AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ GEOLOGII, GEOFIZYKI I OCHRONY ŚRODOWISKA

KATEDRA GEOINFORMATYKI I INFORMATYKI STOSOWANEJ

Praca inżynierska

Optymalizacja wielokryterialna z wykorzystaniem metod inspirowanych naturą

Multi-criteria optimization using nature-inspired methods

Autor: Adam Lewiński
Kierunek studiów: Inżynieria i Analiza Danych
Opiekun pracy: dr hab. inż. Tomasz Danek

Kraków, 2024

Spis treści

1 Wstęp

1.1 Wprowadzenie

W ostatnich latach powstało dużo algorytmów inspirowanych naturą. Z reguły są to algorytmy, które próbują naśladować proces ewolucji bądź zachowanie poszczególnych grup istot (Alanis et al., 2018). Zgodnie z jedną z przyjętych taksonomii, takie metody można podzielić na algorytmy inspirowane biologią, modele hybrydowe (np. wspierane przez sztuczne sieci neuronowe) oraz ogólne modele obliczeniowe sztucznej inteligencji. Wśród algorytmów inspirowanych biologią wyszczególnić można grupę, która odpowiedzialna jest za optymalizację (Jakšić et al., 2023). Znajduje ona zastosowanie w różnych dziedzinach wśród których można wymienić na przykład przetwórstwo żywności (Sarkar et al., 2022), odszumianie zdjęć medycznych (Vineeth et al., 2021), kontrolę ruchu drogowego z uwzględnieniem aspektów ekologicznych (Jovanović et al., 2022), farmację (Luukkonen et al., 2023).

1.2 Cel pracy i zakres pracy

Celem niniejszej pracy naukowej jest zbadanie możliwości zastosowania algorytmów optymalizacji wielokryterialnej inspirowanych naturą w celu konstrukcji systemu przewidującego stan zanieczyszczenia powietrza pyłami zawieszonymi. W wyniku badań został stworzony program komputerowy, którego zadaniem jest tworzenie prognoz poziomu zanieczyszczeń dla powiatu Kraków. Postanowiono skupić się na tym, aby przy niewielkim nakładzie pracy użytkownik napisanej struktury był w stanie zastosować ją dla wybranego przez siebie obszaru, aby uzyskać satysfakcjonujące wyniki.

1.3 Pyły zawieszone

Jedną z przyjętych definicji zanieczyszczenia powietrza jest zdefiniowanie go przez Światową Organizację Zdrowia (WHO) jako zanieczyszczenie środowiska poprzez czynnik chemiczny, fizyczny bądź biologiczny, który modyfikuje właściwości atmosfery (WHO, 2024). Wśród tych zanieczyszczeń można wyróżnić grupę pyłów zawieszonych. Jednym z możliwych podziałów jest rozróżnienie aerozoli atmosferycznych ze względu na wielkość ich średnicy – PM_{10}, $PM_{2.5}$, $PM_{1.0}$. Cząstki tych zanieczyszczeń mają odpowiednio średnicę mniejszą niż 10 μm, 2.5 μm

i 1 µm. Można wskazać zmienne geograficzne, które wpływają na poziom zanieczyszczenia powietrza (Youngseob et al., 2015). Wymienione aerozole nie pozostają bez wpływu na zdrowie człowieka (Li et al., 2020, Thangavel et al., 2022). Istotnym jest, aby monitorować poziom zanieczyszczenia powietrza oraz podejmować świadome decyzje o ekspozycji na opisane czynniki; pomocne mogą być normy publikowane przez WHO (Rysunek 1.1.). Obecnie wynoszą one dla PM_{10} 45 µg/m^3 średnio w ciągu jednego dnia oraz 15 µg/m^3 dla rocznego okresu uśrednienia. W przypadku pyłów $PM_{2.5}$ jest to średnio 15 µg/m^3 w przypadku jednej doby i średnio 5 µg/m^3 w przypadku roku (EEA, 2024).

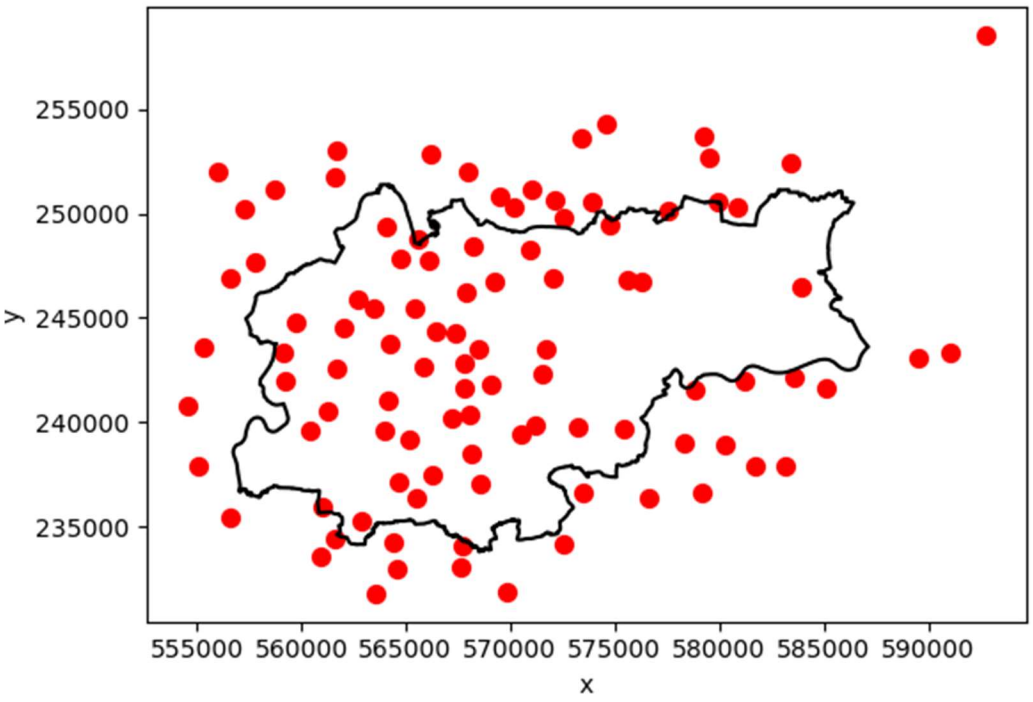
Pollutant	Averaging Time	2005 AQGs	2021 AQGs
$PM_{2.5}$, µg/m^3	Annual	10	5
	24-hour[a]	25	15
PM_{10}, µg/m^3	Annual	20	15
	24-hour[a]	50	45
O_3, µg/m^3	Peak season[b]	-	60
	8-hour[a]	100	100
NO_2, µg/m^3	Annual	40	10
	24-hour[a]	-	25
SO_2, µg/m^3	24-hour[a]	20	40
CO, mg/m^3	24-hour[a]	-	4

Rysunek 1.1. Normy zanieczyszczeń powietrza różnymi cząstkami zaproponowane przez Światową Organizację Zdrowia. Źródło: https://www.who.int/news-room/feature-stories/detail/what-are-the-who-air-quality-guidelines, dostęp z dnia 17.03.2024.

2 Materiały i metody

2.1 Dane

Dane dotyczące zanieczyszczenia powietrza pochodzą z darmowego interfejsu programowania aplikacji (ang. API) udostępnionego przez platformę Airly[1]. Odpytywane niskokosztowe czujniki jakości powietrza znajdują się na terenie powiatu Kraków oraz okolic (Rysunek 2.1). Żądania (ang. requests) były wysyłane codziennie w okresie 15.03.2023 – 31.01.2024 w celu uzyskania danych o godzinnej rozdzielczości z ostatnich 24 godzin. Wykorzystano informacje dotyczące poziomu zanieczyszczenia powietrza pyłami zawieszonymi PM_{10}, $PM_{2.5}$ i $PM_{1.0}$. Sensory tego typu są dosyć gęsto rozmieszczone, ponadto są one mocno skorelowane z odczytami czujników, za które odpowiada Główny Inspektorat Ochrony Środowiska (Zaręba i Danek, 2022).



Rysunek 2.1. Lokalizacja odpytywanych sensorów na terenie powiatu Kraków i okolicy. Układ współrzędnych o kodzie EPSG:2180.

Dane meteorologiczne pochodzą z serwisu Open-Meteo[2]. Informacje zbierano w tym samym okresie co odczyty z czujników. Pobierano również informacje na temat

[1] https://developer.airly.org/
[2] https://open-meteo.com/

prognozy na następne 3 dni. Wśród zebranych danych znalazły się wszystkie zmienne meteorologiczne o godzinnej rozdzielczości, które były oferowane przez serwis w dniu 15.03.2023. Obejmują one między innymi informacje na temat temperatury powietrza oraz gleby, wilgotności powietrza oraz gleby, opadów atmosferycznych, zachmurzenia, ciśnienia atmosferycznego, czy kierunku i prędkości wiatru. Odczyty dla lokalizacji każdego sensora pochodzą z najbliższego punktu siatki pomiarowej platformy Open-Meteo. Zdecydowano się nie korzystać z danych atmosferycznych oferowanych przez udostępnione sensory, ze względu na brakujące wartości oraz pojawiające się wartości odstające, które są spowodowane awariami sprzętu pomiarowego.

