



**POLITECHNIKA
GDAŃSKA**

RURAL IoT: INTELIGENTNE POZYSKIWANIE I FUZJA DANYCH Z WIEJSKIEGO INTERNETU RZECZY

RAPORT TECHNICZNY

Nr umowy: WPN/4/68/Rural IoT/2022

Wykonawcy:

Krzysztof Drypczewski, Dariusz Klimowicz, Henryk Krawczyk, Piotr Orzechowski,
Łukasz Wiszniewski

Centrum Informatyczne Trójmiejskiej Akademickiej Sieci Komputerowej

Jerzy Dembski, Agata Kołakowska, Bogdan Wiszniewski

Wydział Elektroniki, Telekomunikacji i Informatyki

Dariusz Czajka

Avatar Technologies Sp. z o.o.



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI



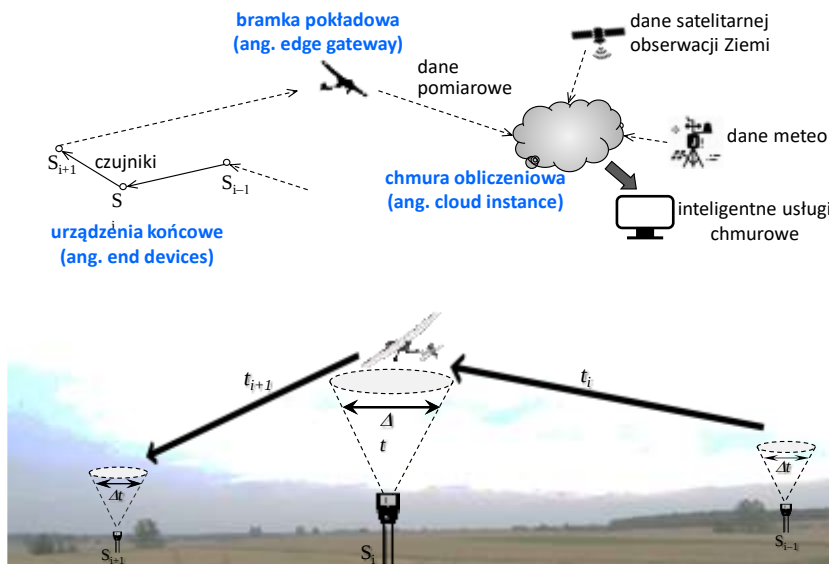
Spis treści

1.	Infrastruktura wiejskiego Internetu Rzeczy	4
2.	Inteligentny czujnik pomiarowy	5
2.1.	Moduł pomiarowy.....	6
2.2.	Moduł transmisji danych.....	8
2.3.	Moduł zasilania	8
2.4.	Oprogramowanie	8
3.	Mobilna bramka pokładowa	9
3.1.	Cykl pracy.....	9
3.2.	Budowa	10
4.	Chmura.....	11
4.1.	Infrastruktura bazowa	12
4.1.1.	Open Stack	12
4.1.2.	Ceph.....	13
4.2.	Akwizycja danych.....	13
4.2.1.	Baza danych pomiarowych	13
4.2.2.	Dane meteorologiczne	14
4.2.3.	Dane satelitarne.....	18
4.3.	Analiza danych pomiarowych	22
4.3.1.	Uzupełnianie brakujących próbek.....	23
4.3.2.	Detekcja anomalii.....	26
4.3.3.	Czyszczenie anomalii.....	29
4.3.4.	Optymalizacja rozmiaru danych	31
4.4.	Rozwój oprogramowania czujników	33
4.4.1.	Wersja początkowa detektora anomalii	33
4.4.2.	Wersje rozwojowe detektora anomalii	34
5.	Jakość danych pomiarowych	35
5.1.	Metody detekcji anomalii.....	35
5.1.1.	Heurystyczne	36
5.1.2.	Neuronowo-heurystyczne (NPH)	38
5.1.3.	Autokodery „binarne” (AEB)	39
5.1.4.	1D-Net (UNB)	41
5.2.	Ocena użyteczności.....	41

6.	Planowanie misji	45
6.1.	Charakterystyki operacyjne BSP	46
6.2.	Uwarunkowania prawne	47
6.3.	Wskazówki i dobre praktyki	48
6.4.	Uwarunkowania środowiskowe	49
6.5.	Klasy scenariuszy	51
	Literatura	53
	Słownik pojęć i skrótów	55

1. Infrastruktura wiejskiego Internetu Rzeczy

Demonstrator wiejskiego Internetu Rzeczy opracowany w projekcie Rural IoT stanowi pełnowymiarowy ekosystem pomiarowy, umożliwiający zbieranie danych in-situ wybranych parametrów gleby z rozległego obszaru nieposiadającego dostępnej infrastruktury telekomunikacyjnej (por. Rysunek 1).



