значення середньої концентрації PM2.5 на відповідну дату.

Перевірка коректності даних.

Програма автоматично перевіряє:

- чи має файл правильне розширення (.csv);
- чи присутні обов'язкові стовпці;
- чи немає проблем із читанням файлу;
- чи формат даних відповідає очікуваному (дати перетворюються у формат datetime).

У разі виникнення помилок користувачу виводяться відповідні повідомлення через спливаючі вікна (messagebox). На рисунку 3.2 можна побачити один із варіантів такого повідомлення.

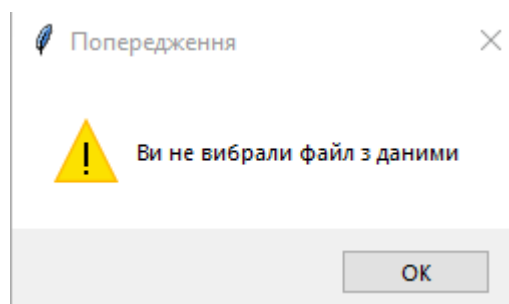


Рисунок 3.2 – Повідомлення про помилку

3.3.2 Вибір методу відновлення пропущених значень

Програма надає користувачеві можливість обрати один з кількох методів відновлення даних:

- поліноміальна регресія (степеня 3): підходить для глобальної апроксимації загальної тенденції даних;
- сплайн (PCHIP): метод побудови кривих, що зберігають монотонність та плавність між відомими точками;
- згладжуючий сплайн (Univariate Spline): компроміс між точністю проходження через точки та згладжуванням шумів у даних;
- ARIMA: класична модель часових рядів, що враховує автокореляцію і сезонність;
- SARIMA: розширена версія ARIMA з урахуванням сезонних ефектів;
- Random Forest Regressor: сучасний метод машинного навчання на основі ансамблю дерев рішень, що добре справляється з нелінійними залежностями.

Ці методи реалізовано через бібліотеки Scikit-learn, SciPy, Statsmodels, що забезпечує надійність і достовірність обчислень.

3.3.3 Автоматичне виявлення та інтерполяція пропущених днів

Після вибору методу програма:

- автоматично визначає відсутні дні у часовому ряді (тобто пропущені календарні дати);
- відновлює значення для кожного з пропущених днів за вибраною моделлю;
- формує окрему таблицю з відновленими значеннями (reconstructed_df).

Таким чином, створюється повний часовий ряд без прогалин, з можливістю подальшого аналізу.

3.3.4 Порівняння з істинними значеннями (опційно)

Програма підтримує завантаження другого CSV-файлу — з істинними значеннями, які використовуються для оцінки точності відновлення:

- ці дані також повинні містити date і median;
- програма об'єднує реконструйовані та істинні значення по днях;
- якщо такі значення доступні, розраховується RMSE — метрика, яка відображає середньоквадратичне відхилення між відновленими та реальними даними.

Ця функція особливо корисна для об'єктивного порівняння ефективності різних методів.

3.3.5 Візуалізація результатів

Побудова лінійного графіка:

- червона лінія — апроксимація (відновлені або згладжені значення);
- синя лінія — істинні значення (за наявності);
- зелені хрестики — саме ті дні, для яких були відновлені пропущені значення.

Графік будується з використанням бібліотеки Matplotlib і виводиться у новому вікні для зручності перегляду.

3.3.6 Збереження результатів

Для зручності подальшої роботи з результатами передбачено:

- експорт таблиці відновлених значень у CSV-файл — користувач може зберегти результати у вигляді таблиці для подальшої обробки в Excel, Google Sheets або іншому програмному забезпеченні;
- збереження графіка у форматі PNG — результат візуалізації можна використати у звітах, презентаціях, друкованих матеріалах.

3.3.7 Інші особливості

Особливості роботи програми:

- програма працює повністю автономно, без підключення до Інтернету;
- всі обчислення відбуваються локально на машині користувача;
- інтерфейс та повідомлення україномовні, що забезпечує зручність у використанні у вітчизняному академічному середовищі;
- враховано обробку помилок — якщо користувач завантажив неправильний файл або виникли проблеми при обробці, програма повідомляє про це у зрозумілій формі.

3.3.8 Висновок

Таким чином, програма реалізує повноцінний функціонал для аналізу та реконструкції часового ряду PM_{2.5}. Її можливості охоплюють усі етапи: від завантаження і перевірки даних до їхнього інтелектуального відновлення, візуалізації, порівняння точності та збереження результатів. Гнучкість у виборі алгоритмів робить її ефективним інструментом як для дослідників, так і для практичних аналітиків.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Умови запуску програми

Розроблена програма реалізована мовою програмування Python з графічним інтерфейсом користувача (GUI), створеним з використанням бібліотеки tkinter. Середовище виконання:

- операційна система: Windows 10/11 або Linux;
- Python: версія 3.9+;
- необхідні бібліотеки: pandas, numpy, matplotlib, scikit-learn, scipy, statsmodels, tkinter.

Програму можна запускати безпосередньо за допомогою Python інтерпретатора.

4.2 Інструкція з використання

Інтерфейс програми дозволяє користувачеві у декілька кроків завантажити дані, обрати метод заповнення пропущених значень та переглянути результати.

Крок 1. Користувач натискає кнопку «Вибрати CSV-файл з пропусками» та обирає файл, що містить стовпці date і median.

Крок 2. За бажанням можна завантажити істинні значення через кнопку «Вибрати істинний CSV» для розрахунку похибки.

Крок 3. У випадяючому списку обирається метод апроксимації: поліноміальна регресія, сплайн (PCHIP), згладжуючий сплайн, ARIMA, SARIMA, Random Forest.

Крок 4. Натискається кнопка «Відновити дані», після чого:

- виводяться результати відновлення в текстовому полі;
- будується графік зі справжніми, апроксимованими та відновленими даними;

– за наявності істинних даних обчислюється середньоквадратична помилка (RMSE).

Результати можна зберегти у форматі CSV або PNG.

4.3 Тестування програми

Програму було протестовано за допомогою функціонального та інтеграційного підходів. Було перевірено:

- роботу з валідними та невалідними CSV-файлами;
- реакцію на відсутність необхідних стовпців (date, median);
- коректність обчислень кожного з методів;
- обробку випадків, коли відсутні пропущені значення;
- відображення графіка та коректність його збереження.

Програма коректно обробляє винятки та повідомляє користувача про помилки через спливаючі вікна.

4.4 Експериментальне дослідження

Для проведення дослідження було використано реальні дані моніторингу якості повітря за показником PM2.5, що характеризує концентрацію дрібнодисперсного пилу в атмосфері. Збір даних здійснювався протягом 2025 року в трьох українських містах: Запоріжжі, Дніпрі та Львові. Вибір саме цих населених пунктів обумовлений відмінностями в географічному розташуванні та екологічній ситуації. Запоріжжя розташоване в безпосередній близькості до зони бойових дій, що може впливати на стан повітря як через військову активність, так і через порушення інфраструктури. Дніпро є великим промисловим центром із функціонуючими підприємствами, що генерують значне антропогенне навантаження на атмосферу. Львів, у свою чергу, знаходиться на відносно безпечній відстані від зони бойових дій і вирізняється більш стабільною інфраструктурною

ситуацією, зокрема значною часткою житлової та рекреаційної забудови. Цей підхід дає змогу вивчати змінність динаміки PM2.5 у різних соціально-економічних та географічних контекстах.

На рисунках 4.1, 4.2 та 4.3 представлено результати вимірювань концентрації PM2.5, здійснені протягом 2025 року.

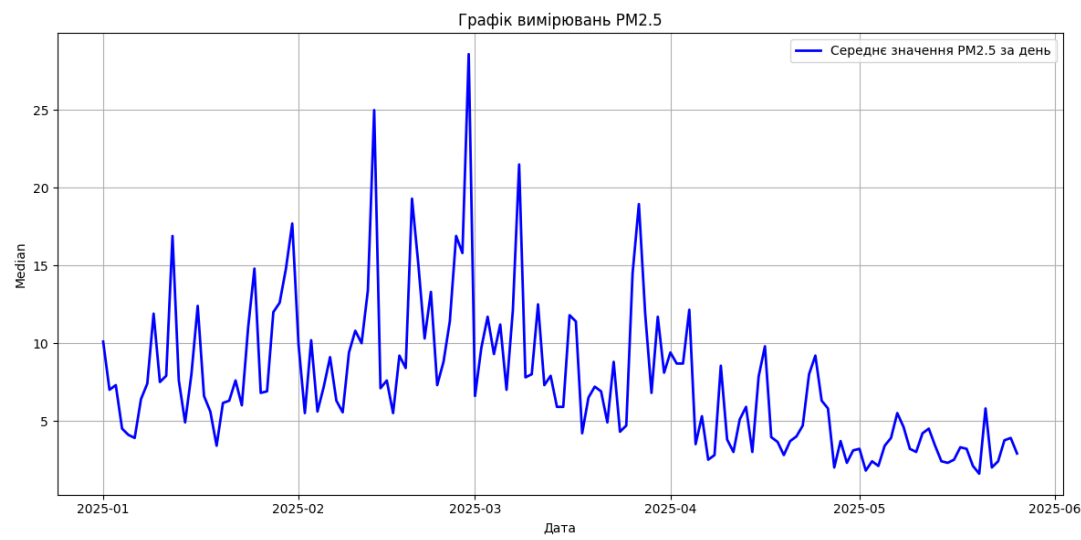


Рисунок 4.1 – Дані мониторінгу в місті Запоріжжя

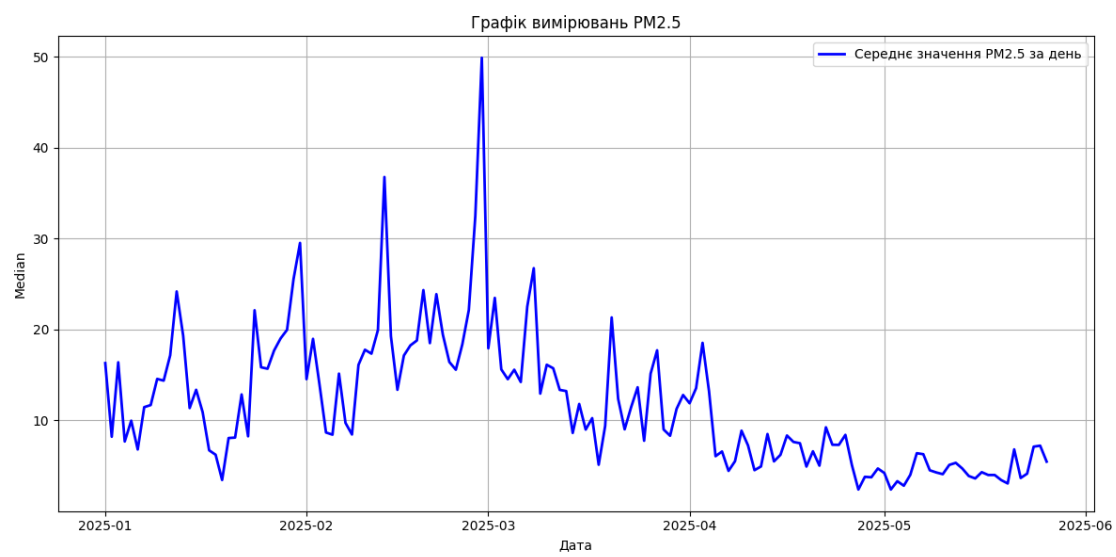


Рисунок 4.2 – Дані мониторінгу в місті Дніпро

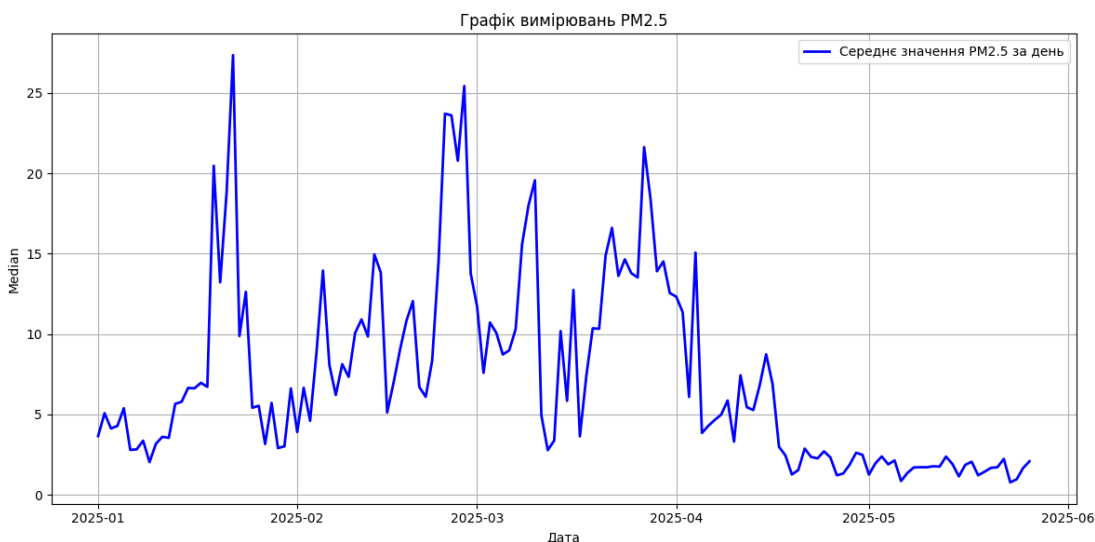


Рисунок 4.3 – Дані мониторінгу в місті Львів

З метою моделювання ситуації з втратою даних та подальшої оцінки ефективності методів їх відновлення, з початкового набору було цілеспрямовано вилучено значення показника PM2.5 за один тиждень у кожному з наступних місяців: лютому, березні та квітні. Такий підхід дозволяє створити контрольоване середовище з пропущеними даними, що є необхідним для об'єктивного порівняння точності різних методів реконструкції часових рядів.