2.2 AutoML

W dzisiejszych czasach można zaobserwować wzrost liczby prac naukowych poruszających to zagadnienie (Pugliese et al., 2021). Automatyczne uczenie maszynowe umożliwia stworzenie własnych modeli oraz potoków (ang. pipelines) uczenia maszynowego bez dogłębnej znajomości szeroko rozumianej danologii (ang. data science). Eksperci dziedzinowi mogą dzięki temu korzystać z najnowszych osiągnieć naukowych i technologicznych bez konieczności dogłębnej edukacji w zakresie uczenia maszynowego (He et al., 2021). Istnieją gotowe rozwiązania takie jak Auto-WEKA 2.0 czy Auto-sklearn, które mają na celu zautomatyzować selekcję cech, optymalizację hiperparametrów, bądź dobór parametrów potoku uczenia maszynowego (Waring et al., 2020). Ponadto takie systemy mogą wykorzystywać różne strategie doboru optymalnych parametrów, algorytmów, metod; z powodzeniem można stosować przeszukiwanie w siatce (ang. grid search), przeszukiwanie losowe, metodę spadku gradientu, algorytmy ewolucyjne i inne metody (Zöller i Huber, 2021). W niniejszej pracy zostanie poruszony temat doboru odpowiednich parametrów potoku uczenia maszynowego za pomocą algorytmów optymalizacji wielokryterialnej.

2.3 Optymalizacja wielokryterialna

Celem optymalizacji wielokryterialnej, zwanej też optymalizacją Pareto, jest znalezienie zestawu akceptowalnych rozwiązań danego problemu opisanego za pomocą funkcji f_i dla $i = 1, 2, \ldots, k$ (Ngatchou et al., 2005). Zbiór optymalnych rozwiązań w sensie Pareto składa się z rozwiązań, które nie są zdominowane przez

inne rozwiązania, mówiąc inaczej dominują wszystkie pozostałe. Niech dany będzie następujący problem (1):

$$\min_{x \in X}\big(f_1(x), f_2(x), \dots, f_k(x)\big) \text{ dla } k \geq 2, \quad \text{gdzie } X \subseteq \mathbb{R}^n. \tag{1}$$

Rozwiązanie x można opisać zgodnie z przedstawioną zależnością (2) (Miettinen, 1999).

$$f: X \to \mathbb{R}^k \qquad x \mapsto \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{pmatrix} \tag{2}$$

Rozwiązanie x_1 dominuje rozwiązanie x_2 jeżeli spełnione są dwa następujące warunki:

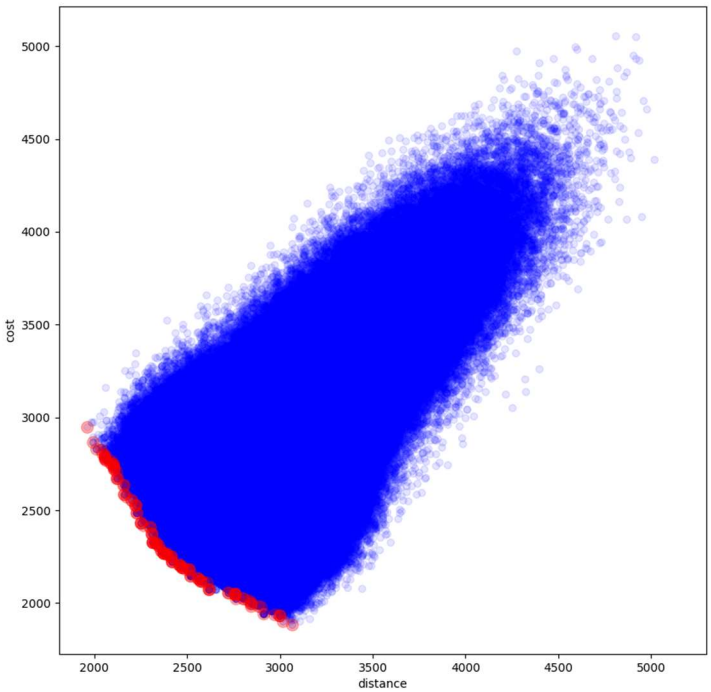
1. Rozwiązanie x_1 jest nie gorsze niż rozwiązanie x_2 ze względu na wszystkie kryteria (3).

$$\forall i \in \{1, 2, \dots, k\}, f_i(x_1) \leq f_i(x_2) \tag{3}$$

2. Rozwiązanie x_1 jest lepsze niż rozwiązanie x_2 ze względu na co najmniej jedno kryterium (4) (Deb, 2005).

$$\exists i \in \{1, 2, \dots, k\}, f_i(x_1) < f_i(x_2) \tag{4}$$

Zbiór rozwiązań dominujących tworzy front Pareto (Rysunek 2.2.).

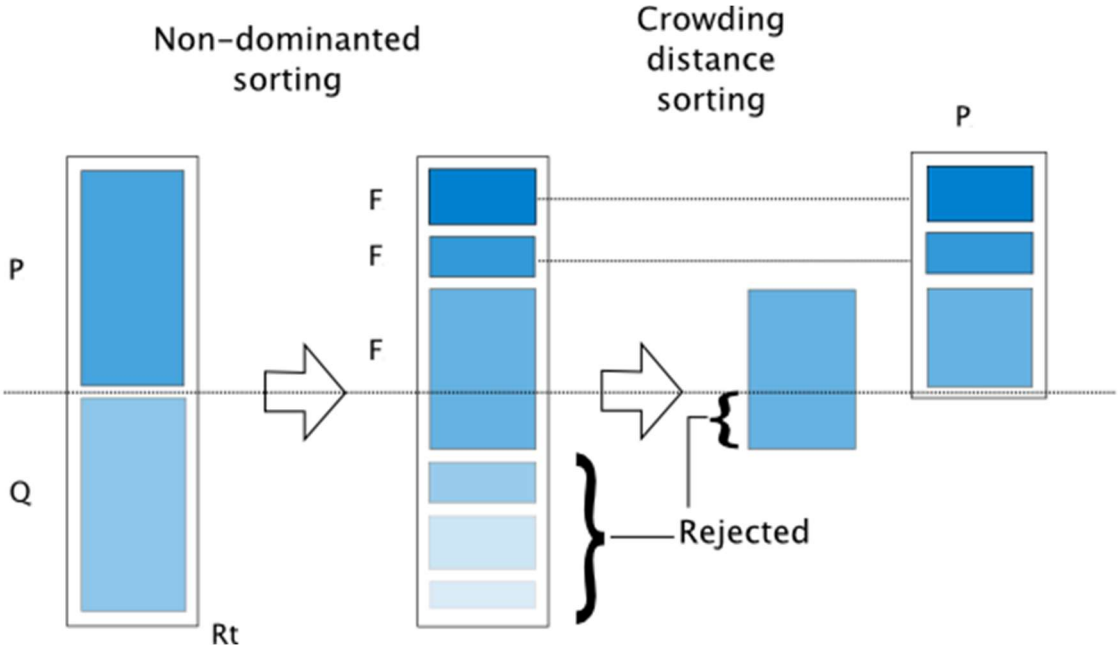


Rysunek 2.2. Przykładowy front Pareto. Koszt i dystans to odpowiednio $f_1(x)$ i $f_2(x)$, niebieskie punkty to zbiór eksplorowanych rozwiązań, czerwone punkty to rozwiązania dominujące wszystkie pozostałe. Rysunek przedstawia poszukiwanie rozwiązań dla problemu komiwojażera przy użyciu macierzy dystansu oraz macierzy kosztu.

2.4 NSGA-II

NSGA-II to algorytm genetyczny służący do optymalizacji wielokryterialnej. Algorytm NSGA-II został opisany w 2002 (Deb et al., 2002) jako ulepszenie metody NSGA zaproponowanej w 1994 roku (Srinivas i Deb, 1994). NSGA-II cechuje się mniejszą złożonością obliczeniową, brakiem parametru udostępniania (ang. sharing parameter) oraz posiada mechanizm elitaryzmu (Deb et al., 2002).

Działanie algorytmu zostało zaprezentowane na rysunku (Rysunek 2.3.) oraz opisane w kolejnych krokach (Miernik et al., 2021):



Rysunek 2.3. Działanie algorytmu NSGA-II, gdzie P to populacja wstępna; Q to populacja powstała na skutek operatorów selekcji i mutacji; Rt to zbiór osobników P i Q (Rt = P ∪ Q) w generacji t; F to kolejne fronty Pareto powstałe na skutek sortowania i przydzielania rangi (pierwszy front zawiera rozwiązania dominujące wszystkie pozostałe, drugi front dominuje wszystkie rozwiązania oprócz tych z pierwszego frontu itd.). Zbiór P kolejnej generacji uzyskiwany jest poprzez wybór kolejnych frontów F o coraz to niższej randze, a w przypadku potrzeby podziału frontu wybierane są wartości na podstawie metryki Manhattan (Blank i Deb, 2020). Źródło: https://pymoo.org/algorithms/moo/nsga2.html, dostęp z dnia 17.03.2024.

1) Inicjalizacja populacji P składającej się z losowo wygenerowanych rozwiązań
2) Wygenerowanie populacji Q na skutek zastosowania operatorów krzyżowania oraz mutacji. Populacja Q jest tak samo liczna jak populacja P.
3) Sortowanie osobników populacji Rt ($\mathrm{Rt} = \mathrm{P} \cup \mathrm{Q}$) według przypisanej rangi tworząc kolejne fronty Pareto F. Pierwszy front Pareto zawiera rozwiązania dominujące wszystkie pozostałe - posiadające najwyższą rangę. Kolejne zbiory będą zawierać coraz to niższe rangi, będą zdominowane przez

zestawy o wyższej randze oraz będą dominować fronty Pareto o niższej randze.

4) Stworzenie nowej populacji P dla kolejnego pokolenia t, które będzie tak samo liczne jak wstępna populacja P. Nowy zbiór osobników P powstaje na skutek sumowania kolejnych frontów Pareto F poczynając od tego o najwyższej randze, który zawiera rozwiązania dominujące wszystkie pozostałe. W przypadku, gdy wynikowa populacja po dodaniu kolejnego zbioru rozwiązań F miałaby być większa niż początkowa populacja, część rozwiązań jest odrzucana. Selekcja odbywa się na podstawie metryki Manhattan w celu wyboru jak najbardziej zróżnicowanych rozwiązań.
5) Powrót do kroku 2, jeżeli nie osiągnięto kryterium stopu.
6) Wynikiem końcowym jest front Pareto o najwyższej randze z ostatniej populacji P.