Rysunek 1: Ekosystem pomiarowy Rural IoT

Trzy podstawowe komponenty tego ekosystemu obejmują:

- **Urządzenia końcowe** (*ang. end devices*) to zbiór $S = \{s_i, i = 1, \dots, n\}$ inteligentnych czujników pomiarowych rozmieszczonych na danym obszarze, takim jak pola uprawne czy kompleksy leśne. Każdy czujnik stanowi urządzenie autonomiczne, wyposażone w pamięć wewnętrzną i jednostkę obliczeniową w postaci mikrokontrolera (MCU), zasilanego z akumulatora doładowywanego w razie potrzeby z ogniwa fotowoltaicznego. Zadaniem każdego urządzenia końcowego jest regularny odczyt wartości parametrów gleby z umieszczonych w niej sensorów podłączonych do czujnika, zapisywanie tych danych w wewnętrznej pamięci w postaci szeregów czasowych próbek sygnałów reprezentujących mierzone wartości nieelektryczne oraz usunięcie z nich próbek niepoprawnych lub nadmiarowych, tak by w miarę możliwości przesyłany pakiet (*ang. payload*) zmieścił się w okienku czasowym Δt . Wielkość tego okienka oznacza przedział czasu, w którym przelatujący *bezzałogowy statek powietrzny* (BSP) znajduje się w zasięgu modułu transmisji danych, w jaki wyposażone jest urządzenie końcowe.
- **Bramkę pokładową** (*ang. edge gateway*) umieszczoną na pokładzie BSP pilotowanego z ziemi przez operatora lub wykonującego lot autonomicznie. BSP porusza się po możliwie najkrótszej trasie pokrywającej cały obszar S występowania czujników - z uwzględnieniem lokalizacji i siły sygnału każdego czujnika s_i oraz minimalnej długości okienka czasowego Δt_i wymaganej do przetransmitowania danych na pokład BSP. Zadaniem bramki pokładowej jest odbiór pakietów, ich zdekodowanie, usunięcie pakietów zdublowanych oraz ich zapisanie w pamięci pokładowej. Po wylądowaniu BSP dane te zostają przesłane do chmury obliczeniowej za pośrednictwem najbliższego dostępnego punktu sieci telekomunikacyjnej.
- **Chmurę obliczeniową** (*ang. cloud instance*), z osadzonymi na niej w formie stosu technologicznego elementami funkcjonalnymi obejmującymi oprogramowanie (narzędzia i usługi) przeznaczone do

realizacji trzech klas zadań: gromadzenia pakietów z próbkami sygnałów odczytanych z czujników, kodowania ich wartości do określonej formy binarnej i sekwencjonowania ich w szeregi czasowe próbek reprezentujących sygnał w formie przydatnej do dalszej analizy (1), interpretacji tych próbek poprzez wybór i porównywanie ich poszczególnych fragmentów z przebiegiem obwiedni sygnału wzdłuż jego większej całości (2) oraz rozwoju modeli uczenia maszynowego do analizy sygnałów, zasilanych danymi dostarczającymi przez BSP ze wszystkich czujników znajdujących się na monitorowanym obszarze S , a także danymi z innych źródeł (3). Modele te, po odpowiedniej redukcji (ang. knowledge distillation), mogą być instalowane na urządzeniach końcowych jako ulepszenia (aktualizacje) ich oprogramowania.