4.4.2 Використані методи

У цій роботі для заповнення пропущених даних у часовому ряді медіанних концентрацій PM2.5 застосовано метод поліноміальної регресії третього ступеня. Цей підхід було обрано з огляду на його здатність апроксимувати нелінійні залежності між часовими відмітками та відповідними значеннями показника забруднення повітря.

Вихідні дані зчитуються з CSV-файлу, де кожен запис містить дату та відповідне значення медіани. Дати було перетворено у числовий формат — кількість днів відносно мінімальної дати у вибірці. Це забезпечує можливість застосування регресійного аналізу.

Для побудови моделі виконано розширення ознак шляхом генерації поліноміальних характеристик за допомогою інструменту `PolynomialFeatures` з бібліотеки `scikit-learn`. У роботі використано поліном третього ступеня, що дозволяє описати кубічні залежності. Далі, за допомогою лінійної регресії (`LinearRegression`) було навчено модель на розширених вхідних даних (матриці характеристик), після чого здійснено прогноз значень як для наявних, так і для відсутніх (пропущених) днів.

На рисунках 4.4, 4.5 та 4.6 представлені графіки з відтвореними точками вимірювань, отримані за допомогою методу «Поліноміальна регресія».

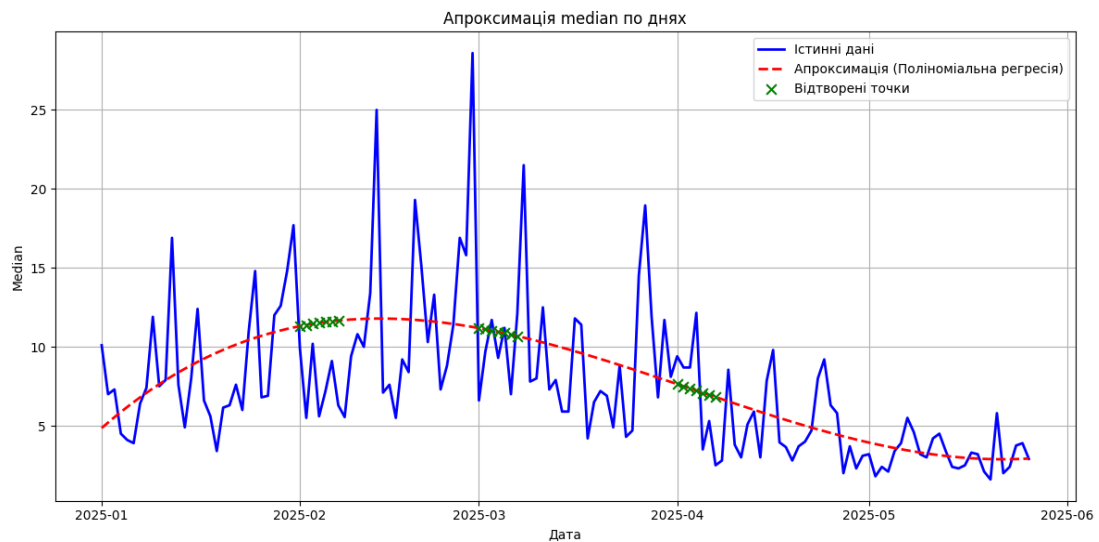


Рисунок 4.4 – Відтворенні дані в місті Запоріжжя

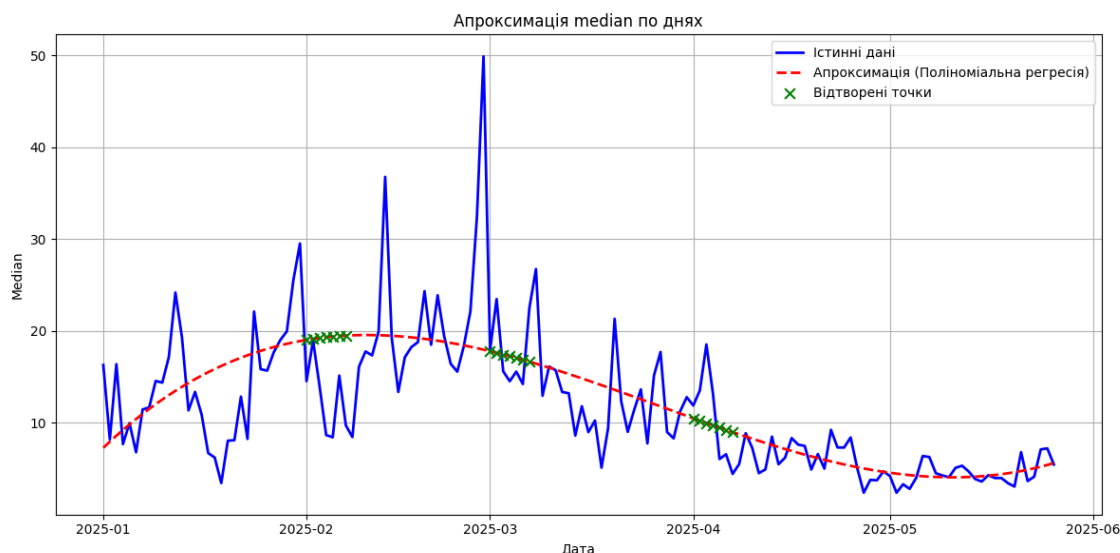


Рисунок 4.5 – Відтворенні дані в місті Дніпро

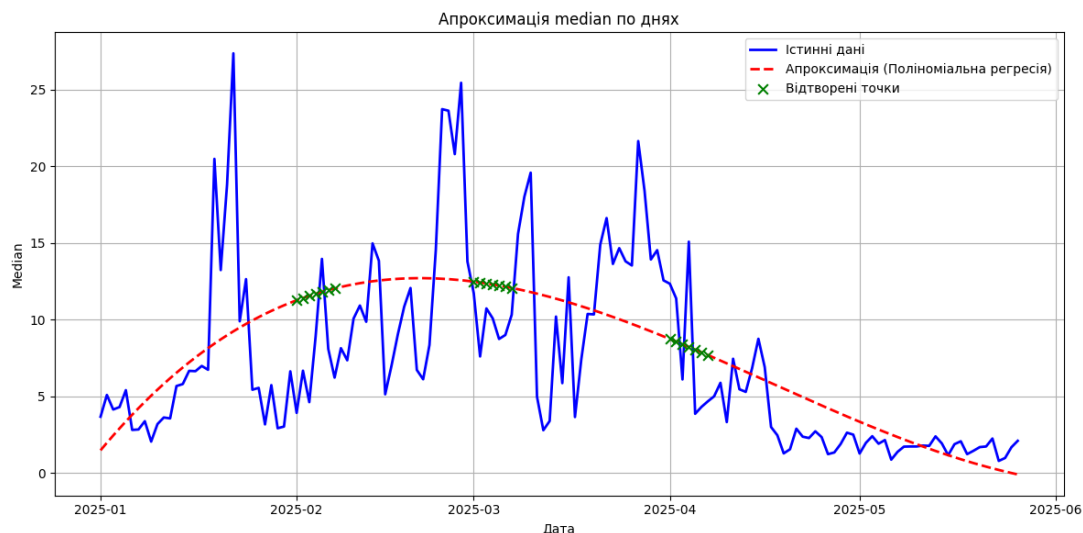


Рисунок 4.6 – Відтворенні дані в місті Львів

Також у рамках дослідження для відновлення пропущених значень у часовому ряді медіанних концентрацій PM_{2.5} було застосовано метод інтерполяції за допомогою РСНІР. Цей метод дозволяє побудувати гладку функцію, що проходить через задані дискретні точки, зберігаючи монотонність і форму вихідних даних, що особливо важливо для часових рядів, де недопустимі різкі та фізично необґрунтовані коливання.

Для підготовки даних кожна дата було перетворено у числовий формат у вигляді кількості днів від початкової дати спостережень. Метод

РСНІР реалізовано із застосуванням функції `PchipInterpolator` із бібліотеки `SciPy`, що дозволяє побудувати кусочно-квадратичний ермітовий сплайн. Такий сплайн забезпечує неперервність першої похідної, а також уникнення осциляцій, властивих класичним кубічним сплайнам, особливо у випадках монотонних або близьких до монотонних даних.

В результаті роботи з інтерполяцією РСНІР досягається висока якість відновлення пропущених значень, збереження фізично адекватної динаміки сигналу, що було підтверджено графічною візуалізацією та порівнянням із істинними спостереженнями.

На рисунках 4.7, 4.8 та 4.9 представлені графіки з відтвореними точками вимірювань, отримані за допомогою методу «Сплайн (РСНІР)».