2.5 NSGA-III

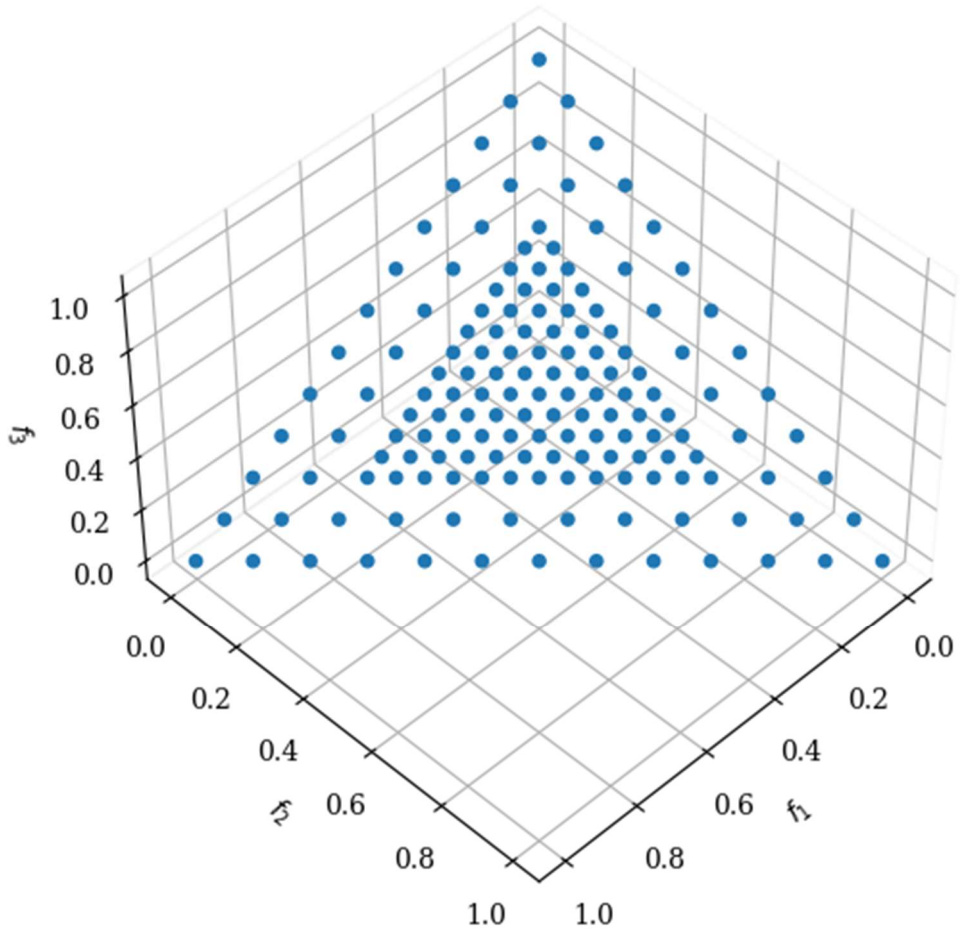
Wraz ze wzrostem liczby optymalizowanych funkcji rośnie liczba rozwiązań niezdominowanych, w związku z czym wymagana jest liczniejsza populacja, aby móc wziąć pod uwagę wszystkie osobniki elitarne. W celu poprawy wyboru rozwiązań do dalszej reprodukcji został zaproponowany algorytm NSGA-III (Deb i Jain, 2014).

Algorytm NSGA-III jest rozszerzeniem opisanego wcześniej algorytmu NSGA-II. Algorytm ten różni się od swojego poprzednika metodą wyboru osobników do nowej populacji, jeżeli dodanie kolejnego frontu Pareto do zbioru osobników w następnym pokoleniu miałoby spowodować stworzenie liczniejszej grupy niż ta pierwotna. W przypadku, gdy suma kolejnych składowych zbiorów F byłaby równoliczna z wstępną populacją P, nie następują żadne modyfikacje (Deb i Jain, 2014). W przeciwnej sytuacji podejmowane są następujące kroki, aby wybrać osobniki do nowej populacji P z ostatniego kwalifikującego się do selekcji frontu Pareto (Campos Ciro et al., 2016):

1) Określenie punktów odniesienia w hiperprzestrzeni (Rysunek 2.4.). Pozwala to zachować różnorodność otrzymanych rozwiązań poprzez późniejsze powiązanie ich z punktami odniesienia.
2) Normalizacja otrzymanych rozwiązań w populacji. Operacja ta odbywa się poprzez obliczenie punktu idealnego określonego poprzez minimum każdej optymalizowanej funkcji. Następnie front Pareto zostaje przesunięty poprzez odjęcie współrzędnych punktu idealnego. Kolejne

zaproponowane przekształcenia pozwalają otrzymać poszukiwaną hiperprzestrzeń
(Deb i Jain, 2014).

3) Powiązanie punktów odniesienia z znormalizowanym frontem Pareto. Każdemu osobnikowi populacji jest przypisywany najbliższy punkt odniesienia.
4) Jeden punkt odniesienia może być powiązany z wieloma rozwiązaniami. Istotnym jest, aby zachować rozwiązania, które są najbliższe do danych punktów odniesienia.



Rysunek 2.4. Punkty odniesienia, wygenerowane metodą Das-Dennis, dla trzech optymalizowanych funkcji. Wykorzystano dwie warstwy, różnie przeskalowane (0.5, 1), na skutek czego punkty odniesienia są gęściej rozłożone w środku i rzadziej na obrzeżu wygenerowanej hiperprzestrzeni. Zastosowano wartość parametru p równą 12 dla każdej warstwy. Źródło: https://pymoo.org/misc/reference_directions.html, dostęp z dnia 17.03.2024.

3 Implementacja

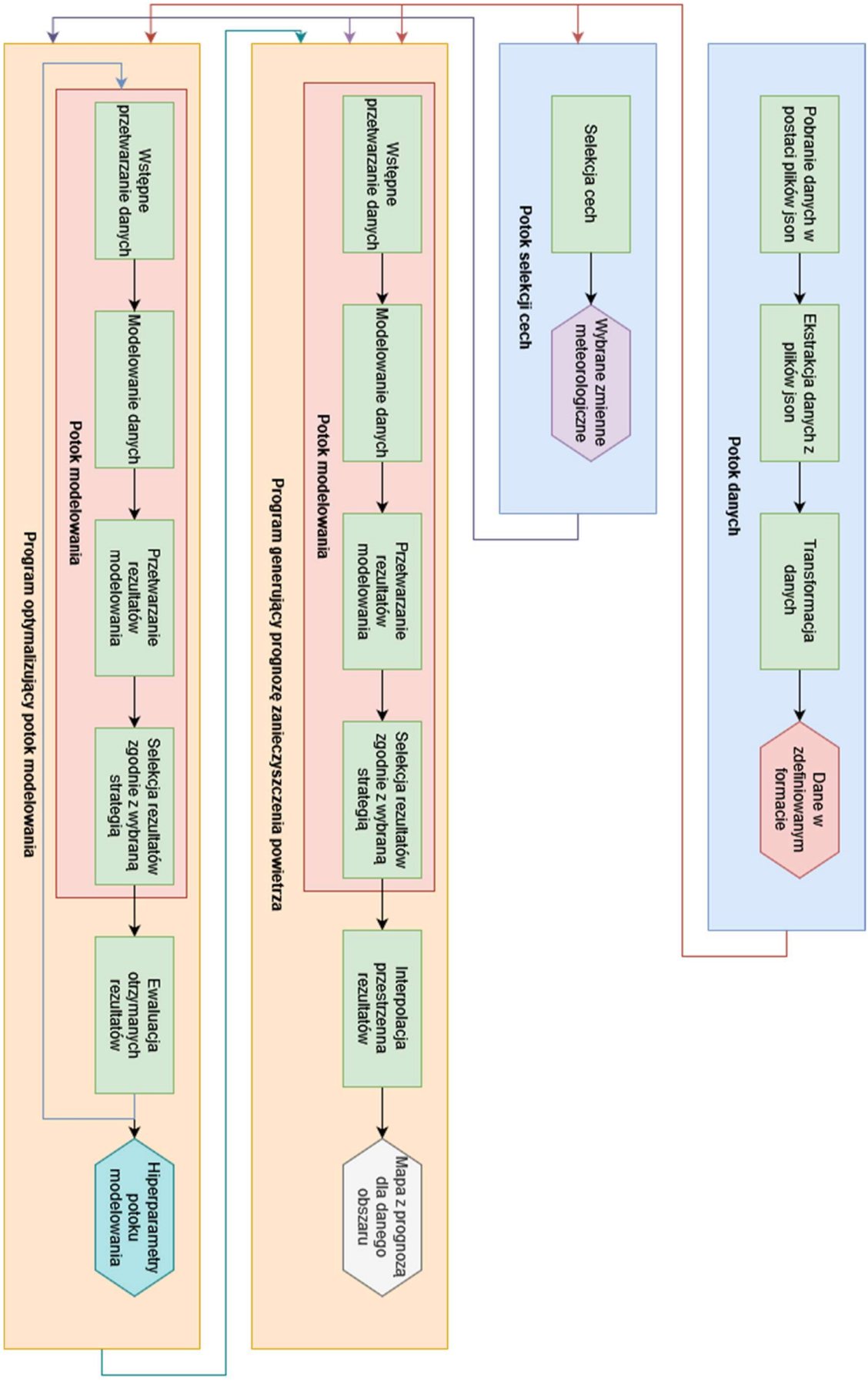
3.1 Struktura systemu

Struktura programu została zaprezentowana na rysunku (Rysunek 3.1.). System składa się z potoków odpowiedzialnych za przygotowanie danych oraz selekcję zmiennych meteorologicznych, które następnie wykorzystywane są przez program generujący prognozę zanieczyszczenia powietrza oraz program optymalizujący potok modelowania. Każdy z modułów może być uruchomiony przez użytkownika w zależności od potrzeb. Ponadto system pozwala na łatwą modyfikację i tworzenie nowych potoków. W konstrukcji systemu starano się zastosować dobre praktyki obecne systemach MLOps (Machine Learning Operations) (Google, 2024).