2. Inteligentny czujnik pomiarowy

Czujnik inteligentny pracuje nieprzerwanie w trybie 24/7, wykonując samodzielnie (bez komunikacji z innymi czujnikami): cykliczne odczyty i przetwarzanie wstępne danych pomiarowych oraz sporadyczne (zależnie od warunków pogodowych umożliwiających loty BSP) przesyłanie pakietów danych na pokład BSP. Operacje te realizuje za pomocą własnej jednostki obliczeniowej (MCU), stanowiącej kluczowy komponent *modułu pomiarowego* czujnika, sprzężonego z *modułem transmisji danych* – obu zasilanych przez *moduł zasilania* wykorzystujący akumulator doładowywany z ogniwa fotowoltaicznego.

Minimalny zestaw parametrów gleby podlegających cyklicznym pomiarom w punkcie umieszczenia czujnika w glebie obejmuje: *kwasowość* (pH), *temperaturę* (T) i *wilgotność* (M) oraz dodatkowo *nasłonecznienie* (PV). Ze względu na zmienne (trudne) warunki pogodowe dane in-situ zbierane z sensorów umieszczonych bezpośrednio w glebie mogą zawierać wartości odstające, niespójne, błędne, zaszumione czy też nieistotne z punktu widzenia dalszego ich przetwarzania w chmurze obliczeniowej. Z tego względu powinny zostać wstępnie przetworzone przez MCU czujnika inteligentnego w celu zmniejszenia rozmiaru pakietów danych przesyłanych przez moduł transmisji danych na pokład BSP. To z kolei wymaga opracowania architektury sprzętowej i systemowej czujnika, która będzie optymalna pod względem poboru energii dostarczanej jej przez moduł zasilania, przy założeniu określonych charakterystyk bazowych zastosowanych algorytmów przetwarzania wstępnego i technologii komunikacji bezprzewodowej. Zasady pomiaru wartości parametrów gleby przedstawia Tabela 1.

Tabela 1: Parametry gleby i metody ich pomiaru

Sygnał		Definicja	Jedn.	Zasada pomiaru
Nazwa	Symb.			
Kwasowość	pH	Liczba $\text{pH} = -\log(H^+)$, gdzie H^+ oznacza stężenie jonów wodorowych. Wg [1] $\text{pH}=7$ dla roztworów obojętnych, $\text{pH} < 7$ dla kwaśnych, $\text{pH} > 7$ dla zasadowych	[0,14]	Pomiar siły elektromotorycznej ogniwa złożonego ze szklanej elektrody wskaźnikowej i elektrody odniesienia umieszczonych w gruncie. Konieczna kalibracja poprzez pomiar siły elektromotorycznej dla dwóch lub więcej roztworów buforowych o ściśle znanych wartościach pH.
Temperatura	T	Dobowa temperatura gruntu na wybranej głębokości (ok. 1m)	°C	Pomiar rezystancji termistora umieszczonego w gruncie.
Wilgotność (objętościowa)	M	Stosunek objętości wody zawartej w glebie do całkowitej objętości gleby przed wysuszeniem.	%	Pomiar pojemności kondensatora w postaci płytki drukowanej umieszczonej w gruncie.
Nasłonecznienie	PV	Poziom światła dziennego warunkujący wytwarzanie prądu elektrycznego przez ogniwo fotowoltaiczne	V	Pomiar napięcia otwartego źródła napięciowego.

Schemat blokowy opracowanego czujnika inteligentnego opracowanego w projekcie przedstawia Rysunek 2; schemat ten obejmuje trzy wspomniane wcześniej moduły logiczne: *moduł pomiarowy* odpowiedzialny za odczyt danych z sensorów umieszczonych w glebie i ich przetwarzanie wstępne, *moduł transmisji danych*, które po przetworzeniu są przesyłane do bramki pokładowej drona oraz *moduł zasilania* umożliwiający całodobową pracę wszystkich układów czujnika.