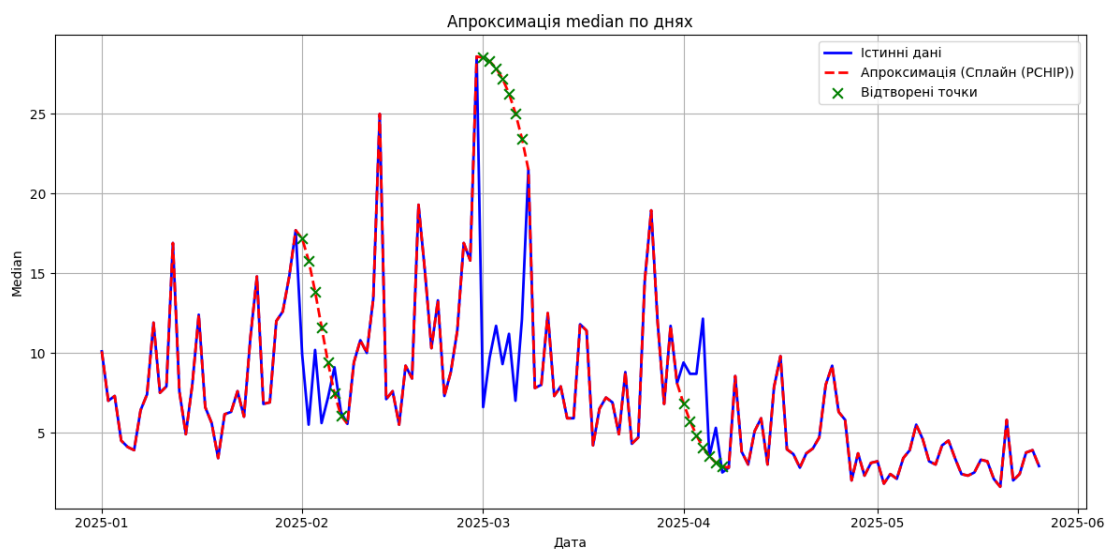


Рисунок 4.7 – Відтворені дані в місті Запоріжжя

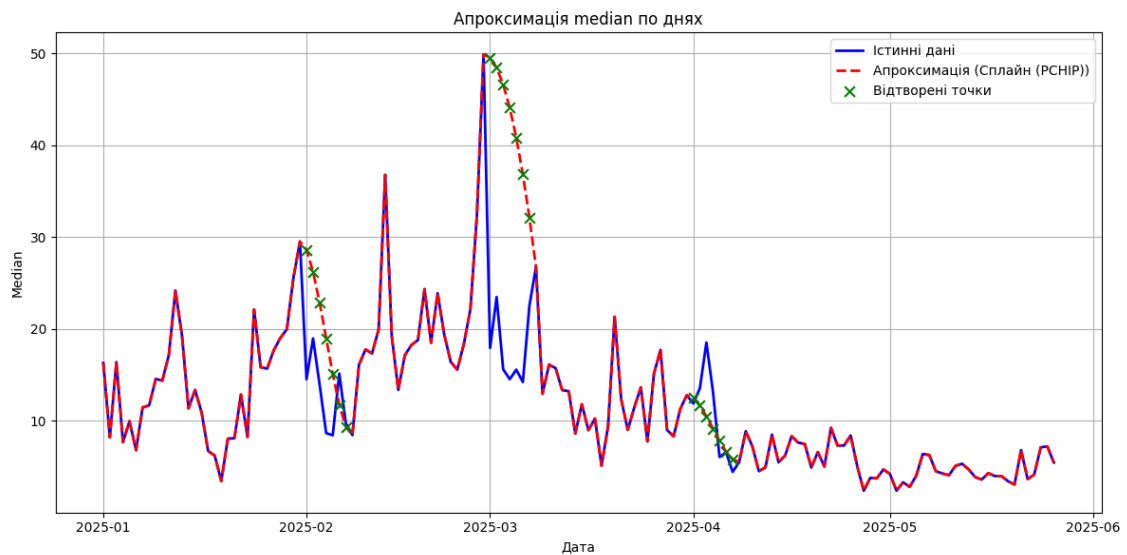


Рисунок 4.8 – Відтворені дані в місті Дніпро

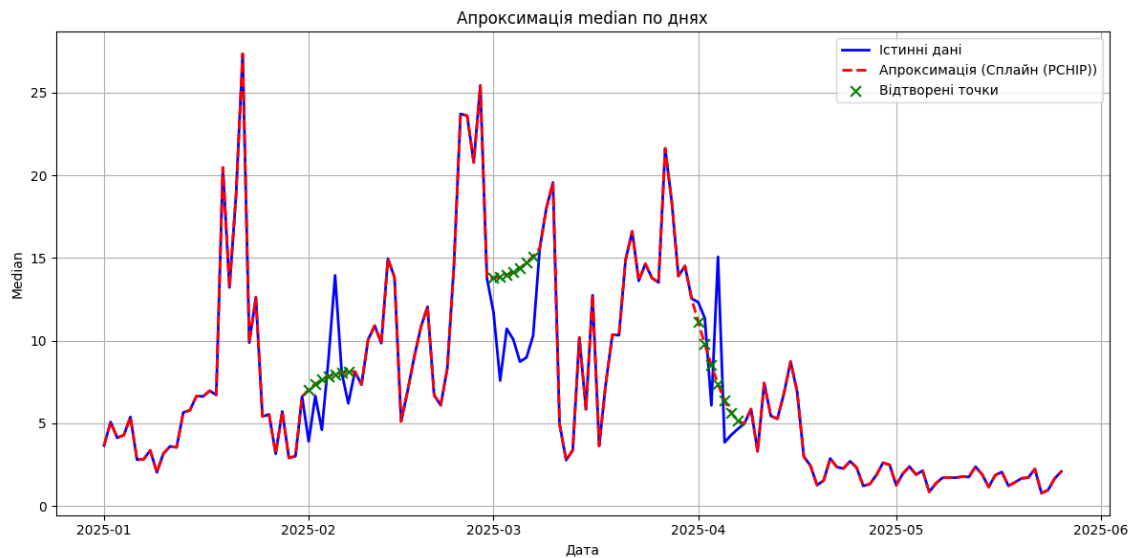


Рисунок 4.9 – Відтворені дані в місті Львів

Наступний метод, який був використан – згладжуючий сплайн (Univariate Spline). Він забезпечує баланс між точністю апроксимації та гладкістю функції, що моделює залежність.

Для підготовки даних дати було перетворено у числовий формат як кількість днів від початкової дати вимірювань

Метод реалізовано з використанням класу `UnivariateSpline` із бібліотеки `SciPy`. Цей клас дозволяє побудувати гладку сплайн-функцію, яка мінімізує суму квадратичних відхилень між апроксимацією та

спостереженнями з додатковим регуляризаційним членом, що контролює ступінь згладжування. Регулюючий параметр s (параметр гладкості) визначає компроміс між точним проходженням крізь усі точки і забезпеченням гладкої кривої, що запобігає надмірному коливанию моделі.

У цій роботі було вибрано параметр згладжування $s = 1.5$, що забезпечує адекватну компромісну апроксимацію вихідних даних з урахуванням їх природної мінливості.

Побудований згладжуючий сплайн використовувався для прогнозування значень на всьому інтервалі дат, включаючи пропущені точки, шляхом обчислення значень функції у відповідних точках. Це дозволяє отримати неперервну, плавну криву, яка добре відображає загальну тенденцію часових змін, зменшуючи вплив шуму і локальних флуктуацій.

На рисунках 4.10, 4.11 та 4.12 представлені графіки з відтвореними точками вимірювань, отримані за допомогою методу «Згладжуючий сплайн».

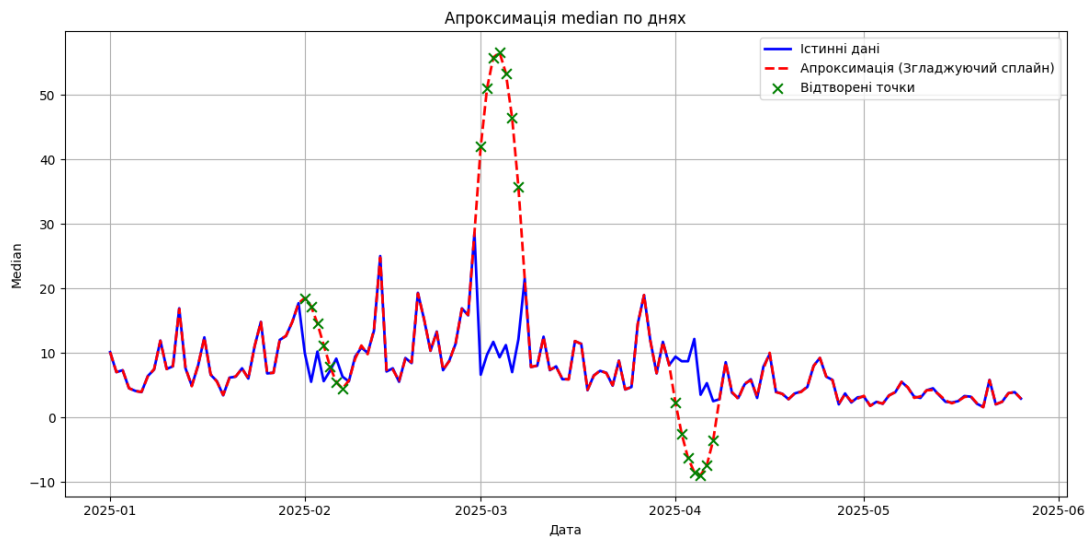


Рисунок 4.10 – Відтворені дані в місті Запоріжжя

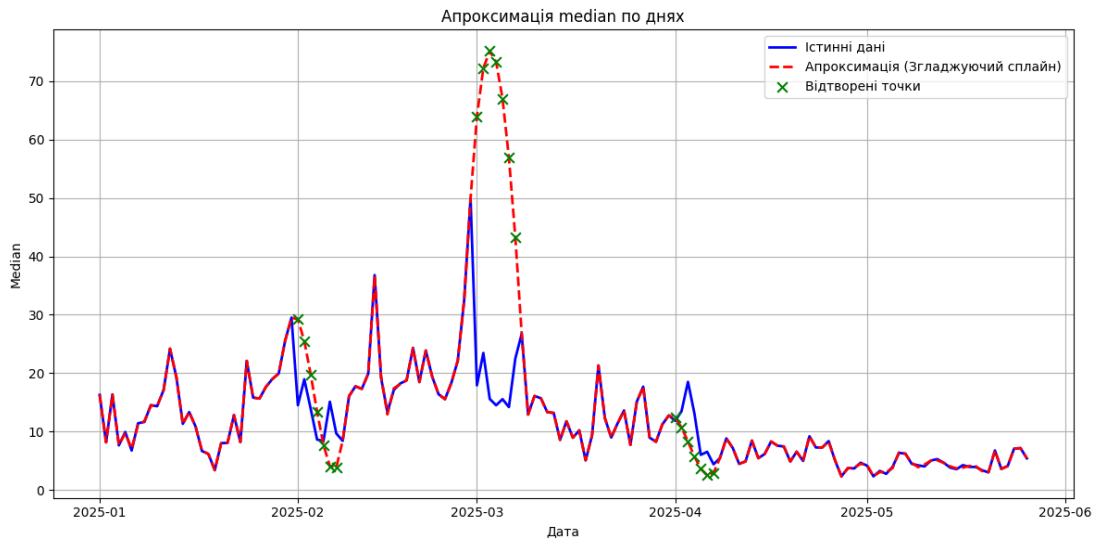


Рисунок 4.11 – Відтворені дані в місті Дніпро

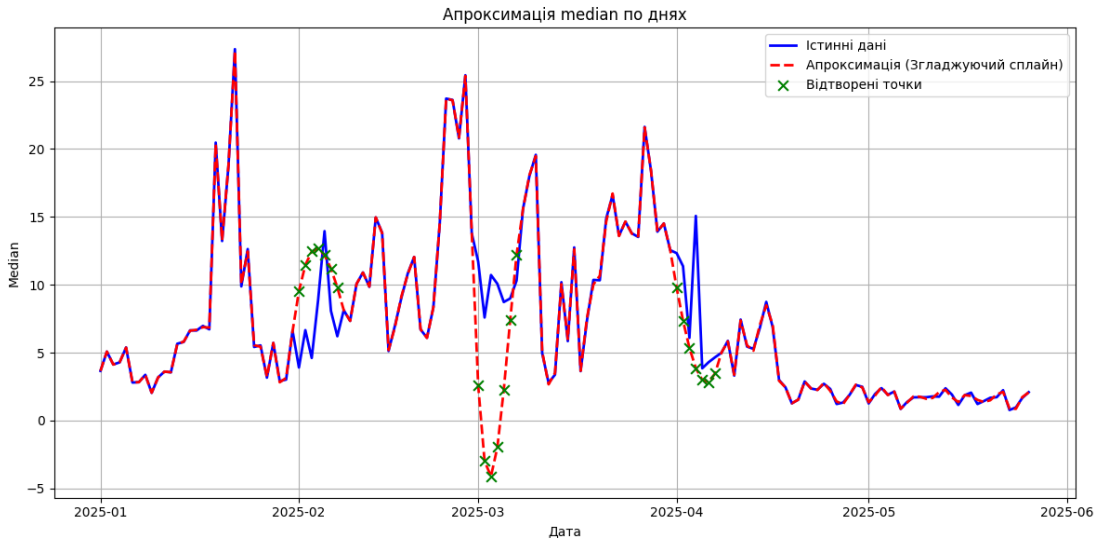


Рисунок 4.12 – Відтворені дані в місті Львів

Для відновлення пропущених значень у часовому ряді медіанних концентрацій PM2.5 застосовано класичну модель авторегресії інтегрованого ковзного середнього — ARIMA. Цей підхід дозволяє моделювати часові ряди з урахуванням як авторегресивної компоненти, так і компоненти ковзного середнього, а також враховує необхідність диференціювання для забезпечення стаціонарності ряду.

Для побудови моделі вихідні дані у форматі (date,median) було перетворено у часовий ряд із часовою індексацією, де індексом виступали дати, а значеннями — відповідні медіанні концентрації.

Параметри моделі були отримані за допомогою методу максимальної правдоподібності через функцію ARIMA із бібліотеки statsmodels. Після підгонки модель використовується для прогнозування значень на всьому діапазоні дат, включаючи пропущені періоди.

Завдяки інтеграції в моделі диференціювання досягається ефективне моделювання нестационарних часових рядів із трендовими компонентами. Використання ARIMA дозволяє враховувати кореляції між поточними та минулими значеннями, а також вплив випадкових збурень, що сприяє більш точному відновленню пропущених значень.

На рисунках 4.13, 4.14 та 4.15 представлені графіки з відтвореними точками вимірювань, отримані за допомогою методу ARIMA.