Potok danych (ang. ETL pipeline) odpowiedzialny jest za dostarczenie danych w zdefiniowanym formacie. Zastosowanie wewnętrznego formatu reprezentacji danych, który jest połączeniem formatu długiego (ang. long data format) oraz szerokiego formatu (ang. wide data format) pozwala na jego łatwą modyfikację poprzez dodanie nowych zmiennych do predykcji (np. tlenki siarki), czy zmiennych pogodowych, zachowując pełną kompatybilność z resztą systemu. Potok wykonuje operacje na wielu plikach, każdy sensor posiada niezależnie od innych kilka plików, które reprezentują różne etapy przetworzenia danych. System w ten sposób zużywa mniej pamięci RAM oraz jest mniej podatny na różnego rodzaju awarie i błędy w kodzie.

Potok selekcji cech odpowiedzialny jest za wybór zmiennych meteorologicznych użytych do modelowania. Potok danych dostarcza wiele różnych zmiennych egzogenicznych, które następnie są selekcjonowane w automatyczny sposób. W prowadzonych badaniach skorzystano z jednej metody selekcji oraz zdecydowano się na izolację tego modułu względem innych komponentów systemu. Automatyczny sposób wyboru cech został opisany w kolejnym podrozdziale.

Program generujący prognozę zanieczyszczenia powietrza składa się z potoku odpowiedzialnego za modelowanie danych oraz zadań odpowiedzialnych za stworzenie mapy zanieczyszczeń powietrza dla zadanego obszaru. Poszczególne komponenty zostaną opisane w kolejnych podrozdziałach.



Rysunek 3.1. Schemat stworzonej struktury

Program optymalizujący potok modelowania danych ma za zadanie dobór odpowiednich hiperparametrów dla potoku modelowania danych. Proces ten odbywa się z wykorzystaniem optymalizacji wielokryterialnej za pomocą algorytmów inspirowanych naturą. W realizacji wykorzystanej podczas badań, zdecydowano się na ograniczenie liczby danych, które zostały wzięte pod uwagę w procesie doboru parametrów dla całego procesu. W celu redukcji ilości informacji posłużono się klasteryzacją szeregów czasowych opartą na metryce euklidesowej, aby wybrać reprezentatywną grupę sensorów (Tavenard et al., 2020). Podział na grupy odbył się według wartości zmiennych zależnych (pyły zawieszone) oraz ilości brakujących informacji. Wybrano takie zmienne, aby uwzględnić jakość działania systemu dla różnych szeregów czasowych oraz w warunkach, kiedy liczba danych jest ograniczona np. ze względu na awarię sensora.

3.2 Selekcja zmiennych meteorologicznych

Selekcja zmiennych meteorologicznych została przeprowadzona za pomocą współczynnika współliniowości (VIF). Ze zbioru zmiennych objaśniających podlegających temu procesowi wykluczono temperaturę powietrza na wysokości 2 metrów, względną wilgotność powietrza na wysokości 2 metrów oraz ciśnienie na średniej wysokości morza. Następnie postępowano zgodnie z algorytmem przedstawionym na diagramie (Rysunek 3.2.). Do obliczenia VIF zastosowano równanie (5); jako próg przyjęto wartość 5 zgodnie z „praktyczną zasadą” (Marcoulides i Raykov, 2018).

$$VIF_i = \frac{1}{1 - R_i} \tag{5}$$

Rysunek 3.2. Schemat blokowy selekcji cech za pomocą VIF.

3.3 Potok modelowania danych

Potok modelowania danych składa się z wstępnego przetwarzania danych, modelowania danych, obróbki otrzymanych rezultatów oraz selekcji właściwych rezultatów jako prognozę wartości zmiennych zależnych dla każdego sensora. Każdy etap jest konfigurowalny poprzez różnego rodzaju hiperparametry całego potoku. Potok zaczytuje kolejno dane z czujników, a następnie przeprowadza szereg operacji, aby wykonać prognozę dla podanej daty i horyzontu prognozy (ang. forecast horizon). Ponadto dane na wyjściu znajdują się w zdefiniowanym długim formacie, aby były kompatybilne z różnymi komponentami, które mogą wykorzystywać wygenerowane rezultaty.

Wstępne przetwarzanie danych zaczyna się od utworzenia kolejnych szeregów czasowych zgodnie z przyjętą strategią sprawdzianu krzyżowego dla szeregów czasowych (Hyndman i Athanasopoulos, 2021). Do tego celu używane są różne okna: rozszerzające się okno (ang. expanding window splitter) oraz ruchome okna (ang. sliding window splitters) z różnymi parametrami. Potok modelowania może operować na wielu oknach równocześnie. Dobór odpowiednich okien użytych w systemie może zostać dobrany w procesie optymalizacji potoku. Następnie dane są sprawdzane pod kątem aktywności sensora. Jeżeli nie są obecne odczyty z czujnika w zdefiniowanym okresie poprzedzającym datę modelowania, odnotowywany jest odpowiedni komunikat o błędzie dla danego sensora. Ponadto weryfikowana jest ilość wartości brakujących, czy nie przekracza pewnego podanego progu procentowego. Przy użyciu tych filtrów należy uwzględnić braki wynikające z przerw na konserwację

techniczną systemu oraz inne czynniki powodujące jego nieaktywność. Kolejnym krokiem jest uzupełnienie wartości brakujących poprzez wstawianie wartości sprzed podanej liczby godzin. Stosowany jest Filtr Hampela (Hampel, 1974), aby usunąć wartości odstające, które są obecne w danych po przeprowadzonej imputacji. Ostatnim etapem jest ponowna imputacja danych, tym razem za pomocą interpolacji liniowej.

Przygotowane dane następnie są użyte z różnymi modelami uczenia maszynowego w celu wygenerowania prognoz zanieczyszczenia powietrza w lokalizacjach danych sensorów. Dla danych z każdego czujnika używany jest każdy model, który został podany jako hiperparametr potoku modelowania. Modele mogą zostać podane przez eksperta dziedzinowego jak i wybrane podczas procesu doboru optymalnych hiperparametrów potoku. Modele użyte w procesie optymalizacji to: Prophet (Taylor i Letham, 2017), SARIMAX (Seabold i Perktold, 2010), Naive Forecaster (Hyndman i Athanasopoulos, 2021), Croston (Croston, 1972), Linear Regression (Pedregosa et al., 2011), Random Forest Regressor (Breiman, 2001) i Decision Tree Regressor (Breiman et al., 1984). W celu możliwości zastosowania wielu instancji modeli z różnymi parametrami, zostały one pogrupowane w zestawy zgodnie z wiedzą dziedzinową. W ten sposób wartość optymalizowanej funkcji zostaje zaokrąglona do liczby całkowitej, a następnie zamieniona na postać binarną. Wartość bitu decyduje o użyciu danego zestawu modeli w potoku.

Wyniki predykcji stworzone przez model podlegają pewnym przekształceniom. Skonstruowany system przygotowany jest, aby być w stanie odwrócić transformacje przenoszące modelowane dane z jednej przestrzeni w inną. Ponadto aplikowana jest podana funkcja (6) dla każdej przewidzianej wartości w celu eliminacji wartości nieprawidłowych (ujemnych).

$$f(x) = \max(0, x) \tag{6}$$

Ostatnim krokiem potoku modelowania danych jest selekcja właściwych rezultatów. Jako wejście do tego etapu podane są prognozy wykonane dla każdego danego okna oraz dla każdego danego modelu. Zgodnie z przyjętą strategią, rezultatem może być średnia bądź mediana wszystkich modeli, w tym przypadku właściwości sprawdzianu krzyżowego nie są używane, bądź nawet niegenerowane we wcześniejszych krokach. Inną dopuszczalną strategią jest użycie sprawdzianu krzyżowego do oceny wyników dla każdego sensora z osobna zgodnie z przyjętą metryką, a następnie wybranie modelu, który sprawdzał się najlepiej dla posiadanych odczytów z danego czujnika. W tym przypadku musi zostać wygenerowana większa liczba prognoz.

3.4 Ewaluacja

Potok uczenia maszynowego oceniany jest ze względu na metryki podane przez użytkownika. Osoba korzystająca z systemu może zdecydować jakie funkcje zostaną poddane procesowi minimalizacji. Pozwala to stworzyć całą pulę optymalnych rozwiązań, spośród których można wybrać te odpowiadające bieżącym potrzebom. Można również utrzymywać kilka programów w celu produkcji wielu prognoz i prezentacji ich w postaci panelu (ang. dashboard). Ponadto domyślna konfiguracja dodaje metryki obliczające czas działania potoku modelowania oraz ujemny procent udanych predykcji. Ważnym jest, aby dobrać odpowiedni czas działania programu względem jakości uzyskiwanych rezultatów, aby maksymalizować czas istotności prognozy oraz minimalizować koszt utrzymania systemu. Optymalizowanie ujemnego procenta udanych predykcji pozwala dobrać takie parametry, które pozwolą na uzyskanie rezultatów w skrajnych warunkach działania systemu np. przy niedoborze danych lub ich nadmiarze. Dzięki takiemu podejściu, możliwe jest wybranie takich hiperparametrów, które mogą ułatwić wprowadzenie nowych czujników do systemu poprzez odpowiedni dobór obsługiwanych okien (balans pomiędzy czasem działania, osiągami oraz wymaganą liczbą obserwacji).