2.1. Moduł pomiarowy

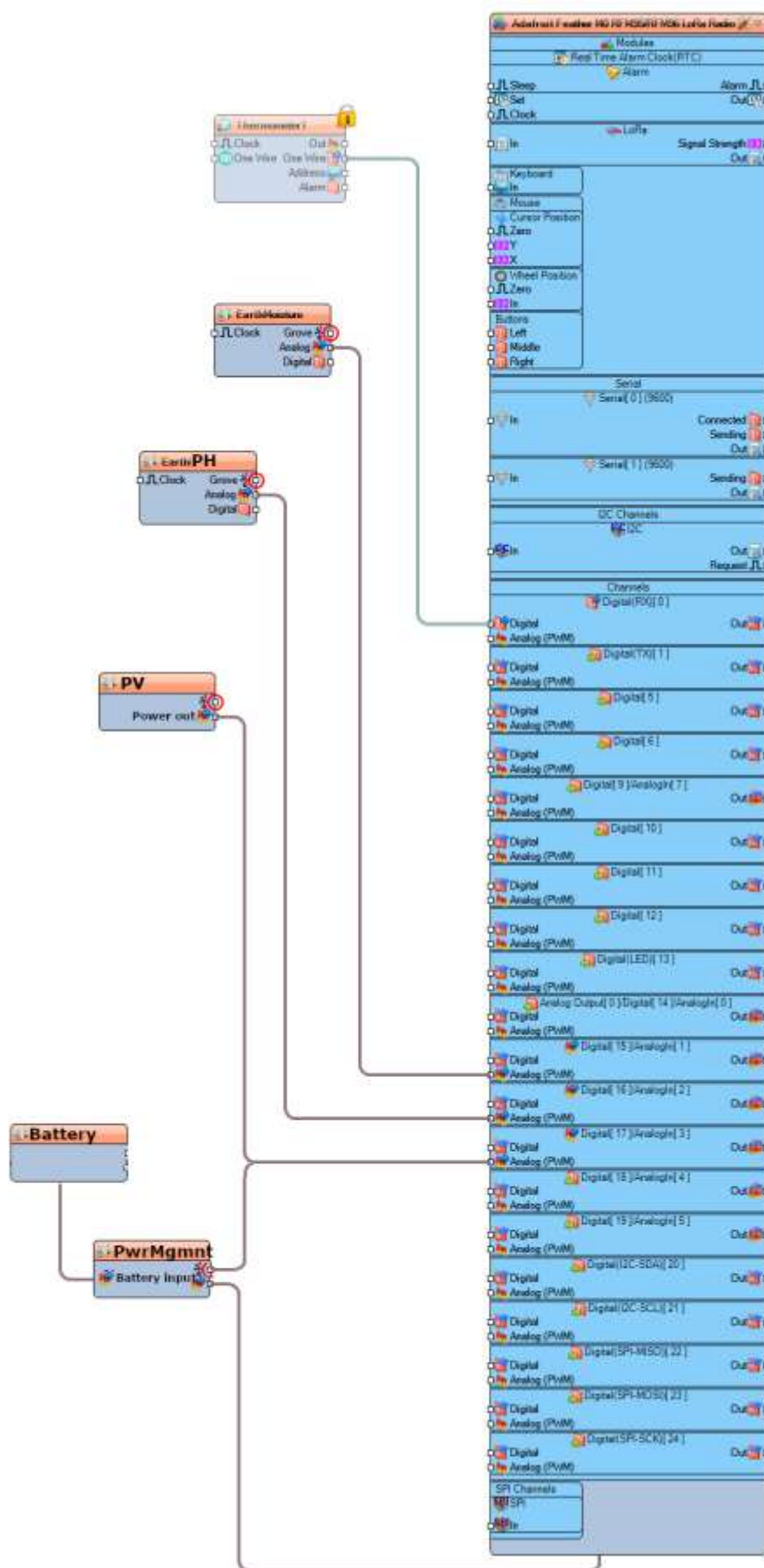
Moduł pomiarowy wykorzystuje 32 bitowy mikrokontroler kompatybilny ze środowiskiem Arduino [3], dostępny w formie gotowego zestawu produkowanego przez firmę Adafruit. Zestaw ten jest wyposażony w liczne wejścia i wyjścia analogowe oraz cyfrowe umożliwiające podłączenie wielu peryferyjnych czujników. W projekcie wykorzystano trzy wejścia (piny) analogowe oraz jedno cyfrowe, odpowiednio:

AnalogIn[1] – czujnik pomiarowy wilgotności gleby (**EarthMoisture**);

AnalogIn[2] – czujnik pomiarowy kwasowości gleby (**EarthPH**);

AnalogIn[3] - wyjście z ogniwa fotowoltaicznego (**PV**) do pomiaru nasłonecznienia;

Digital(Rx)[0] - cyfrowy termometr (**Thermometer1**) działający w protokole OneWire [4].