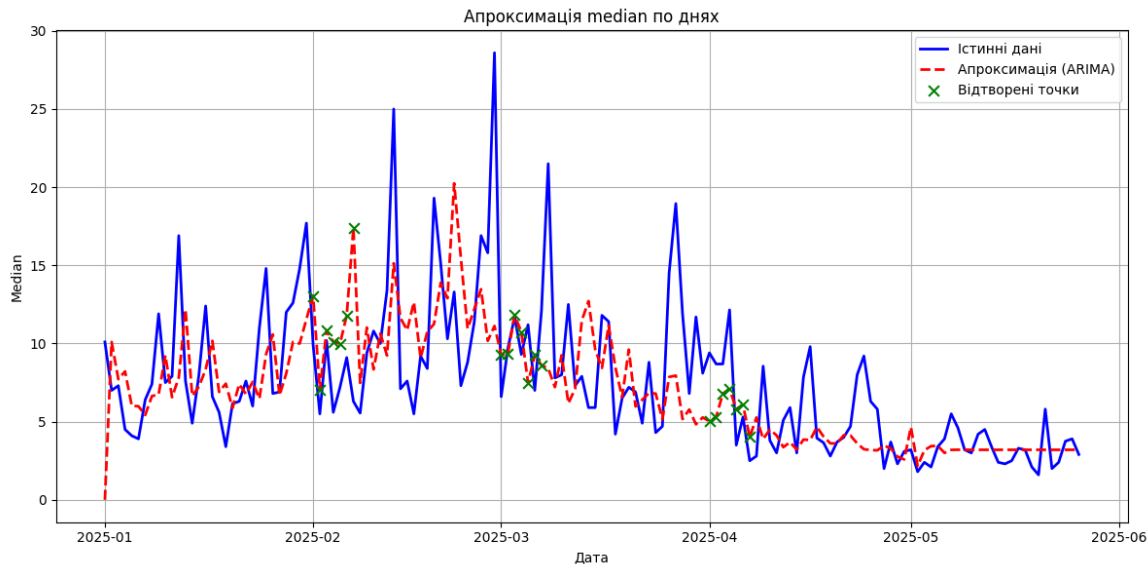


Рисунок 4.13 – Відтворені дані в місті Запоріжжя

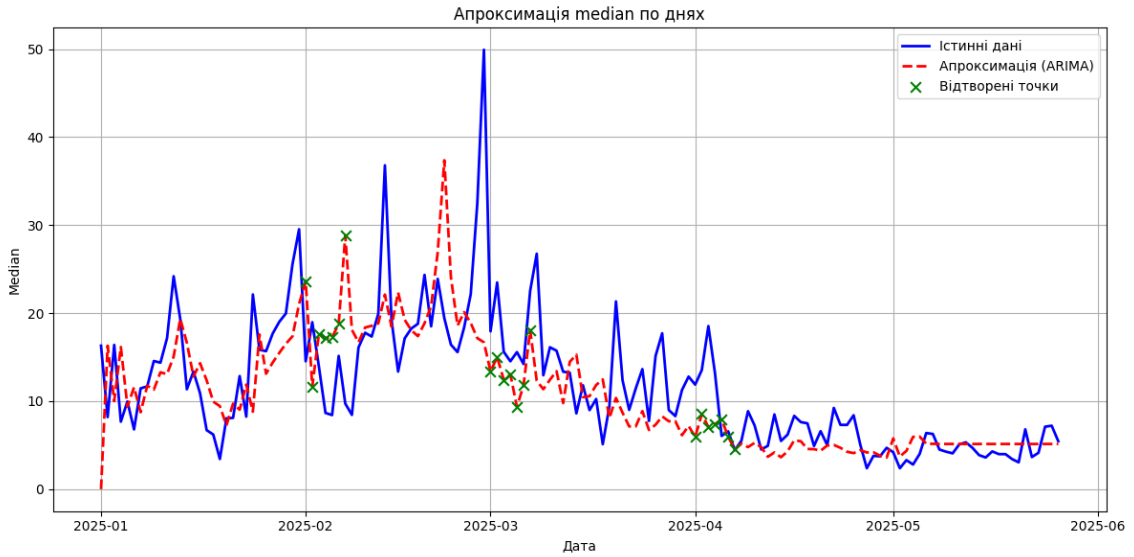


Рисунок 4.14 – Відтворені дані в місті Дніпро

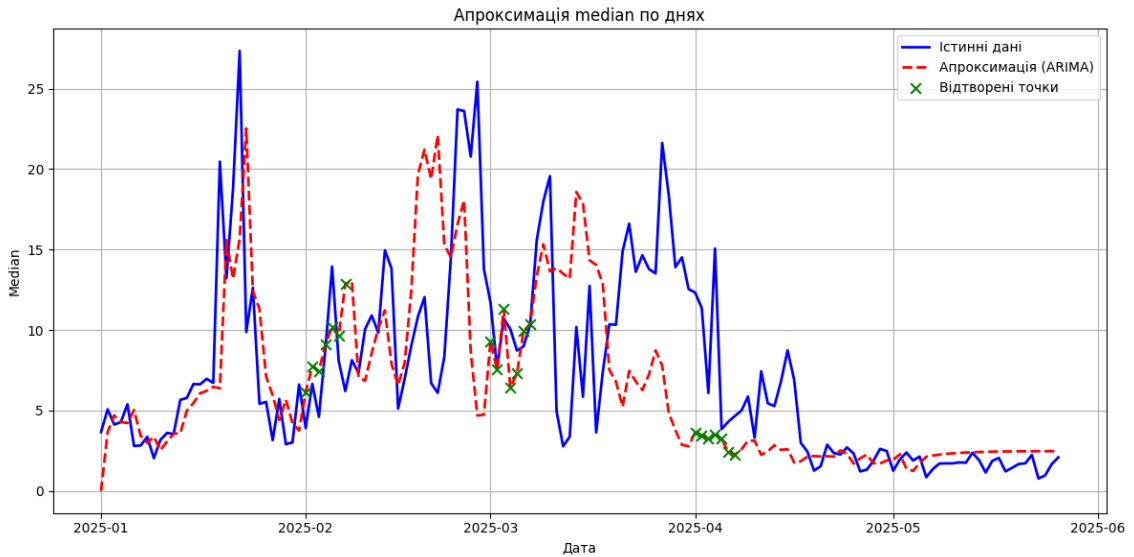


Рисунок 4.15 – Відтворені дані в місті Львів

Наступний використаний метод – SARIMA. Він розширює класичну ARIMA-модель, враховуючи сезонні компоненти у даних.

Модель була реалізована за допомогою класу SARIMAX із бібліотеки statsmodels, який дозволяє ефективно налаштовувати параметри сезонної авторегресії, інтегрування та ковзного середнього. Параметри моделі оцінювалися методом максимальної правдоподібності.

Після побудови моделі вона застосовувалась для прогнозування значень по всьому діапазону спостережень, включно з пропущеними датами,

шляхом предикції рівнів часових рядів. Такий підхід дозволяє враховувати як довгострокові тенденції, так і сезонні коливання, що забезпечує більш точне відновлення пропущених значень у даних із вираженою сезонною структурою.

Застосування SARIMA-моделі забезпечило адекватне моделювання часових рядів із сезонними закономірностями.

На рисунках 4.16, 4.17 та 4.18 представлені графіки з відтвореними точками вимірювань, отримані за допомогою методу SARIMA.

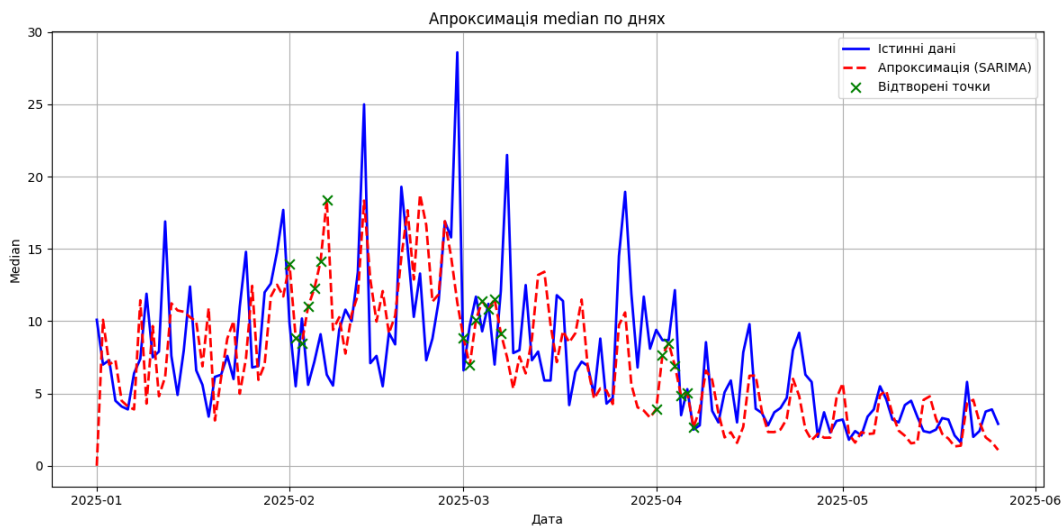


Рисунок 4.16 – Відтворені дані в місті Запоріжжя

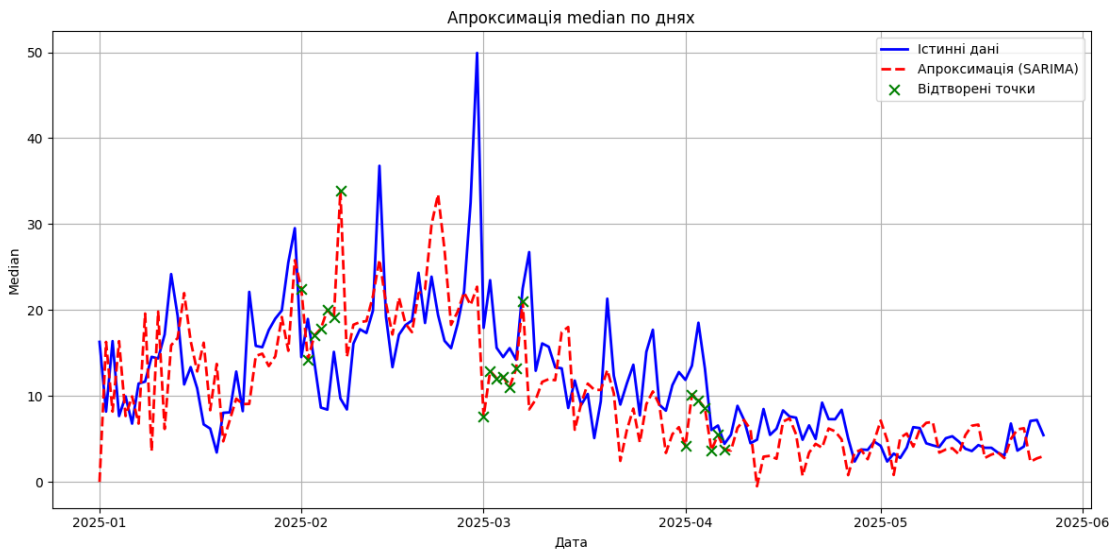


Рисунок 4.17 – Відтворені дані в місті Дніпро

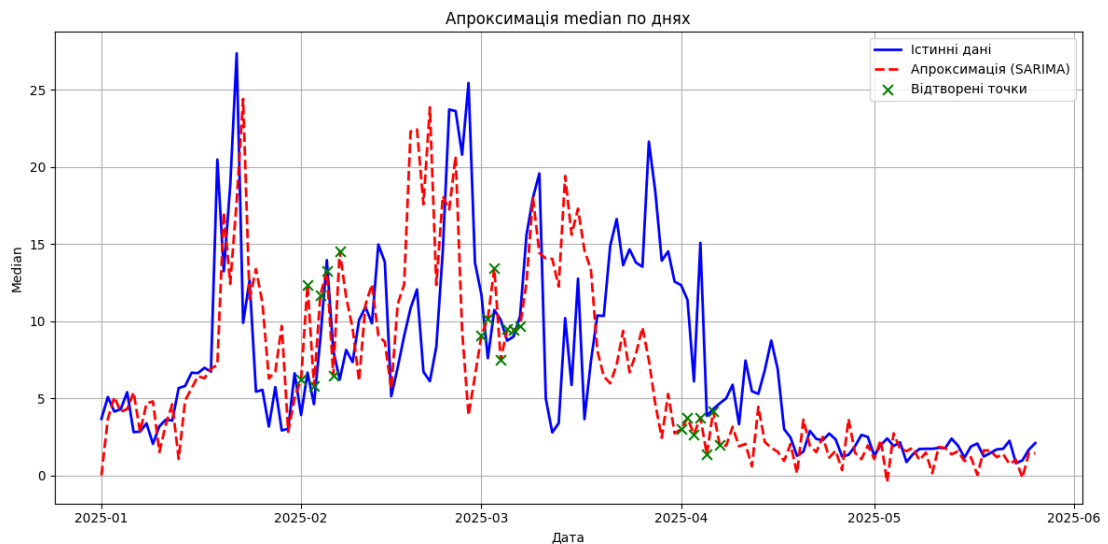


Рисунок 4.18 – Відтворені дані в місті Львів

Також застосовано ансамблевий метод машинного навчання — Random Forest Regressor, який є реалізацією методу випадкового лісу для задач регресії.

Модель RandomForestRegressor із бібліотеки sklearn.ensemble була навчена на повних (наявних) значеннях вибірки. В результаті побудовано ансамбль незалежних рішень (дерев регресії), що базуються на випадкових підвибірках даних та ознак. Остаточне прогнозне значення для кожного дня визначалося як середнє значення по всіх деревах ансамблю, що суттєво зменшує варіативність і запобігає перенавчанню, властивому окремим деревам.

Модель застосовувалась для прогнозування значень по всьому часовому інтервалу, зокрема для дат, для яких були відсутні спостереження. Отримані значення для пропущених днів були збережені та використані для подальшого аналізу.

На рисунках 4.19, 4.20 та 4.21 представлені графіки з відтвореними точками вимірювань, отримані за допомогою методу Random Forest.