3.5 Interpolacja przestrzenna

Ostateczne prognozy wygenerowane przez system poddawane są interpolacji w celu uzyskania mapy prezentującej zanieczyszczenie powietrza danym aerozolem w zdefiniowanym czasie. Interaktywne mapy pozwalają na łatwą prezentację danych z różnych godzin. Stworzona struktura umożliwia przeprowadzenie tego procesu za pomocą różnych metod z wykorzystaniem pakietu języka Python – SciPy (Virtanen et al., 2020). W przeprowadzonych badaniach skorzystano z radialnej funkcji bazowej (7) charakterystycznej dla TPS (ang. thin plate splines) (Bookstein, 1989).

$$\varphi(r) = r^2 \log r \tag{7}$$

4 Wyniki

4.1 Ocena uzyskanych systemów

W ramach badań wygenerowano systemy za pomocą opisanych algorytmów optymalizacji wielokryterialnej. Horyzont prognozy, który użyto przy optymalizacji, to 24 godziny. Rozwiązania generowano dla czterech różnych dat, które charakteryzowały się różnymi przebiegami zanieczyszczenia powietrza w ciągu doby (różne okresy roku). Obydwa algorytmy uruchomiono z takimi samymi parametrami liczby generacji oraz rozmiaru populacji, pozostałe parametry zostały ustawione jako domyślne (Blank i Deb, 2020). Jako wartość dla każdej optymalizowanej funkcji wykorzystywano wartość maksymalną z czterech różnych przebiegów. Uzyskane wyniki porównano z wartościami historycznymi, ocena utworzonych systemów odbyła się przy użyciu wszystkich sensorów, data predykcji jest odległa o ponad jeden miesiąc od ostatniej daty użytej do optymalizacji. Ze względu na to, że proces optymalizacji odbywał się przy użyciu danych z trzech sensorów, a ostateczna ocena wykorzystywała wszystkie, niektóre rozwiązania (Tabela 4.1., Tabela 4.2.) mogą być zdominowane przez inne. Ponadto wszystkie rozwiązania posiadały taką samą wartość udanych predykcji, dlatego ta funkcja została pominięta w zestawieniu. Ostatnie tabela (Tabela 4.3.) zawiera ocenę predykcji oferowanych przez serwis Airly, dane pochodzą z tego samego źródła co dane historyczne. Warto zwrócić uwagę, że dane z Airly nie zawierają prognozy dla pyłu zawieszonego $PM_{1.0}$.

Tabela 4.1. Wartość optymalizowanych funkcji na zbiorze testowym dla rozwiązań otrzymanych w wyniku działania algorytmu NSGA-III. Użyto pierwiastka średniego błędu kwadratowego dla trzech zmiennych zależnych oraz czasu działania systemu.

Nazwa rozwiązania	RMSE $PM_{1.0}$	RMSE $PM_{2.5}$	RMSE PM_{10}	Czas działania [s]
NSGA-III X1	10.60	17.32	20.62	399
NSGA-III X2	11.03	17.91	21.14	348
NSGA-III X3	11.79	19.13	22.38	254
NSGA-III X4	13.58	22.11	26.11	326
NSGA-III X5	11.30	18.43	22.08	17680

Tabela 4.2. Wartość optymalizowanych funkcji na zbiorze testowym dla rozwiązań otrzymanych w wyniku działania algorytmu NSGA-II. Użyto pierwiastka średniego błędu kwadratowego dla trzech zmiennych zależnych oraz czasu działania systemu.

Nazwa rozwiązania	RMSE $PM_{1.0}$	RMSE $PM_{2.5}$	RMSE PM_{10}	Czas działania [s]
NSGA-II X1	10.85	18.02	21.50	24175
NSGA-II X2	10.44	17.40	20.65	25898
NSGA-II X3	12.53	20.55	24.24	19717
NSGA-II X4	13.07	21.45	25.36	18203
NSGA-II X5	12.22	19.91	23.39	31595
NSGA-II X6	12.84	21.06	24.88	19468
NSGA-II X7	11.20	18.52	21.87	22957
NSGA-II X8	10.41	17.30	20.44	23392
NSGA-II X9	11.90	19.66	23.35	27766
NSGA-II X10	10.52	17.38	20.58	26468
NSGA-II X11	11.67	19.25	22.75	30985

Tabela 4.1. Wartość metryk dla prognozy oferowanej przez serwis Airly. Nieobecne są dane dotyczące prognozy pyłu $PM_{1.0}$. Zdecydowano się nie podawać czasu działania.

Nazwa rozwiązania	RMSE $PM_{1.0}$	RMSE $PM_{2.5}$	RMSE PM_{10}	Czas działania [s]
Prognoza Airly	-	14.89	15.73	-

4.2 Specyfika uzyskanych rozwiązań

W przypadku użycia NSGA-II liczba różnych okien do selekcji danych treningowych była większa. Wszystkie potoki modelowania używały rozszerzającego się okna. Najpopularniejszy parametr w zbiorze rozwiązań optymalnych wykluczał ruchome okna o wielkości horyzontu prognozy oraz jego trzykrotności. W przypadku, kiedy te okna były używane, wykluczone były ruchome okna większe niż trzydziestokrotność horyzontu prognozy. Wielkość okien może mieć kluczowe znaczenie m.in. ze względu na to, że okno Filtru Hampela było sztywno ustawione na wartość 72 obserwacji (Löning et al., 2019). W przypadku algorytmu NSGA-III tylko jedno rozwiązanie używało rozszerzającego się okna. Obecne były również

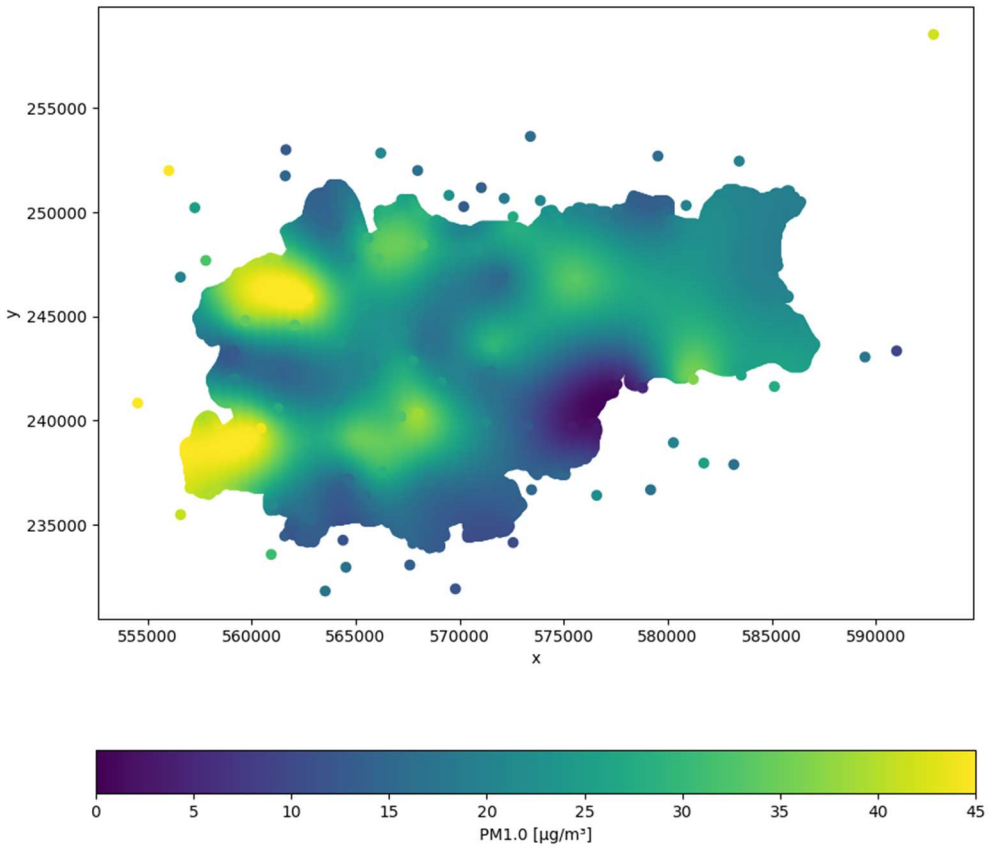
ruchome okna o długości 24, i 72 obserwacji. Redukcja liczby okien i ich rozmiaru miała wpływ na skrócenie czasu obliczeń.

Potoki modelowania korzystały ze wszystkich danych modeli. W wyniku optymalizacji tylko jedno rozwiązanie (NSGA-II X8) używało mniejszej liczby modeli, (2 pakiety, 6 modeli) mimo możliwości doboru tylko pojedynczych modeli bądź pakietów złożonych z samych prostych modeli – naiwnych. Ponadto preferowane było użycie Prophet, SARIMAX (zwykle bez komponentu sezonowego) i Random Forrest Regressor. Również modele naiwne były dokładane w większości przypadków. Dobór odpowiednich modeli sprawił, że prognozy dla procesu NSGA-III zajęły mniej czasu.