Rysunek 2: Schemat blokowy czujnika inteligentnego

Do użytku wewnętrznego zespołu wykonawców projektu

Procesor taktowany zegarem 64MHz dysponuje 256KB pamięci Flash na przechowywanie wykonywalnego kodu oprogramowania czujnika oraz 32KB pamięci RAM do przechowywania danych. Piny analogowe procesora podłączone są do wbudowanego 10-bitowego przetwornika analogowo-cyfrowego.

2.2. Moduł transmisji danych

Przesyłanie danych przetworzonych przez moduł pomiarowy wykorzystuje moduł RFM9x LoRa 868/915MHz, który umożliwia nawiązywanie połączenia z siecią LoRaWAN działającą w Europie w paśmie 868MHz. Moc wyjściowa modułu od +5 do +20 dBm (maks. 100mW) umożliwia uzyskanie łączności na dystansie od 2km do 20km – w zależności od zadanych ustawień w jego oprogramowaniu i rodzaju zastosowanej anteny. Pobór prądu podczas pracy waha się od szczytowej wartości ok. 100mA podczas nadawania do 30mA podczas nasłuchiwania.

2.3. Moduł zasilania

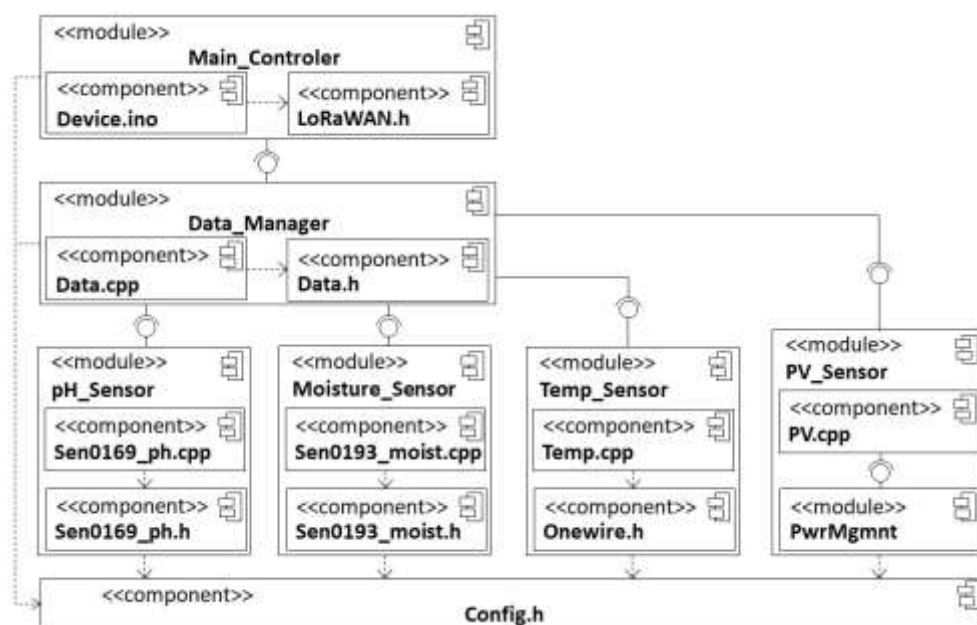
Moduł zasilania składa się z trzech elementów:

- ogniwa fotowoltaicznego (**PV**) o napięciu nominalnym 6 V,
- akumulatora litowo-polimerowej (**Battery**) o napięciu znamionowym 3,3 V i pojemności od 300 mAh do 500 mAh,
- menedżera zasilania (**PwrMgmt**).

Menadżer zasilania odpowiada za optymalną dystrybucję energii. Przy działającym ogniwie fotowoltaicznym zasila układ z pominięciem akumulatora. Jeśli napięcie akumulatora spada uruchamia również jego doładowanie. Z kolei przy zaniku napięcia z ogniwa fotowoltaicznego (np. silne zachmurzenie lub w nocy) przełącza zasilanie na akumulator.

2.4. Oprogramowanie

Strukturę oprogramowania czujnika inteligentnego przedstawia Rysunek 3.