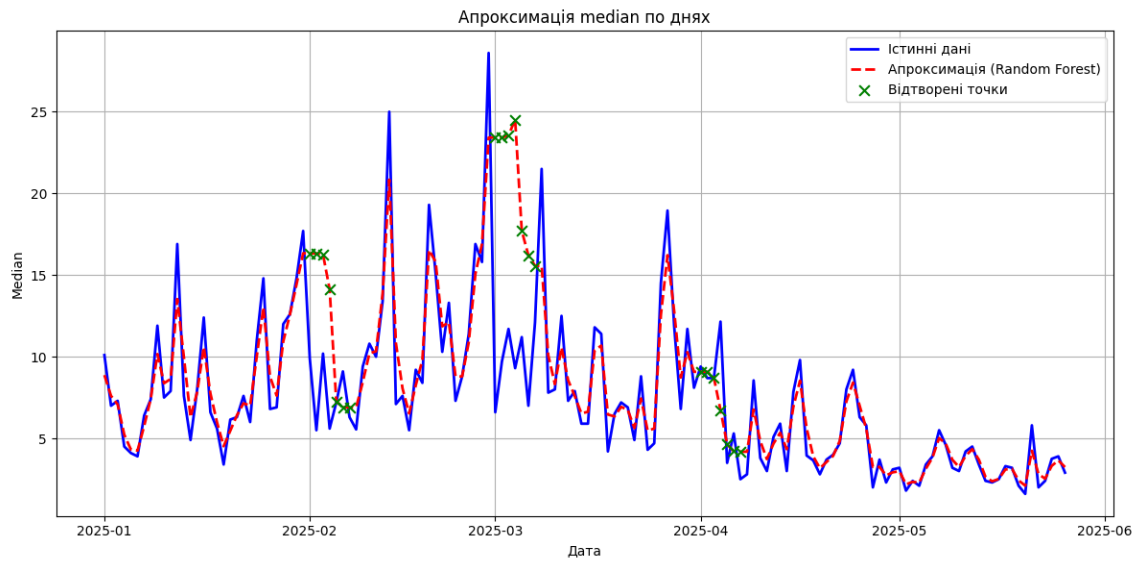


Рисунок 4.19 – Відтворені дані в місті Запоріжжя

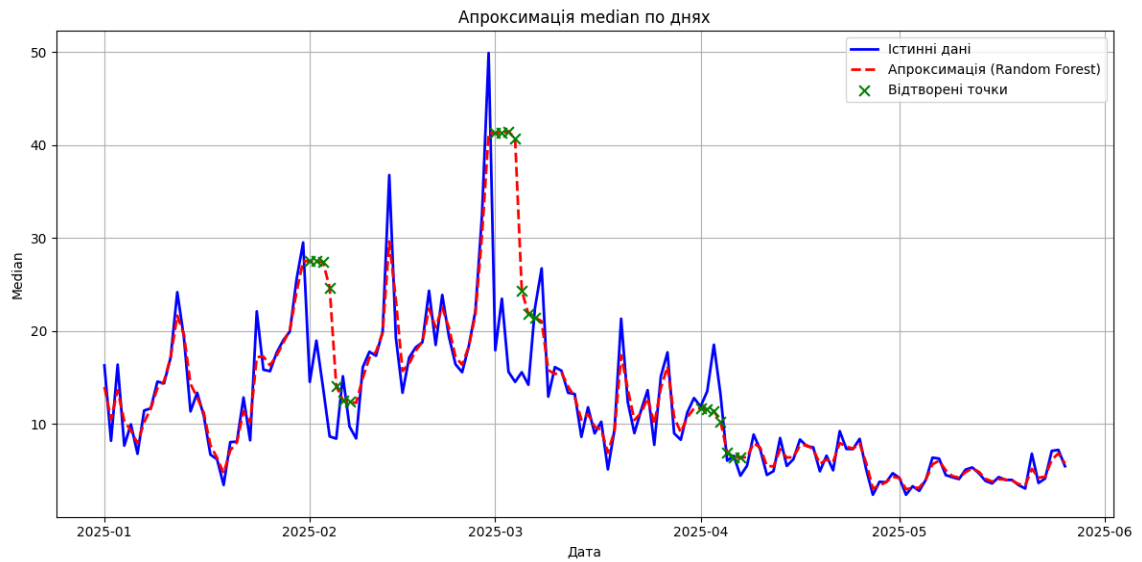


Рисунок 4.20 – Відтворені дані в місті Дніпро

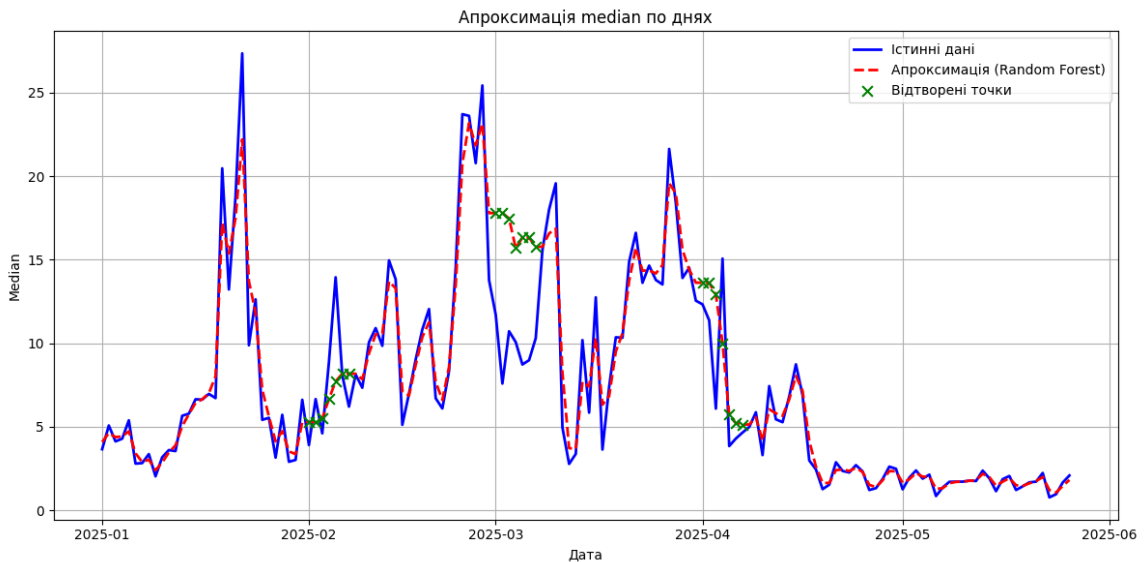


Рисунок 4.21 – Відтворені дані в місті Львів

4.4.2 Оцінка точності відновлених результатів

У процесі дослідження та порівняння ефективності різних методів відновлення пропущених значень часових рядів медіанних концентрацій $PM_{2.5}$, особлива увага приділялася кількісній оцінці точності отриманих результатів. З цією метою у роботі було застосовано середньоквадратичну помилку як ключову метрику якості апроксимації.

RMSE обчислювалась для кожного з розглянутих методів, зокрема «Поліноміальної регресії», сплайнів (PCHIP і згладжуючого сплайну), ARIMA, SARIMA та Random Forest, шляхом порівняння відновлених значень для пропущених днів із відповідними істинними значеннями, за умови їх наявності у тестовому наборі даних. Для цього попередньо було завантажено окремий файл з повними (неушкодженими) даними, які виступали як еталон при оцінюванні точності [16].

Обчислення RMSE виконувалося за формулою (4.1):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}, \quad (4.1)$$

де $\hat{y}_i$ – значення, відновлене обраним методом для i -го дня з пропущеними даними;

y_i – відповідне істинне значення;

n – загальна кількість пропущених точок.

Застосування саме RMSE як метрики дозволяє враховувати не лише середню похибку, але й її варіацію, надаючи більшу вагу суттєвим відхиленням від істинних значень. Такий підхід забезпечує об'єктивне порівняння точності різних моделей та дозволяє виявити методи, що краще справляються із завданням реконструкції часових рядів у присутності пропусків.

Результати обчислення RMSE були представлені у вигляді числових значень у графічному інтерфейсі користувача, що дозволило здійснити комплексний аналіз точності та стабільності кожного з методів. Результати вимірювань середньоквадратичної помилки можна побачити в таблиці 4.1.

Таблиця 4.1 – Результати обчислення середньоквадратичної помилки для кожного з використаних методів відновлення у розрізі міст

Метод	Запоріжжя	Дніпро	Львів
Поліноміальна регресія	3.3435	5.4449	4.1081
Сплайн (PCHIP)	10.7069	15.8703	3.7568
Згладжуючий сплайн	24.3721	28.4775	6.6059
ARIMA	3.7341	7.1642	4.2926
SARIMA	4.1790	8.0255	4.6167
Random Forest	7.8399	12.2426	4.8987

Результати, наведені в таблиці 3.1, демонструють, що Поліноміальна регресія забезпечила найменші значення RMSE у більшості випадків (зокрема, для Запоріжжя та Львова), що свідчить про її високу здатність до апроксимації тренду в умовах обмеженої складності даних. Метод ARIMA також показав конкурентні результати, особливо для міста Запоріжжя, де

RMSE становила 3.7341, що є другим за точністю показником після поліноміальної регресії.

У той же час Згладжуючий сплайн виявився менш ефективним для міст Запоріжжя та Дніпра, продемонструвавши найвищі значення RMSE (24.3721 та 28.4775 відповідно), що свідчить про його чутливість до коливань та можливе перенавчання на наявних даних. Метод Сплайн (PCHIP) показав посередні результати — відносно високу похибку для Запоріжжя і Дніпра, проте досить низьку RMSE для Львова (3.7568), що може свідчити про придатність цього методу для рядів з певною структурою.

Методи, побудовані на основі ансамблевих моделей — зокрема, Random Forest — продемонстрували стабільні, але не найкращі результати, маючи RMSE у межах 7.8399–12.2426. Така поведінка моделі може бути пов'язана з її інерційністю до дрібномасштабних змін та відсутністю механізму врахування часової динаміки.

SARIMA показала себе як надійний метод для моделювання сезонних коливань, зокрема в місті Львів, де RMSE склала лише 4.6167, однак у Дніпрі похибка була дещо вищою (8.0255), що свідчить про складність узгодження сезонної структури в даному регіоні.

Загалом, отримані результати дозволяють зробити висновок, що вибір методу для заповнення пропусків у часових рядах повинен враховувати специфіку локальних даних: ступінь коливань, наявність сезонності, а також тривалість пропущених інтервалів. У контексті аналізованих даних найбільш універсальним та ефективним методом виявилася Поліноміальна регресія, яка забезпечила найкращий компроміс між точністю та стійкістю до локальних змін.

ВИСНОВКИ

У межах проведеного дослідження було здійснено глибокий аналіз та порівняння ефективності різних методів відновлення пропущених значень у часових рядах концентрацій твердих частинок PM_{2.5}, що є критично важливим аспектом екологічного моніторингу. Завдяки використанню реальних даних із трьох міст України (Запоріжжя, Дніпро, Львів) вдалося досягти високої репрезентативності результатів, що забезпечує релевантність висновків для практичного застосування.

Використання таких методів, як поліноміальна регресія, інтерполяція сплайнами, ARIMA, SARIMA та Random Forest, дало змогу отримати комплексну оцінку їхньої ефективності. За результатами розрахунків середньоквадратичної помилки (RMSE), найкращі показники точності продемонстрував метод поліноміальної регресії третього ступеня. Це дає підстави вважати його найбільш доцільним і практичним для задач відновлення даних у даній галузі.

Практичним результатом дослідження стало створення спеціалізованого програмного забезпечення для Windows, яке поєднує зручність графічного інтерфейсу та потужні інструменти аналізу. Застосунок дозволяє обирати метод відновлення, обробляти користувацькі набори даних, візуалізувати результати та експортувати їх для подальшого використання. Розробка реалізована мовою Python із використанням сучасних бібліотек, що гарантує гнучкість і масштабованість рішення.

Загалом, результати дослідження підтверджують важливість якісного заповнення пропущених значень у часових рядах, зокрема через:

- підвищення точності прогнозів;
- точніше виявлення трендів і сезонності;
- покращену інтерпретацію результатів;
- розширення можливостей для застосування сучасних методів машинного навчання.

Таким чином, представлений підхід має не лише теоретичну цінність, а й вагоме практичне значення для фахівців у галузі екологічного моніторингу, аналітики даних та розробки програмного забезпечення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гринчишин Н. М. Якість атмосферного повітря за вмістом твердих мікрочастинок ($pm_{2,5}$) у містах України в умовах карантину та воєнного стану / Н. М. Гринчишин // Вісник Львівського державного університету безпеки життєдіяльності. – Львів, 2023. – Вип. №27. – С. 6-15.
2. Вплив забруднення повітря на здоров'я людини (за даними 29 грудня 2024). Донецький обласний ЦКПХ [Електрон. ресурс]. – Режим доступу : <https://dn.cdc.gov.ua/article/vplyv-zabrudnennya-povitrya-na-zdorovya-lyudyny/>
3. Бочкова С. Забруднення повітря в Україні – погляд з космосу. Дослідження за результатами аналізу супутникових знімків Copernicus Sentinel 5p та даних про забруднення повітря з контрольованою якістю Служби моніторингу атмосфери Copernicus / С. Бочкова, Р. Боговіч, М. Гринчар ; ред. Я. Лобохі. – Прага-Київ: World from space, 2020. – 38 с.
4. Адаменко С. Я. Дослідження закономірностей змін $pm_{2,5}$ та pm_{10} в атмосферному повітрі Прикарпаття / С. Я. Адаменко, Л. М. Архипова // Екологічна безпека та природокористування. – Івано-Франківськ, 2024. – Вип. 3 (51). – С. 47-58.
5. За останні два роки Київ вибудував найбільшу в Україні систему моніторингу якості повітря (+фото, звіт) [Електрон. ресурс]. – Режим доступу : https://kyivcity.gov.ua/news/za_ostanni_dva_roki_kiv_vibuduvav_naybilshu_v_ukrani_sistemu_monitoringu_yakosti_povitrya/
6. Гурєєва К. М. Огляд методів машинного навчання в задачі прогнозування фінансових часових рядів / К. М. Гурєєва, О. В. Кудін, А. О. Лісняк // Computer Science and Applied Mathematics, (2). – Запоріжжя, 2018. – С. 18-28.
7. Тищенко С. І. Моделювання впливу цифрових загроз на фінансові ринки за допомогою аналізу часових рядів та виявлення аномалій засобами Python / Тищенко С. І., Пархоменко О. Ю., Хилько І. І. // Modern

Economics. – Миколаїв, 2024. – Вип. № 44. – С. 205-212.

8. Камінський Р. М. Відновлення пропусків в результатах тестування та ідентифікації операторського персоналу / Камінський Р. М., Кунанець Н. Е., Пасічник В. В. // Вісник національного університету «Львівська політехніка», кафедри інформаційних систем та мереж. – Львів, 2018. – С. 92-104.

9. Єменець К. Методи прогнозування часових рядів з вираженою сезонністю на основі трансформерів. / Єменець К. // Вісник Хмельницького національного університету, технічні науки. – Хмельницький, 2014. – Вип. 333(2). – С. 131-134.