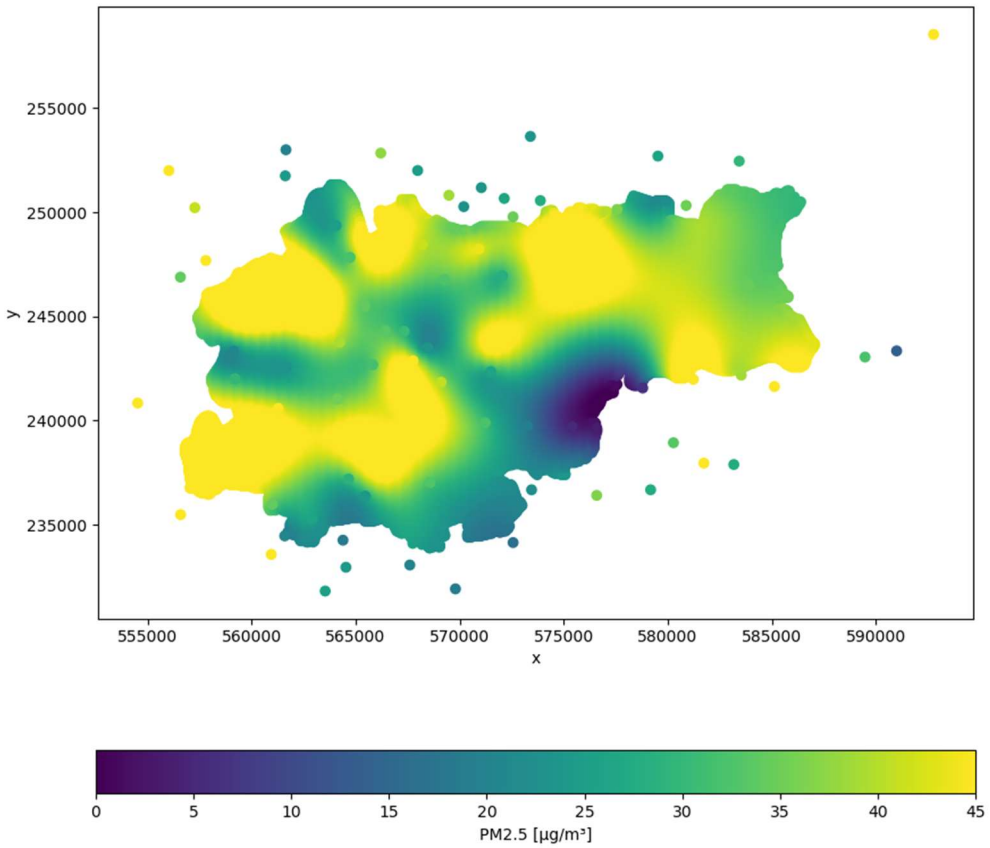
Strategie użyte do wyboru predykcji to średnia i mediana. W puli rozwiązań optymalnych nie pojawił się żaden przypadek sprawdzianu krzyżowego o rozmiarze trzykrotności horyzontu prognozy (wymaga to czterokrotnego uruchomienia potoku modelowania w celu wygenerowania jednej prognozy). W przypadku algorytmu NSGA-II zawsze stosowana była strategia mediany. Algorytm NSGA-III generował rozwiązania w większości opierające się na użyciu średniej.

4.3 Analiza wyników

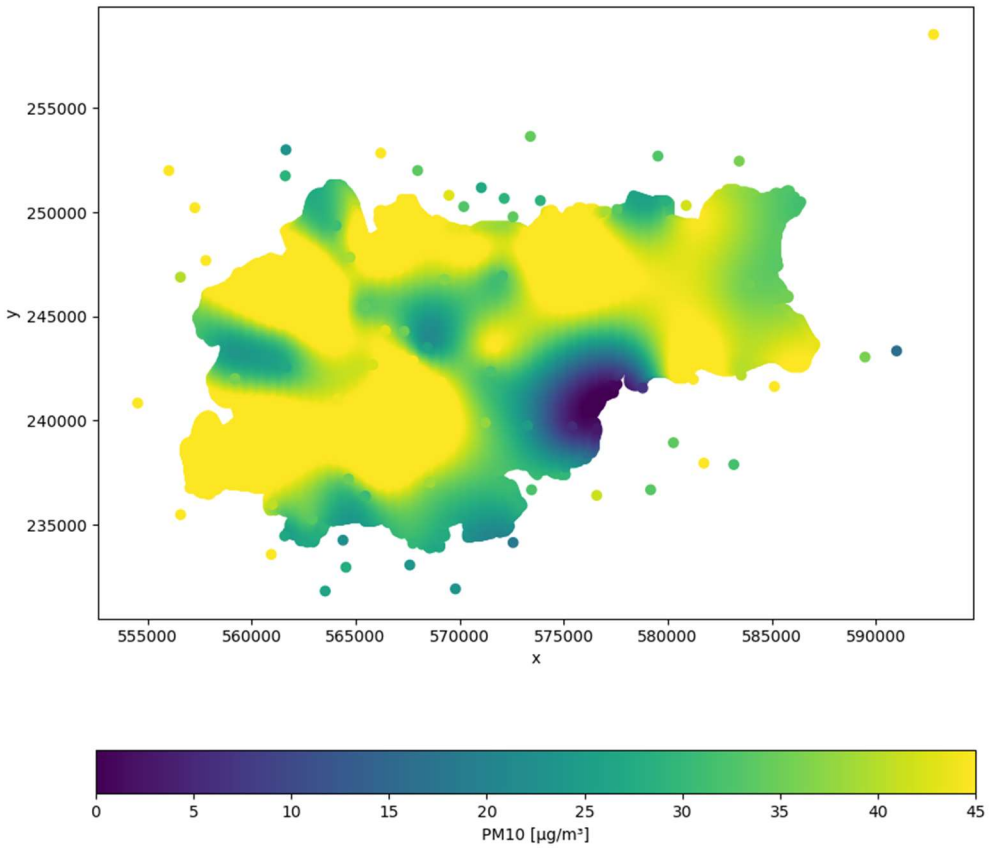
Jako wynik końcowy działania systemu wykonano mapy obrazujące zanieczyszczenie powietrza różnymi pyłami zawieszonymi (Rysunek 4.1., Rysunek 4.2., Rysunek 4.3.). Wybrano jedną godzinę, aby zaprezentować działanie interpolacji i przykładowy rezultat. Obszar interpolacji obejmuje teren powiatu Kraków. Wyraźnie można zaobserwować grupowanie się pyłów w poszczególnych rejonach na mapie.



Rysunek 4.1. Mapa powstała na skutek działania systemu (NSGA-III X8). Prezentowana mapa przedstawia teren powiatu Kraków (EPSG:2180), punkty opisujące prognozowane wartości zanieczyszczenia powietrza aerozolem $PM_{1.0}$ w lokalizacjach sensorów oraz wartości interpolowane pomiędzy nimi. Mapę wykonano dla daty 30.01.2024, godziny 17 UTC.



Rysunek 4.2. Mapa powstała na skutek działania systemu (NSGA-III X8). Prezentowana mapa przedstawia teren powiatu Kraków (EPSG:2180), punkty opisujące prognozowane wartości zanieczyszczenia powietrza aerozolem $PM_{2.5}$ w lokalizacjach sensorów oraz wartości interpolowane pomiędzy nimi. Mapę wykonano dla daty 30.01.2024, godziny 17 UTC.

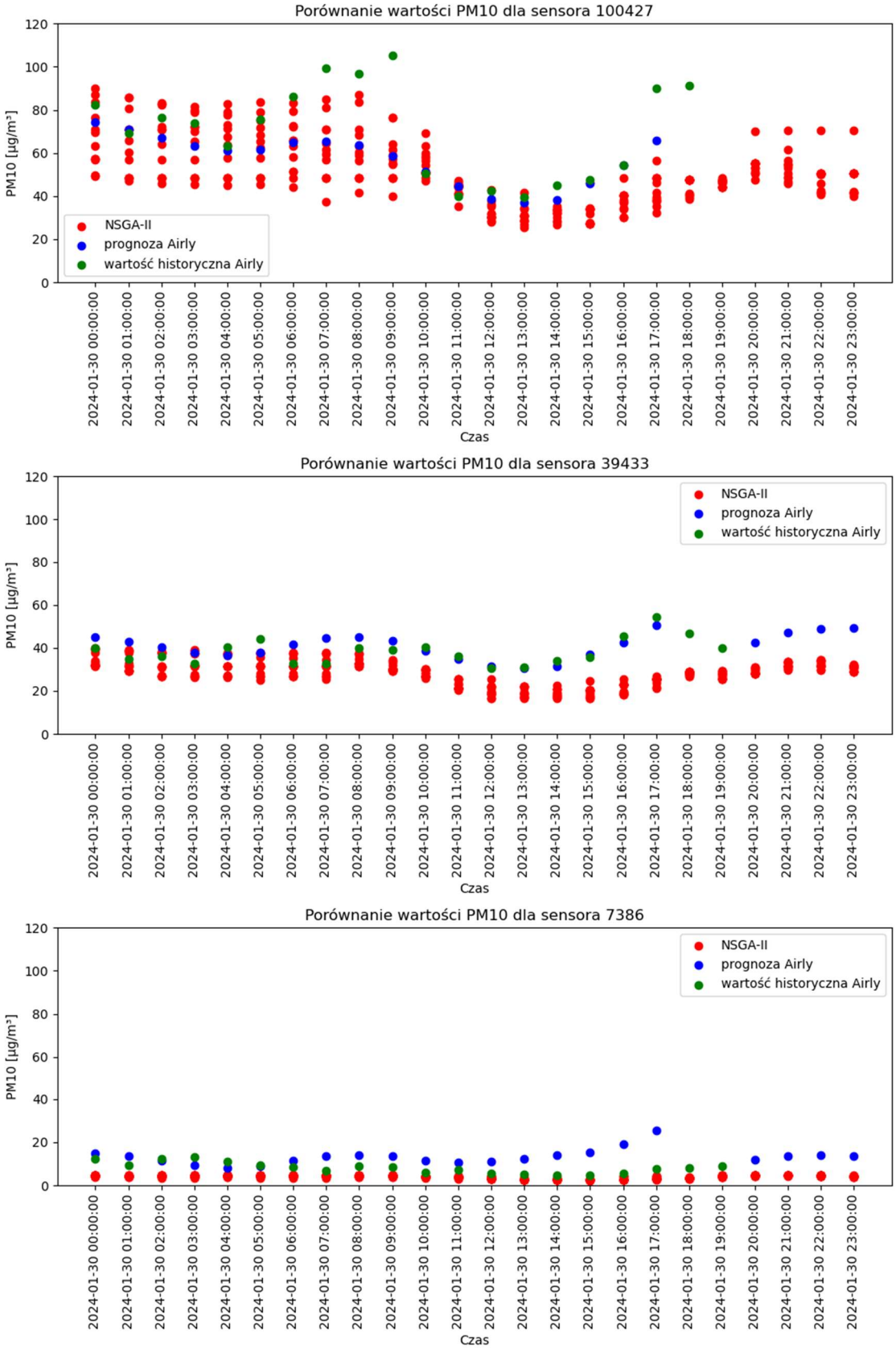


Rysunek 4.3. Mapa powstała na skutek działania systemu (NSGA-III X8). Prezentowana mapa przedstawia teren powiatu Kraków (EPSG:2180), punkty opisujące prognozowane wartości zanieczyszczenia powietrza aerozolem PM_{10} w lokalizacjach sensorów oraz wartości interpolowane pomiędzy nimi. Mapę wykonano dla daty 30.01.2024, godziny 17 UTC.