Rysunek 3: Diagram komponentów oprogramowania czujnika inteligentnego

Zadania poszczególnych komponentów opisano poniżej:

Do użytku wewnętrznego zespołu wykonawców projektu

- **Main_Controller** steruje pobieraniem danych pomiarowych z czujników i ich wysyłaniem do bramki odbiorczej (**Device.ino**). Częstość wykonywania tych akcji jest zdefiniowana w pliku **Config.h** i zmienia się dynamicznie w następstwie zdarzeń LoRaWAN.
- **Data_Manager** obsługuje bufor danych pomiarowych z czujników, którego rozmiar ze zdefiniowanym w pliku **Config.h** jest systematycznie wypełniany danymi oraz opróżniany po wysłaniu paczki danych i potwierdzeniu jej odebrania. Dane pomiarowe pobierane są z podłączonych czujników temperatury (**Temp_Sensor**), pH (**pH_Sensor**), wilgotności (**Moisture_Sensor**) oraz na podstawie odczytu napięcia panelu solarnego zasilającego sondę pomiarową (**PV_Sensor**).
- **pH_Sensor** steruje odczytem pH gleby z sensora typu DFRobot (**Sen0169_ph**). Przed uruchomieniem należy go wstępnie skalibrować zmierzonymi referencyjnymi wartościami w roztworach o pH=4,0 (roztwór kwaśny) oraz pH=7,0 (roztwór neutralny).
- **Moisture_Sensor** steruje odczytem wilgotności gleby z sensora typu DFRobot (**SEN0193_moist**). Przed uruchomieniem należy go wstępnie skalibrować zmierzonymi wartościami referencyjnymi dla wilgotności 0% i 100%.
- **Temp_Sensor** steruje pomiarem temperatury z cyfrowego termometru Dallas DS18B20, który komunikuje się za pomocą protokołu OneWire (parametry zdefiniowane w **Onewire.h**). Termometr nie wymaga kalibracji, po jego uruchomieniu można odczytywać dane.
- **PV_Sensor** realizuje bezpośredni pomiar napięcia ogniwa fotowoltaicznego. Pomiar odbywa się za pomocą wbudowanego w układ Adafruit przetwornika analogowo-cyfrowego (por. Rysunek 2).
- **LoRaWAN.h** zawiera definicje parametrów dostępowych do sieci AppSKey (Application Session Key) i NwksKey (Network Session Key). Ze względu na brak stałej obecności bramki w zasięgu czujnika używany jest protokół dostępowy ABP. Każda sonda pomiarowa ma w pliku **LoRaWAN.h** wprowadzony swój unikalny adres DevAddr.
- **Config.h** zawiera definicje parametrów użytkowe zachowywania się sondy pomiarowej takie jak wielkość bufora danych oraz częstotliwość pobierania i wysyłania danych.

Ponieważ czujniki przez większość czasu znajdują się poza zasięgiem bramki mobilnej BSP moduł **Data_Manager** implementuje mechanizm buforowania zebranych danych pomiarowych a moduł **Main_Controller** realizuje próby wielokrotnego ich wysyłania. Bufor o parametryzowanej wielkości może gromadzić dane zebrane w okresie ostatnich kilku/kilkunastu godzin, po czym starsze dane są nadpisywane przez nowe. Co ustalony dłuższy okres czasu (rzędu 10-20 minut) podejmowana jest próba wysłania danych do bramki mobilnej w trybie ABP (bez konieczności wcześniejszej rejestracji) z żądaniem potwierdzenia odbioru. Gdy przelatujący w pobliżu czujnika BSP potwierdzi odebranie pierwszej paczki danych, czujnik wysyła kolejne zgromadzone w buforze dane pomiarowe z dużo większą częstotliwością, aż do całkowitego opróżnienia bufora. W ten sposób jest możliwe wysyłanie danych pomiarowych zebranych podczas możliwie długiego okresu czasu pomiędzy kolejnymi lotami pomiarowymi BSP.