10. Основні забруднювачі атмосферного повітря : характеристики, вплив на організм людини. Товариство Автоекотрилад. [Електрон. ресурс]. – Режим доступу : <https://eco.aep.kiev.ua/novini/osnovni-zabrudnyuvachi-atmosfernogo-povitrya-harakteritstiki-vpliv-na-organizm-lyudini/>

11. Нестор А. А. Сучасний екологічний стан навколишнього середовища України. / Нестор А. А. // Комплексне використання ресурсів довкілля: збірник матеріалів II Всеукраїнської науково-практичної конференції (Дрогобич, 20 лист. 2024 р.). – Дрогобич, 2024. – С. 131-136.

12. Національна доповідь про стан навколишнього природного середовища в Україні у 2020 році [Електрон. ресурс]. – Режим доступу : <https://mepr.gov.ua/wp-content/uploads/2022/10/Natsionalna-Dopovid-2020-2.pdf>

13. Контроль засміченості повітря: інноваційні детектори запиленості [Електрон. ресурс]. – Режим доступу : <https://simvolt.ua/kontrol-zasmichenosti-povitria-innovatsiini-detektory-zapylenosti/>

14. Шкала індексу якості повітря та кольорова легенда [Електрон. ресурс]. – Режим доступу : <https://aqicn.org/scale/uk>

15. Відкриваються нові можливості для контролю якості повітря. Національний університет біоресурсів і природознавства України (за даними статистики 13 лип. 2020 р.) [Електрон. ресурс]. – Режим доступу : <https://nubip.edu.ua/node/78991>

16. Класифікація похибок вимірювань [Електрон. ресурс]. – Режим доступу : <https://disted.edu.vn.ua/courses/learn/2604>
17. Podkovalihina, O. (2025) Multi-Agent System for Analysis and Prediction of PM2.5 Levels: Software Engineering and Data Smoothing Methods [Електрон. ресурс]. – Режим доступу : [https://doi.org/10.52058/2786-6025-2025-4\(45\)-1031-1044](https://doi.org/10.52058/2786-6025-2025-4(45)-1031-1044)
18. Забруднення повітря в Dmytra Yavornytskoho Avenue, Dnipro, Ukraine [Електрон. ресурс]. – Режим доступу : <https://aqicn.org/station/ukraine-dnipro-dmytra-yavornytskoho-avenue/uk/>

ДОДАТОК А
Текст програми

A.1 Імпорт необхідних бібліотек

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from scipy.interpolate import PchipInterpolator, UnivariateSpline
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
import tkinter as tk
from tkinter import filedialog, messagebox, ttk
```

A.2 Реалізація функції обробки файлу

```
latest_reconstructed_df = None
latest_figure = None
true_df = None

def process_file(file_path, method, text_widget):
    global latest_reconstructed_df, latest_figure, true_df

    if not file_path.endswith(".csv"):
        messagebox.showerror("Помилка", "Неправильно вибраний файл.  
Виберіть CSV.")
        return

    try:
        df = pd.read_csv(file_path)
    except Exception as e:
        messagebox.showerror("Помилка читання", f"Не вдалося зчитати  
файл: {e}")
        return

    if "date" not in df.columns or "median" not in df.columns:
        messagebox.showerror("Помилка формату", "CSV повинен містити  
стовпці 'date' та 'median'")
        return
```

A.3 Реалізація функції обробки та підготовки вхідних даних

```

df["date"] = pd.to_datetime(df["date"])
df = df.sort_values("date")
df["days"] = (df["date"] - df["date"].min()).dt.days

X = df["days"].values
y = df["median"].values

all_days = np.arange(X.min(), X.max() + 1)
all_dates = df["date"].min() + pd.to_timedelta(all_days, unit='d')
existing_days = pd.Series(X)
missing_days = np.setdiff1d(all_days, existing_days)

y_pred = None
missing_y_pred = None
method_label = method

```

A.4 Метод «Поліноміальна регресія»

```

if method == "Поліноміальна регресія":
    degree = 3
    poly = PolynomialFeatures(degree)
    X_poly = poly.fit_transform(X.reshape(-1, 1))
    model = LinearRegression()
    model.fit(X_poly, y)
    y_pred = model.predict(poly.transform(all_days.reshape(-1, 1)))
    if len(missing_days) > 0:
        missing_y_pred =
model.predict(poly.transform(missing_days.reshape(-1, 1)))

```

A.5 Метод «Сплайн(PCHIP)»

```

elif method == "Сплайн (PCHIP)":
    interpolator = PchipInterpolator(X, y)
    y_pred = interpolator(all_days)
    if len(missing_days) > 0:
        missing_y_pred = interpolator(missing_days)

```

A.6 Метод «Згладжующий сплайн»

```
elif method == "Згладжующий сплайн":
    spline = UnivariateSpline(X, y, s=1.5)
    y_pred = spline(all_days)
    if len(missing_days) > 0:
        missing_y_pred = spline(missing_days)
```

A.7 Метод ARIMA

```
elif method == "ARIMA":
    y_series = pd.Series(y, index=pd.date_range(start=df["date"].min(),
periods=len(y)))
    model = ARIMA(y_series, order=(2, 1, 2)).fit()
    all_dates = pd.date_range(start=df["date"].min(),
end=df["date"].max())
    forecast = model.predict(start=all_dates[0], end=all_dates[-1])
    y_pred = forecast.values

    if len(missing_days) > 0:
        missing_dates = df["date"].min() + pd.to_timedelta(missing_days,
unit='d')
        forecast_missing = model.predict(start=missing_dates.min(),
end=missing_dates.max(), typ='levels')
        missing_y_pred = forecast_missing[missing_dates].values
```

A.8 Метод SARIMA

```
elif method == "SARIMA":
    try:
        y_series = pd.Series(y, index=pd.date_range(start=df["date"].min(),
periods=len(y)))
        model = SARIMAX(y_series, order=(1, 1, 1), seasonal_order=(1, 1,
1, 7)).fit(dispatch=False)
        all_dates = pd.date_range(start=df["date"].min(),
end=df["date"].max())
        forecast = model.predict(start=all_dates[0], end=all_dates[-1])
        y_pred = forecast.values

        if len(missing_days) > 0:
```

```

        missing_dates = df["date"].min() +
pd.to_timedelta(missing_days, unit='d')
        forecast_missing = model.predict(start=missing_dates.min(),
end=missing_dates.max(), typ='levels')
        missing_y_pred = forecast_missing[missing_dates].values
    except Exception as e:
        messagebox.showerror("SARIMA помилка", f"SARIMA не
вдалося побудувати модель: {e}")
    return

```

A.9 Метод Random Forest

```

elif method == "Random Forest":
    model = RandomForestRegressor()
    model.fit(X.reshape(-1, 1), y)
    y_pred = model.predict(all_days.reshape(-1, 1))
    if len(missing_days) > 0:
        missing_y_pred = model.predict(missing_days.reshape(-1, 1))

    else:
        messagebox.showerror("Помилка", f"Невідомий метод: {method}")
    return

```

A.10 Реалізація побудови таблиці відновлених значень та обчислення RMSE

```

text_widget.delete(1.0, tk.END)

    if len(missing_days) > 0:
        missing_dates = df["date"].min() + pd.to_timedelta(missing_days,
unit='d')
        reconstructed_df = pd.DataFrame({
            "date": missing_dates,
            "days": missing_days,
            "median": missing_y_pred
        })

        text_widget.insert(tk.END, "Відтворені значення для відсутніх
днів:\n\n")
        formatted = reconstructed_df.copy()
        formatted["date"] = formatted["date"].dt.strftime("%Y-%m-%d")

```

```

formatted["median"] = formatted["median"].map("{:.3f}".format)
text_widget.insert(tk.END, formatted.to_string(index=False))

if true_df is not None:
    true_df_copy = true_df.copy()
    true_df_copy["days"] = (true_df_copy["date"] -
df["date"].min()).dt.days
    merged = pd.merge(reconstructed_df, true_df_copy, on="days",
suffixes=("_reconstructed", "_true"))
    if not merged.empty:
        mse = np.mean((merged["median_reconstructed"] -
merged["median_true"]) ** 2)
        rmse = np.sqrt(mse)
        text_widget.insert(tk.END, f"\n\nСередньоквадратична
помилка (RMSE): {rmse:.4f}")
    else:
        text_widget.insert(tk.END, "\n\nІстинні значення для
пропущених днів не знайдено.")
    else:
        text_widget.insert(tk.END, "Всі дні присутні — відсутні значення
не виявлені.")
    reconstructed_df = pd.DataFrame()

latest_reconstructed_df = reconstructed_df

```

A.11 Реалізація побудови графіка

```

fig, ax = plt.subplots(figsize=(12, 6))
if true_df is not None:
    ax.plot(true_df["date"], true_df["median"], label="Істинні дані",
color='blue', linewidth=2)
    ax.plot(all_dates, y_pred, label=f"Апроксимація ({method_label})",
color='red', linestyle='--', linewidth=2)
    if not reconstructed_df.empty:
        ax.scatter(reconstructed_df["date"], reconstructed_df["median"],
color='green', label="Відтворені точки", marker='x', s=60, zorder=5)

    ax.set_title("Апроксимація median по днях")
    ax.set_xlabel("Дата")
    ax.set_ylabel("Median")
    ax.grid(True)
    ax.legend()
    fig.tight_layout()

```

```
plt.show()  
latest_figure = fig
```

A.12 Реалізація функції завантаження файлів

```
def browse_file():
    filename = filedialog.askopenfilename(filetypes=[("CSV files",
    "*.csv")])
    if filename:
        file_path_var.set(filename)

def browse_true_file():
    global true_df
    filename = filedialog.askopenfilename(filetypes=[("CSV files",
    "*.csv")])
    if filename:
        try:
            df = pd.read_csv(filename)
            if "date" in df.columns and "median" in df.columns:
                df["date"] = pd.to_datetime(df["date"])
                true_df = df.sort_values("date")
                true_file_label.config(text=f"Істинні дані завантажено")
            else:
                messagebox.showerror("Формат", "CSV повинен містити
                стовпці 'date' та 'median'")
        except Exception as e:
            messagebox.showerror("Помилка читання", f"Не вдалося
            зчитати файл: {e}")
```

A.13 Реалізація кнопки обробки файлів

```
def run_processing():
    file_path = file_path_var.get()
    method = method_var.get()
    if not file_path:
        messagebox.showwarning("Попередження", "Ви не вибрали файл з
        даними")
    else:
        process_file(file_path, method, text_output)
```

A.14 Реалізація збереження CSV-файлу

```
def save_reconstructed_csv():
    global latest_reconstructed_df
```

```

        if latest_reconstructed_df is not None and not
latest_reconstructed_df.empty:
            file_path = filedialog.asksaveasfilename(defaultextension=".csv",
filetypes=[("CSV files", "*.csv")])
            if file_path:
                latest_reconstructed_df.to_csv(file_path, index=False)
                messagebox.showinfo("Успіх", f"Файл збережено: {file_path}")
            else:
                messagebox.showwarning("Немає даних", "Немає відновлених
даних для збереження")

```

A.15 Реалізація збереження графіка

```

def save_figure():
    global latest_figure
    if latest_figure is not None:
        file_path = filedialog.asksaveasfilename(defaultextension=".png",
filetypes=[("PNG files", "*.png")])
        if file_path:
            latest_figure.savefig(file_path)
            messagebox.showinfo("Успіх", f"Графік збережено: {file_path}")
        else:
            messagebox.showwarning("Немає графіка", "Немає графіка для
збереження")

```

A.16 Побудова GUI та запуск програми

```

root = tk.Tk()
root.title("Відновлення даних PM2.5")

file_path_var = tk.StringVar()
method_var = tk.StringVar(value="Поліноміальна регресія")

frame = tk.Frame(root, padx=10, pady=10)
frame.pack()

btn_browse = tk.Button(frame, text="Вибрати CSV-файл з пропусками",
command=browse_file)
btn_browse.grid(row=0, column=0, padx=5, pady=5)

```

```

entry_file = tk.Entry(frame, textvariable=file_path_var, width=50,
state='readonly')
entry_file.grid(row=0, column=1, padx=5, pady=5)

btn_true = tk.Button(frame, text="Вибрати істинний CSV",
command=browse_true_file, bg='lightblue')
btn_true.grid(row=1, column=0, padx=5, pady=5)

true_file_label = tk.Label(frame, text="Істинні дані не завантажено",
fg='gray')
true_file_label.grid(row=1, column=1, padx=5, pady=5, sticky='w')

label_method = tk.Label(frame, text="Метод відновлення:")
label_method.grid(row=2, column=0, padx=5, pady=5)

method_menu = ttk.Combobox(frame, textvariable=method_var,
state="readonly", width=47)
method_menu['values'] = [
    "Поліноміальна регресія",
    "Сплайн (PCHIP)",
    "Згладжуючий сплайн",
    "ARIMA",
    "SARIMA",
    "Random Forest",
]
method_menu.grid(row=2, column=1, padx=5, pady=5)

btn_run = tk.Button(frame, text="Відновити дані",
command=run_processing, bg='lightgreen')
btn_run.grid(row=3, column=0, columnspan=2, pady=10)

text_output = tk.Text(root, height=12, width=80, wrap='none',
font=('Courier New', 10))
text_output.pack(padx=10, pady=5)

save_buttons_frame = tk.Frame(root)
save_buttons_frame.pack(pady=10)

btn_save_csv = tk.Button(save_buttons_frame, text="Зберегти CSV",
command=save_reconstructed_csv, width=20)
btn_save_csv.grid(row=0, column=0, padx=20)

btn_save_plot = tk.Button(save_buttons_frame, text="Зберегти графік",
command=save_figure, width=20)
btn_save_plot.grid(row=0, column=1, padx=20)

```