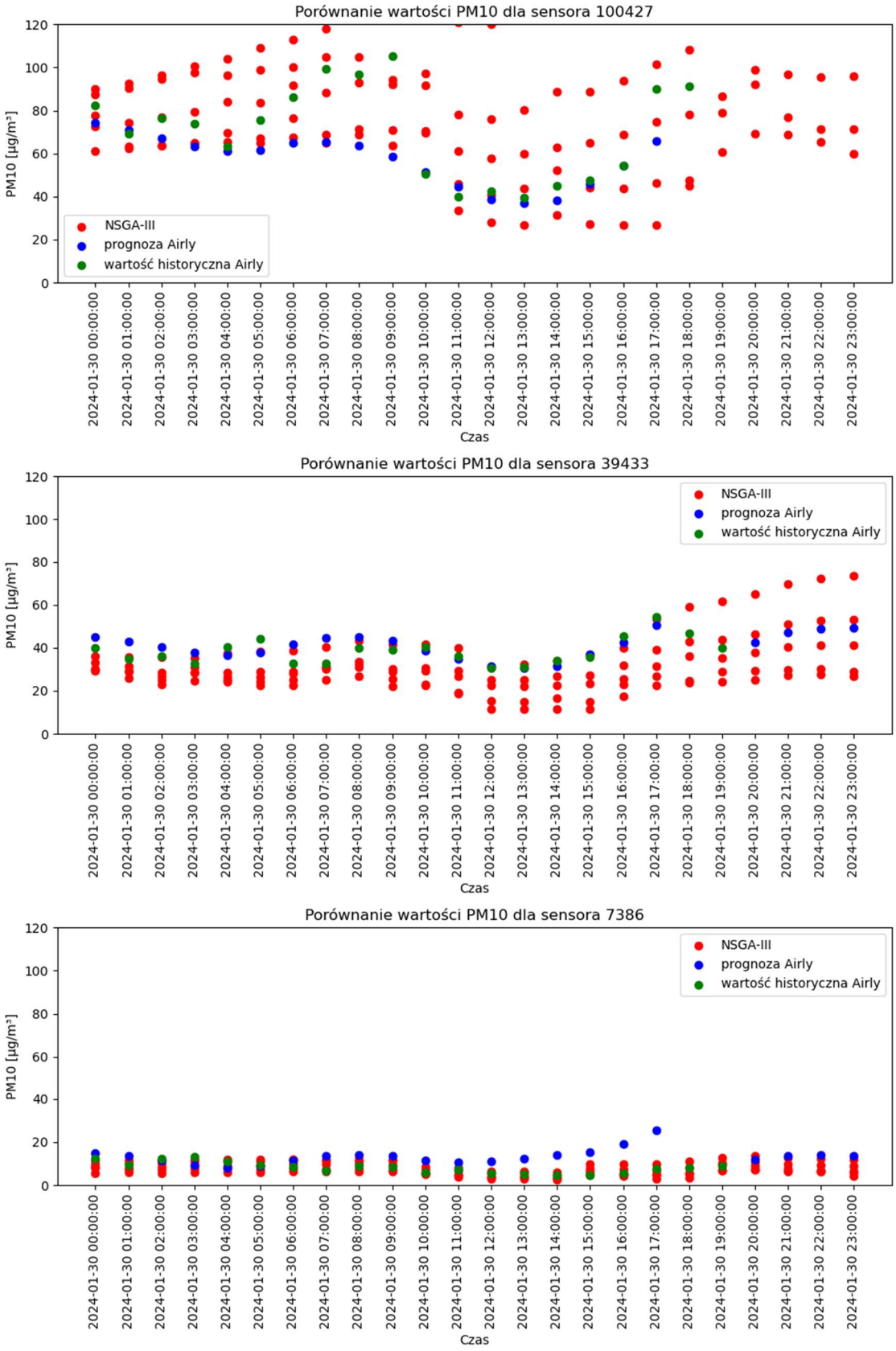
W celu analizy wyników wykonano również wykresy zestawiające prognozy przez utworzone systemy, prognozę Airly oraz wartości historyczne. Zwizualizowano trzy różne sensory, które zostały wybrane za pomocą klasteryzacji (Tavenard et al., 2020), aby zaprezentować różne charakterystyki zadanego problemu.

W przypadku, kiedy został wykorzystany algorytm NSGA-II (Rysunek 4.4.), utworzone systemy mają dość zbliżone charakterystyki. Prognozowane wartości są bardziej zwarte niż w przypadku systemów generowanych przez algorytm NSGA-III (Rysunek 4.5.). Ponadto można zaobserwować, że pojawia się problem w wykryciu nagłego skoku poziomu zanieczyszczenia powietrza pyłem zawieszonym PM_{10}. Może to wynikać ze stosowanych detekcji wartości odstających oraz imputacji.

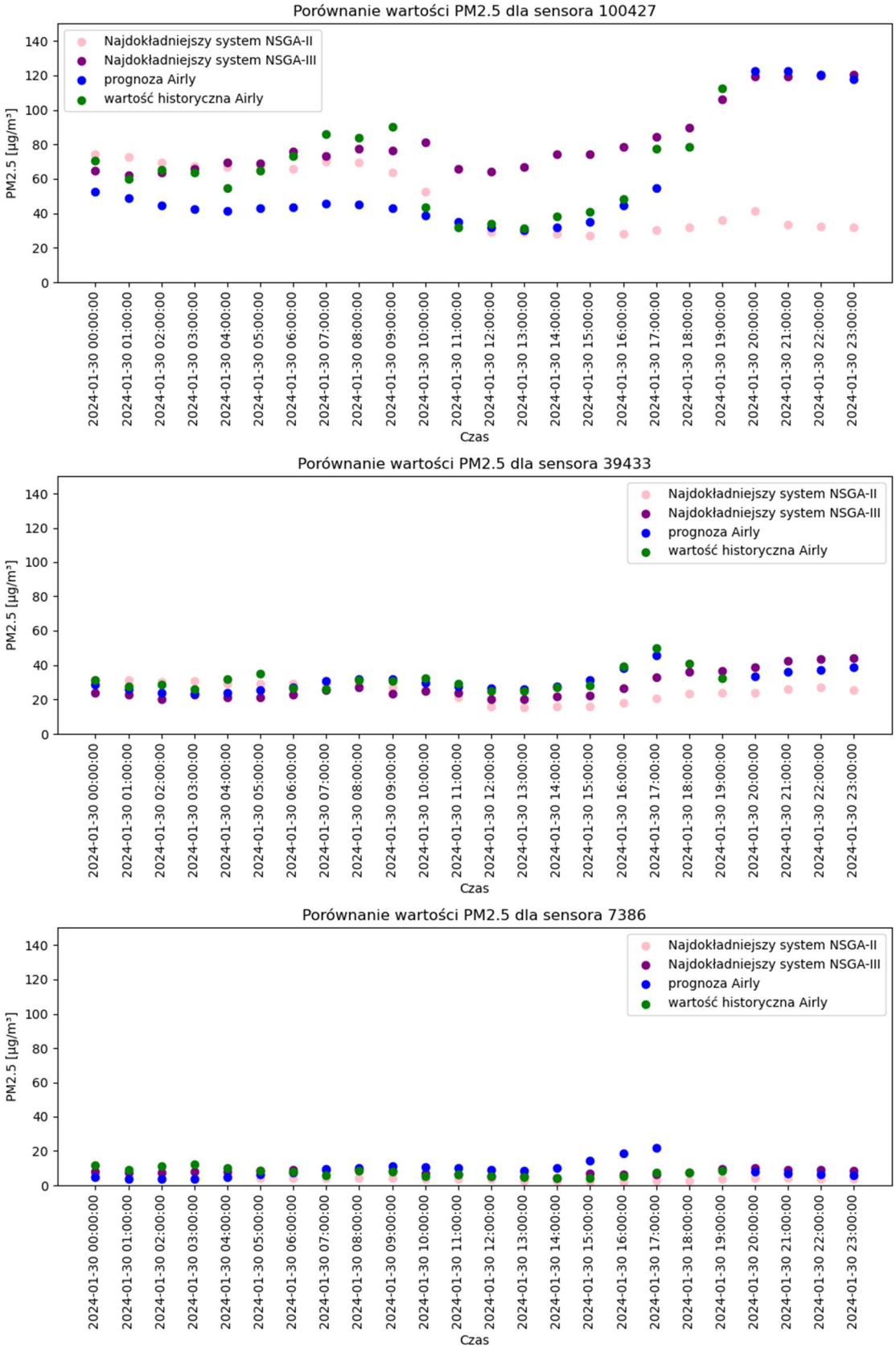
Ponadto zostało wykonane zestawienie najdokładniejszych utworzonych systemów do przewidywania zanieczyszczenia powietrza pyłem zawieszonym $PM_{2.5}$ z danymi pochodzącymi z Airly (Rysunek 4.6.). Wizualizacja została wykonana w celu oceny charakterystyki prognozy Airly i prognoz wykreowanych przez utworzone systemy. System stworzony przez proces algorytmu NSGA-II (NSGA-II X8) lepiej poradził sobie z predykcją spadku zanieczyszczenia odczytywanego przez sensor 100427. System powstały na skutek działania algorytmu NSGA-III (NSGA-III X1) był mniej wrażliwy na ten spadek, natomiast dzięki temu lepiej wskazywał wartości w godzinach wieczornych. Ponadto system (NSGA-III X1) był w stanie sprawniej reagować na nagłe zmiany stężenia aerozolu w atmosferze. Podczas oceny otrzymanych rozwiązań należy również przyjrzeć się specyfice technicznej sensorów nisko kosztowych i ich porównaniu względem sensorów rządowych (Danek i Zaręba, 2021)



Rysunek 4.4. Prezentacja wartość historycznych i prognoz dostarczonych przez Airly wraz z rozwiązaniami generowanymi przez systemy utworzone podczas optymalizacji algorytmem NSGA-II. Pojawiają się braki w wartościach historycznych oraz prognozach Airly.