```
root.mainloop()
```

ДОДАТОК Б
Слайди презентації

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЗАПОРІЗЬКА ПОЛІТЕХНІКА"
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК І ТЕХНОЛОГІЙ
КАФЕДРА ПРОГРАМНИХ ЗАСОБІВ

Дипломна робота на тему:
ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ВІДНОВЛЕННЯ ВІДСУТНІХ ДАНИХ МОНІТОРИНГУ ЯКОСТІ
ПОВІТРЯ НА ОСНОВІ ДАНИХ КОНЦЕНТРАЦІЇ РМ_{2.5}

Виконав:

студент групи КНТ - 22Г

Керівник роботи: доцент

Кирило КОЛОСОВ

Олена ПОДКОВАЛІХІНА

2025

Рисунок Б.1 – Слайд 1 «Титульний слайд»

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА РОБОТИ

Об'єкт дослідження: процес обробки та аналізу даних моніторингу якості повітря.

Предмет дослідження: методи та алгоритми машинного навчання та статистичного аналізу для відновлення пропущених даних у часових рядах концентрації РМ_{2.5}.

Мета роботи: дослідження та порівняльний аналіз методів відновлення пропущених даних у часових рядах концентрації РМ_{2.5} з метою підвищення точності відновлення

2

Рисунок Б.2 – Слайд 2 «Об'єкт, предмет та мета роботи»

Актуальність та постановка задачі

Актуальність відновлення відсутніх значень полягає в забезпеченні цілісності та надійності екологічних даних, необхідних для точного аналізу, прогнозування рівнів забруднення повітря та своєчасного інформування населення.

Постановка задачі:

- Дані: часові ряди PM_{2.5} з 3 міст: Запоріжжя, Дніпро та Львів.
- Методи: 6 алгоритмів (Поліноміальна регресія, згладжуючий сплайн, сплайн (PCHIP), ARIMA, SARIMA, RandomForest)
- Критерій: мінімізація RMSE.
- Результат: порівняння ефективності методів для кожного міста.

3

Рисунок Б.3 – Слайд 3 «Актуальність та постановка задачі»

Вибір мови програмування

Критерій	C#	Java	Python
Продуктивність і оптимізація	Висока	Середня	Середня
Простота вивчення	Середня	Середня	Висока
Наявність бібліотек для ML/статистики	Обмежена	Обмежена	Наявні
Побудова GUI (інтерфейсу)	Через WinForms / WPF	Через JavaFX / Swing	Tkinter – просто і швидко
Підтримка моделей ARIMA/SARIMA	Складна	Складна	Наявна, через statsmodels
Інтеграція з matplotlib	Через сторонні засоби	Через сторонні засоби	Так
Гнучкість при роботі з CSV	Обмежена	Середня	Pandas – стандарт де-факто
Сумісність з машинним навчанням (Random Forest)	Можлива, але громіздка	Складна	Так (sklearn)
Швидкість розробки	Середня	Середня	Висока
Автоматичне керування пам'яттю	Так	Так	Так

4

Рисунок Б.4 – Слайд 4 «Вибір мови програмування»

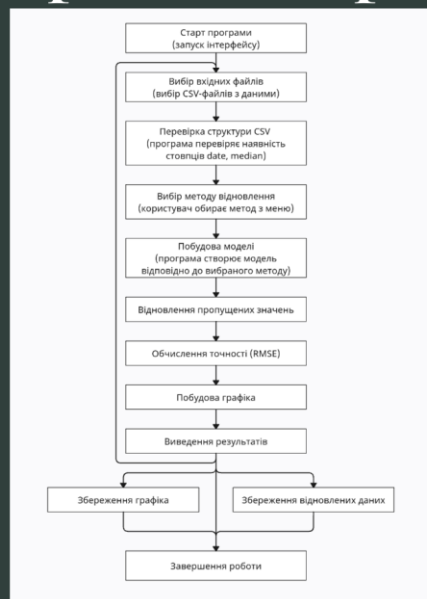
Вибір методів для відновлення даних

Метод	Суть методу	Переваги використання
Поліноміальна регресія	Апроксимація даних певного ступеня	- Швидкість обчислення - Простота інтерпретації
PCHIP (Сплайн)	Кусково-кубічна інтерполяція, що зберігає монотонність даних	- Уникає «перевищення» значень - Ефективний для даних з різкими коливаннями - Зберігає фізичну достовірність
Згладжуючий сплайн	Апроксимація з урахуванням балансу між точністю та гладкістю	- Гнучкість (регулюється параметром згладжування) - Підходить для даних з високою варіативністю
ARIMA	Моделювання часових рядів з урахуванням авторегресії	- Враховує часову залежність даних - Ефективний для нестационарних рядів
SARIMA	Розширення ARIMA з урахуванням сезонності	- Точність даних з сезонними коливаннями - Автоматичне виявлення періодичності - Надійність для довгострокових прогнозів
Random Forest	Ансамблевий метод машинного навчання на основі дерев	- Ефективна нелінійна модель - Стійкість до перешкод - Можливість врахування додаткових факторів

5

Рисунок Б.5 – Слайд 5 «Вибір методів для відновлення даних»

Схема роботи програми



6

Рисунок Б.6 – Слайд 6 «Схема роботи програми»

Фрагмент коду (методи)

```

if method == "Поліноміальна регресія":
    degree = 3
    poly = PolynomialFeatures(degree)
    X_poly = poly.fit_transform(X.reshape(-1, 1))
    model = LinearRegression()
    model.fit(X_poly, y)
    y_pred = model.predict(poly.transform(all_days.reshape(-1, 1)))
    if len(missing_days) > 0:
        missing_y_pred = model.predict(poly.transform(missing_days.reshape(-1, 1)))

elif method == "Сплайн (PCHIP)":
    interpolator = PchipInterpolator(X, y)
    y_pred = interpolator(all_days)
    if len(missing_days) > 0:
        missing_y_pred = interpolator(missing_days)

elif method == "Згладжуваний сплайн":
    spline = UnivariateSpline(X, y, s=1.5)
    y_pred = spline(all_days)
    if len(missing_days) > 0:
        missing_y_pred = spline(missing_days)

elif method == "ARIMA":
    y_series = pd.Series(y, index=pd.date_range(start=df["date"].min(), periods=len(y)))
    model = ARIMA(y_series, order=(2, 1, 2)).fit()
    all_dates = pd.date_range(start=df["date"].min(), end=df["date"].max())
    forecast = model.predict(start=all_dates[0], end=all_dates[-1])
    y_pred = forecast.values

    if len(missing_days) > 0:
        missing_dates = df["date"].min() + pd.to_timedelta(missing_days, unit='d')
        forecast_missing = model.predict(start=missing_dates.min(), end=missing_dates.max(), typ='levels')
        missing_y_pred = forecast_missing[missing_dates].values

    except Exception as e:
        messagebox.showerror("SARIMA помилка", f"SARIMA не вдалося побудувати модель: {e}")
        return

elif method == "Random Forest":
    model = RandomForestRegressor()
    model.fit(X.reshape(-1, 1), y)
    y_pred = model.predict(all_days.reshape(-1, 1))
    if len(missing_days) > 0:
        missing_y_pred = model.predict(missing_days.reshape(-1, 1))

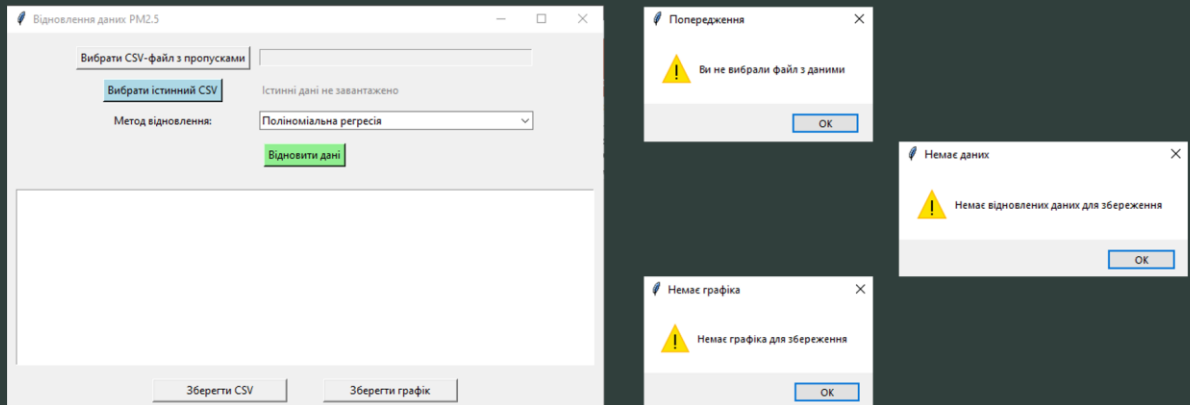
else:
    messagebox.showerror("Помилка", f"Невідомий метод: {method}")
    return

```

7

Рисунок Б.7 – Слайд 7 «Фрагмент коду(методи)»

Інтерфейс користувача



8

Рисунок Б.8 – Слайд 8 «Інтерфейс користувача»

Фрагмент коду (інтерфейс)

```
root = tk.Tk()
root.title("Відновлення даних PM2.5")

file_path_var = tk.StringVar()
method_var = tk.StringVar(value="Поліноміальна регресія")

frame = tk.Frame(root, padx=10, pady=10)
frame.pack()

btn_browse = tk.Button(frame, text="Вибрати CSV-файл з пропущеними", command=browse_file)
btn_browse.grid(row=0, column=0, padx=5, pady=5)

entry_file = tk.Entry(frame, textvariable=file_path_var, width=50, state='readonly')
entry_file.grid(row=0, column=1, padx=5, pady=5)

btn_true = tk.Button(frame, text="Вибрати істинний CSV", command=browse_true_file,
bg="lightblue")
btn_true.grid(row=1, column=0, padx=5, pady=5)

true_file_label = tk.Label(frame, text="Істинні дані не завантажено", fg="gray")
true_file_label.grid(row=1, column=1, padx=5, pady=5, sticky="w")

label_method = tk.Label(frame, text="Метод відновлення:")
label_method.grid(row=2, column=0, padx=5, pady=5)

method_menu = tk.Combobox(frame, textvariable=method_var, state="readonly", width=47)
method_menu['values'] = [
    "Поліноміальна регресія",
    "Сплайн (PCHIP)",
    "Згладжуючий сплайн",
    "ARIMA",
    "SARIMA",
    "Random Forest",
]
method_menu.grid(row=2, column=1, padx=5, pady=5)

btn_run = tk.Button(frame, text="Відновити дані", command=run_processing, bg="lightgreen")
btn_run.grid(row=3, column=0, columnspan=2, pady=10)

text_output = tk.Text(root, height=12, width=80, wrap='none', font=('Courier New', 10))
text_output.pack(padx=10, pady=5)

save_buttons_frame = tk.Frame(root)
save_buttons_frame.pack(pady=10)

btn_save_csv = tk.Button(save_buttons_frame, text="Зберегти CSV",
command=save_reconstructed_csv, width=20)
btn_save_csv.grid(row=0, column=0, padx=20)

btn_save_plot = tk.Button(save_buttons_frame, text="Зберегти графік",
command=save_figure, width=20)
btn_save_plot.grid(row=0, column=1, padx=20)
```

Рисунок Б.9 – Слайд 9 «Фрагмент коду (інтерфейс)»

Вхідні дані

Фрагмент CSV-файлу з повними даними
(м. Запоріжжя, 2025)