Rysunek 4.5. Prezentacja wartość historycznych i prognoz dostarczonych przez Airly wraz z rozwiązaniami generowanymi przez systemy utworzone podczas optymalizacji algorytmem NSGA-III. Pojawiają się braki w wartościach historycznych oraz prognozach Airly.



Rysunek 4.6. Prezentacja wartość historycznych i prognoz dostarczonych przez Airly wraz z rozwiązaniami generowanymi przez najdokładniejsze utworzone systemy. Pojawiają się braki w wartościach historycznych oraz prognozach Airly.

5 Podsumowanie

W ramach badań została stworzona struktura, która narzuca styl pracy z danymi dostarczonymi przez sensory oraz danymi meteorologicznymi. Przeprowadzono z jej udziałem serię automatycznych eksperymentów, które dostarczyły satysfakcjonujące wyniki. Ponadto udało się przejść z jej pomocą cały proces od początku do końca, poprzez pobranie danych, modelowanie oraz stworzenie mapy wynikowej, która zawiera interpolację rezultatów na terenie całego powiatu Kraków, w usystematyzowany sposób. Stworzona struktura otwiera drogę do licznych eksperymentów, poprzez możliwość zarządzania konkretnymi modułami systemu w sposób niezależny od siebie. Za jej pomocą można prowadzić badania skupiając się na poszczególnych aspektach takich jak: selekcja cech, wstępne przetwarzanie danych, modelowanie danych w celu uzyskania prognoz dla odczytów poszczególnych sensorów, czy modelowanie przestrzenne danych poprzez dobór odpowiedniej metody interpolacji. Znajduję się w niej również miejsce na indywidualne traktowanie sensorów, które nie zostało jednak w pełni wykorzystane ze względu na brak rozwiązań, które cechowałyby się użyciem sprawdzianu krzyżowego w celu doboru najdokładniejszego stworzonego modelu względem danej metryki. W celu utylizacji tej własności systemu, można na przykład rozszerzyć horyzont prognozy.

W pracy zaprezentowano możliwość zastosowania optymalizacji wielokryterialnej w celu automatyzacji eksperymentów związanych z uczeniem maszynowym. Przedstawiono wyniki doświadczeń, które wskazują, że sama metoda może być odpowiednia do zarządzania procesami tego rodzaju, zastępując dogłębną analizę danych i pracę naukowców danych (ang. data scientists). Opisano rezultaty działań dwóch algorytmów inspirowanych naturą: NSGA-II i NSGA-III. Zwrócono uwagę na specyfikę rozwiązań przez nie generowanych oraz wykonano porównanie ich charakterystyk. Finalne rezultaty zestawiono z prognozą oferowaną przez serwis Airly.

Kolejne eksperymenty mogą pokazać, czy system wygenerowany w ten sposób jest w stanie rywalizować z rozwiązaniami obecnymi na rynku. Można również rozbudować strukturę poprzez dodanie klasyfikacji danych bądź ich klasteryzacji, w celu stworzenia pełniejszego rozwiązania automatycznego uczenia maszynowego. Interesującym również kierunkiem może być zastosowanie sieci neuronowej do agregacji prognoz dla poszczególnych sensorów. Możliwość wyboru pomiędzy różnymi metodami selekcji cech, detekcji wartości odstających czy interpolacji może także przynieść poprawę wyników poprzez uwzględnienie specyfiki konkretnej geolokalizacji oraz charakterystyki sensora.

6 Bibliografia

Alanis, A., Arana-Daniel, N. i López-Franco, C. (2018). *Bio-inspired Algorithms for Engineering.*

Blank, J. i Deb, K. (2020). *Pymoo: Multi-Objective Optimization in Python.*

Bookstein, F. L. (1989). *Principal Warps: Thin-Plate Splines and the Decomposition of Deformations.*

Breiman, L. (2001). *Random Forests.*

Breiman, L., Friedman, J., Olshen, R. A. i Stone, C. J. (1984). *Classification and Regression Trees.*

Campos Ciro, G., Dugardin, F., Yalaoui, F. i Kelly, R. (2016). *A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraintsp0.*

Croston, J. D. (1972). *Forecasting and Stock Control for Intermittent Demands.*

Danek, T. i Zaręba, M. (2021). *The Use of Public Data from Low-Cost Sensors for the Geospatial Analysis of Air Pollution from Solid Fuel Heating during the COVID-19 Pandemic Spring Period in Krakow, Poland.*

Deb, K. (2005). Multi-Objective Optimization. W E. K. Burke i G. Kendall, *Search Methodologies* (strony 273-316).

Deb, K. i Jain, H. (2014). *An Evolutionary Many-Objective Optimization.*

Deb, K., Pratap, A., Agarwal, S. i Meyarivan, T. (2002). *A fast and elitist multiobjective genetic algorithm: NSGA-II.*

EEA. (2024, Marzec 11). *EEA - publications*. Pobrano z lokalizacji Witryna sieci Web Europejskiej Agencji Środowiska: https://www.eea.europa.eu/publications/status-of-air-quality-in-Europe-2022/europes-air-quality-status-2022/world-health-organization-who-air

Google, C. A. (2024, 03 18). Pobrano z lokalizacji Cloud Google: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Hampel, F. R. (1974). *The influence curve and its role in robust estimation.*

He, X., Zhao, K. i Chu, X. (2021). *AutoML: A Survey of the State-of-the-Art.*

Hyndman, R. J. i Athanasopoulos, G. (2021). *Forecasting: Principles and Practice (3rd ed).*

Jakšić, Z., Devi, S., Jakšić, O. i Guha, K. (2023). *A Comprehensive Review of Bio-Inspired Optimization Algorithms Including Applications in Microelectronics and Nanophotonics.*

Jovanović, A., Stevanović, A., Dobrota, N. i Teodorović, D. (2022). *Ecology based network traffic control: A bee colony optimization approach.*

Li, T., Yu, Y., Sun, Z. i Duan, J. (2022). *A comprehensive understanding of ambient particulate matter and its components on the adverse health effects based from epidemiological and laboratory evidence.*

Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J. i Király, F. (2019). *sktime: A Unified Interface for Machine Learning.*

Luukkonen, S., van den Maagdenberg, H., Emmerich, M. i van Westen, G. (2023). *Artificial intelligence in multi-objective drug design.*

Marcoulides, K. M. i Raykov, T. (2018). *Evaluation of Variance Inflation Factors in Regression Models Using Latent Variable Modeling Methods.*

Miernik, K., Węglińska, E., Danek, T. i Leśniak, A. (2021). *An application of the NSGA-II algorithm in Pareto joint inversion of 2D magnetic and gravity data.*

Miettinen, K. M. (1999). Nonlinear Multiobjective Optimization.

Ngatchou, P., Zarei, A. i El-Sharkawi, M. A. (2005). *Pareto Multi Objective Optimization.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Perrot, M. (2011). *Scikit-learn: Machine Learning in Python .*

Pugliese, R., Regondi, S. i Marini, R. (2021). *Machine learning-based approach: global trends, research directions, and regulatory standpoints.*

Sarkar, T., Salauddin, M., Mukherjee, A., Shariati, M., Rebezov, M., Tretyak, L., . . . Lorenzo, J. (2022). *Application of bio-inspired optimization algorithms in food processing.*

Seabold, S. i Perktold, J. (2010). *statsmodels: Econometric and statistical modeling with python.*

Srinivas, N. i Deb, K. (1994). *Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms.*

Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., . . . Woods, E. (2020). *Tslearn, A Machine Learning Toolkit for Time Series Data.*

Taylor, S. J. i Letham, B. (2017). *Forecasting at scale.*

Thangavel, P., Park, D. i Lee, Y. (2022). *Recent Insights into Particulate Matter (PM2.5)-Mediated Toxicity in Humans: An Overview.*

Vineeth, P., Babu, M. i Suresh, S. (2021). *Performance evaluation and analysis of population-based metaheuristics for denoising of biomedical images.*

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., . . . L. (2020). *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.*

Waring, J., Lindvall, C. i Umeton, R. (2020). *Automated machine learning: Review of the state-of-the-art and opportunities for healthcare.*

WHO. (2024, Marzec 11). *Air pollution*. Pobrano z lokalizacji Witryna sieci Web Światowej Organizacji Zdrowia: https://www.who.int/health-topics/air-pollution

Youngseob, E., Insang, S., Hwan-Cheol, K., Jong-Han, L. i Sun-Young, K. (2015). *Computation of geographic variables for air pollution prediction models in South Korea.*

Zaręba, M. i Danek, T. (2022). *Analysis of Air Pollution Migration during COVID-19.*

Zöller, M. i Huber, M. F. (2021). *Benchmark and Survey of Automated Machine Learning Frameworks.*