```
date,min,max,median,q1,q3,stdev,count
2025-01-21T00:00:00.000Z,2.1,20.3,6.3,3.9,10.6,3.807,590
2025-01-22T00:00:00.000Z,4.1,11.3,7.6,6.8,8.7,1.355,587
2025-01-23T00:00:00.000Z,2.2,13.9,6.4,2.7,6.2,3.11,585
2025-01-24T00:00:00.000Z,4.8,25.11,7.8,16.8,4.897,585
2025-01-25T00:00:00.000Z,6.7,32.2,14.8,12.3,20.5,5.621,531
2025-01-26T00:00:00.000Z,2.5,13.8,6.8,5.8,8.1,875,515
2025-01-27T00:00:00.000Z,3.8,15.6,9.5,5.8,8.1,899,479
2025-01-28T00:00:00.000Z,9.2,18.6,12.1,12.2,12.9,1.243,591
2025-01-29T00:00:00.000Z,10.5,24.4,12.6,12.13,3.1,199,586
2025-01-30T00:00:00.000Z,10.1,40.2,14.8,12.8,21.35,6.372,444
2025-01-31T00:00:00.000Z,12.1,47.4,17.7,15.3,23.4,559,567
2025-02-01T00:00:00.000Z,2.7,57.9,9.9,8.4,11.2,3.422,552
2025-02-02T00:00:00.000Z,2.3,13.8,5.5,4.8,6.7,1.619,580
2025-02-03T00:00:00.000Z,3.1,55.10,19.8,2.13,5.727,587
2025-02-04T00:00:00.000Z,2.6,28.9,5.6,4.4,8.2,3.693,587
2025-02-05T00:00:00.000Z,1.8,36.2,7.2,3.9,8.5,5.545,590
2025-02-06T00:00:00.000Z,4.1,28.1,9.1,7.9,10.5,2.675,577
2025-02-07T00:00:00.000Z,3.4,12.4,6.3,5.3,7.2,1.412,585
2025-02-08T00:00:00.000Z,2.5,12.6,5.55,4.7,5.2,226,582
2025-02-09T00:00:00.000Z,5.4,29.2,9.4,8.4,10.9,3.785,583
2025-02-10T00:00:00.000Z,6.5,24.2,10.8,10.11,6.1,501,492
2025-02-11T00:00:00.000Z,5.7,40.6,10.8,11.8,4.414,542
2025-02-12T00:00:00.000Z,6.4,21.2,13.4,10.15,15.75,3.197,537
2025-02-13T00:00:00.000Z,13.2,69.9,25.20,05.30,5.9,872,488
2025-02-14T00:00:00.000Z,4.2,27.2,7.1,5.9,11.7,5.007,374
2025-02-15T00:00:00.000Z,2.5,17.2,7.6,6.3,11.3,429,478
2025-02-16T00:00:00.000Z,3.1,30.6,5.5,4.59,8.05,4.511,489
2025-02-17T00:00:00.000Z,4.2,49.9,9.2,6.1,18.89,11.978,487
2025-02-18T00:00:00.000Z,4.5,26.2,8.4,6.6,14.5,4.481,369
2025-02-19T00:00:00.000Z,3.4,65.4,19.3,7.8,27.8,13.968,569
2025-02-20T00:00:00.000Z,5.5,53.2,15.1,11.23,9.9,675,582
```

Фрагмент CSV-файлу з пропущеними даними
(м. Запоріжжя, 2025)

```
date,min,max,median,q1,q3,stdev,count
2025-01-21T00:00:00.000Z,2.1,20.3,6.3,3.9,10.6,3.807,590
2025-01-22T00:00:00.000Z,4.1,11.3,7.6,6.8,8.7,1.355,587
2025-01-23T00:00:00.000Z,2.2,13.9,6.4,2.7,6.2,3.11,585
2025-01-24T00:00:00.000Z,4.8,25.11,7.8,16.8,4.897,585
2025-01-25T00:00:00.000Z,6.7,32.2,14.8,12.3,20.5,5.621,531
2025-01-26T00:00:00.000Z,2.5,13.8,6.8,5.8,8.1,875,515
2025-01-27T00:00:00.000Z,3.8,15.6,9.5,5.8,8.1,899,479
2025-01-28T00:00:00.000Z,9.2,18.6,12.1,12.2,12.9,1.243,591
2025-01-29T00:00:00.000Z,10.5,24.4,12.6,12.13,3.1,199,586
2025-01-30T00:00:00.000Z,10.1,40.2,14.8,12.8,21.35,6.372,444
2025-01-31T00:00:00.000Z,12.1,47.4,17.7,15.3,23.4,559,567
2025-02-01T00:00:00.000Z,2.5,12.6,5.55,4.7,5.2,226,582
2025-02-09T00:00:00.000Z,5.4,29.2,9.4,8.4,10.9,3.785,583
2025-02-10T00:00:00.000Z,6.5,24.2,10.8,10.11,6.1,501,492
2025-02-11T00:00:00.000Z,5.7,40.6,10.8,11.8,4.414,542
2025-02-12T00:00:00.000Z,6.4,21.2,13.4,10.15,15.75,3.197,537
2025-02-13T00:00:00.000Z,13.2,69.9,25.20,05.30,5.9,872,488
2025-02-14T00:00:00.000Z,4.2,27.2,7.1,5.9,11.7,5.007,374
2025-02-15T00:00:00.000Z,2.5,17.2,7.6,6.3,11.3,429,478
2025-02-16T00:00:00.000Z,3.1,30.6,5.5,4.59,8.05,4.511,489
2025-02-17T00:00:00.000Z,4.2,49.9,9.2,6.1,18.89,11.978,487
2025-02-18T00:00:00.000Z,4.5,26.2,8.4,6.6,14.5,4.481,369
2025-02-19T00:00:00.000Z,3.4,65.4,19.3,7.8,27.8,13.968,569
2025-02-20T00:00:00.000Z,5.5,53.2,15.1,11.23,9.9,675,582
```

Рисунок Б.10 – Слайд 10 «Вхідні дані»

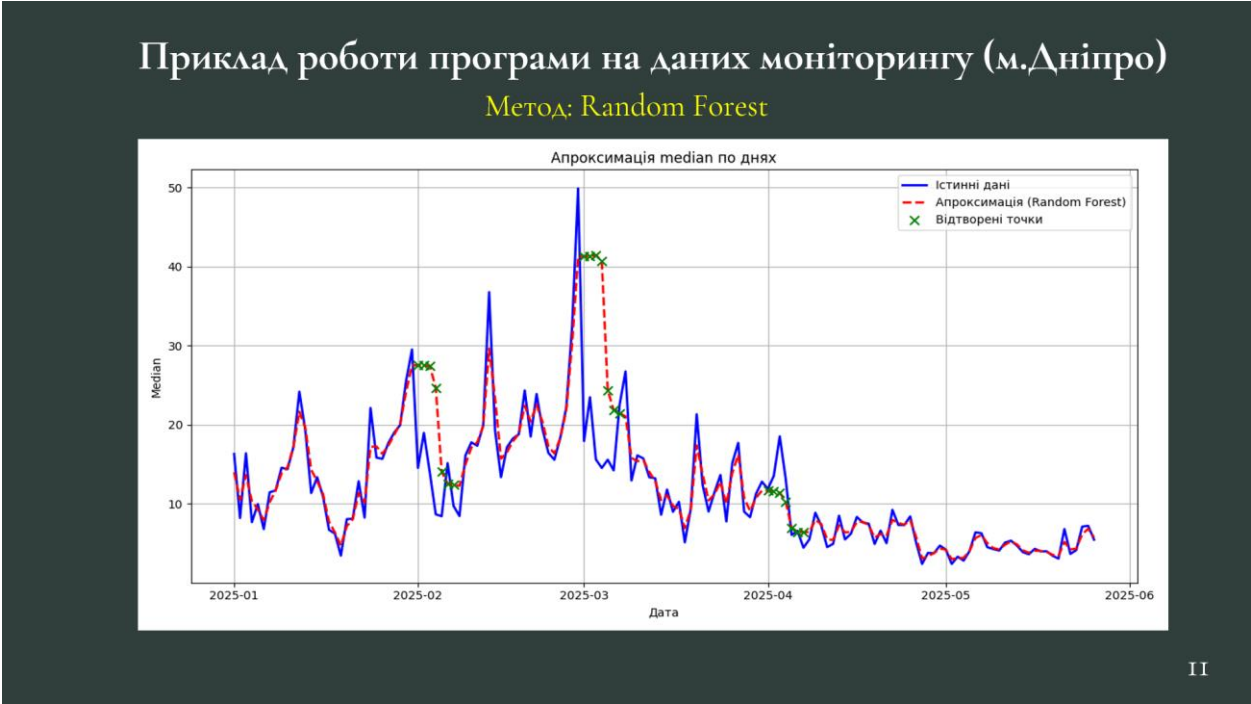


Рисунок Б.11 – Слайд 11 «Приклад роботи програми на даних моніторингу (м.Дніпро)»

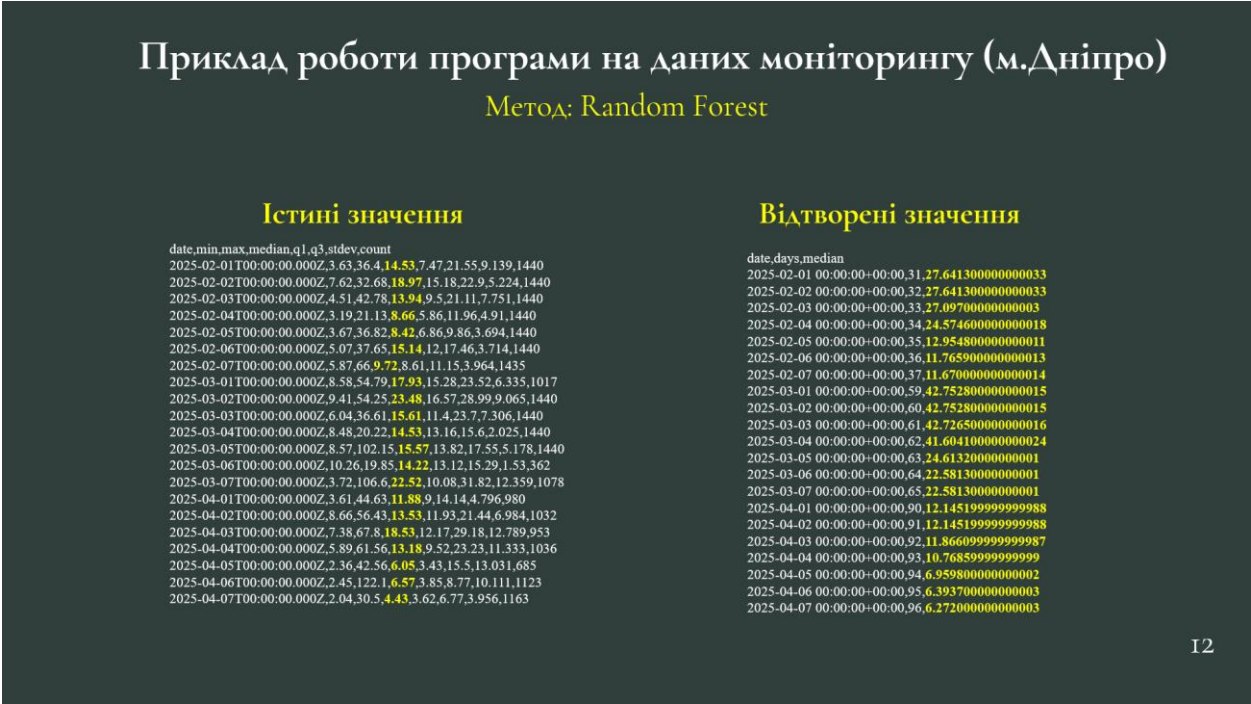


Рисунок Б.12 – Слайд 12 «Приклад роботи програми на даних моніторингу (м.Дніпро)»

Оцінка точності відновлення (RMSE)

RMSE — це квадратний корінь із середньоквадратичної помилки. Вона вимірює середню величину помилки моделі в тих же одиницях, що й дані.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$$

де $\hat{y}_i$ — значення, відновлене обраним методом для i -го дня з пропущеними даними;

y_i — відповідне істинне значення;

n — загальна кількість пропущених точок.

Результат оцінки

м. Дніпро (Поліноміальна регресія)

2025-03-05	63	17.056
2025-03-06	64	16.856
2025-03-07	65	16.650
2025-04-01	90	10.490
2025-04-02	91	10.233
2025-04-03	92	9.979
2025-04-04	93	9.726
2025-04-05	94	9.475
2025-04-06	95	9.227
2025-04-07	96	8.981

Середньоквадратична помилка (RMSE): 5.4449

Результат оцінки

м. Дніпро (Random Forest)

2025-03-05	63	26.136
2025-03-06	64	25.172
2025-03-07	65	22.438
2025-04-01	90	11.850
2025-04-02	91	11.850
2025-04-03	92	11.657
2025-04-04	93	10.643
2025-04-05	94	7.092
2025-04-06	95	6.649
2025-04-07	96	6.606

Середньоквадратична помилка (RMSE): 11.3662

13

Рисунок Б.13 – Слайд 13 «Оцінка точності відновлення (RMSE)»

Результати дослідження

Метод	Запоріжжя	Дніпро	Львів
Поліноміальна регресія	3.3435	5.4449	4.1081
Сплайн (PCHIP)	10.7069	15.8703	3.7568
Згладжуючий сплайн	24.3721	28.4775	6.6059
ARIMA	3.7341	7.1642	4.2926
SARIMA	4.1790	8.0255	4.6167
Random Forest	7.8399	12.2426	4.8987

Поліноміальна регресія продемонструвала найвищу універсальність серед усіх досліджуваних методів у межах розглянутої вибірки даних.

14

Рисунок Б.14 – Слайд 14 «Результати дослідження»

Висновок :

У дослідженні проаналізовано ефективність методів відновлення пропущених даних PM_{2.5} на основі реальних вимірів концентрації PM_{2.5} м. Запоріжжя, Дніпра та Львова. Розглянуто методи: поліноміальна регресія, інтерполяція сплайнами, ARIMA, SARIMA, Random Forest.

- Найкращий результат (за RMSE) — поліноміальна регресія 3-го ступеня.
- Практичний результат — створено програму для Windows із графічним інтерфейсом, побудовану на Python.
- Можливості: вибір методу, обробка користувацьких даних, візуалізація, експорт результатів.

Якісне заповнення пропущених даних забезпечує точніші прогнози, кращу інтерпретацію результатів і створює основу для застосування методів машинного навчання. Дослідження має як наукову, так і практичну цінність для екологічного моніторингу та аналітики.

15

Рисунок Б.15 – Слайд 15 «Висновок»