3. Mobilna bramka pokładowa

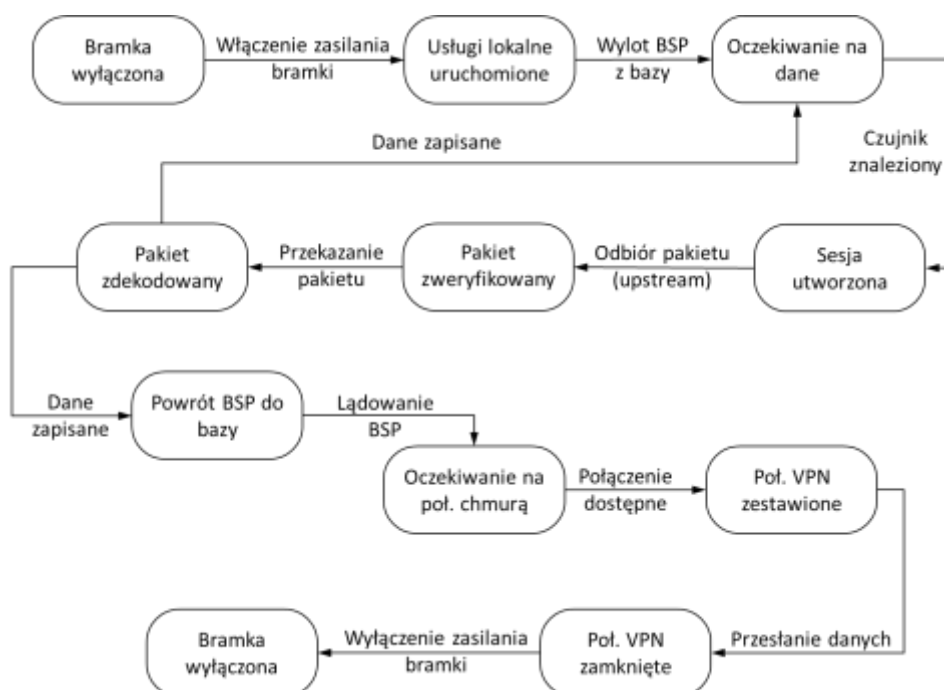
3.1. Cykl pracy

Diagram stanów przedstawiony na Rysunku 4 opisuje pełny cykl pracy bramki mobilnej od jej uruchomienia, poprzez start i przelot BSP po zadanej trasie, aż po lądowanie i załadowanie danych do chmury obliczeniowej. Lokalne usługi uruchamiane po włączeniu bramki realizowane są przez poszczególne komponenty jej oprogramowania, przedstawione na Rysunku 5 i omówione dalej.

Metoda tworzenia sesji (ang. *session establishment*) w opisanym w tym raporcie ekosystemie Rural IoT wymaga dodatkowego wyjaśnienia. W sieci LoRaWAN urządzeń końcowe mogą być aktywowane poprzez protokół ABP (ang. Activation By Personalization) albo OTAA (ang. Over The Air Activation) [5].

W metodzie OTAA adres oraz klucze sesji musiałyby być każdorazowo generowane i wymieniane pomiędzy czujnikiem i bramką. Ponieważ przez większość czasu czujniki w ekosystemie Rural IoT znajdują się poza zasięgiem bramki (i sieci), nie ma potrzeby ustawicznego nawiązywania połączenia z bramką – co powodowałoby m.i. znaczne zużycie energii zasilania czujnika. Dla jej oszczędzania należałoby zaplanować dla wszystkich czujników ekosystemu stały harmonogram usypiania i włączania prób nawiązywania połączenia zsynchronizowanych z przewidywanym harmonogramem lotów BSP przenoszących bramkę. W metodzie ABP adres czujnika (**DevAddr**) oraz jego klucze (**NwksKey**, **AppSKey**) są na stałe zapisane zarówno w kodzie czujnika, jaki i bramki i nie muszą być każdorazowo generowane podczas nawiązywania połączenia. Chociaż metoda ABP jest podatna na próby podszycia się pod czujniki ekosystemu Rural IoT przez obce urządzenia, to brak w nim stacjonarnych bramek oraz zamknięty charakter (żadne komunikaty z czujników nie są przesyłane przez obcą infrastrukturę), jest ona w tym zastosowaniu wystarczająco bezpieczna. Aktywacje i transmisje z czujników nie będą wymagały synchronizacji harmonogramów czujników i bramek.

Weryfikacja i dekodowanie pakietu pobranego z czujnika realizowana jest odpowiednio przez komponenty **Network_Server** i **Application_Server** opisane dalej.

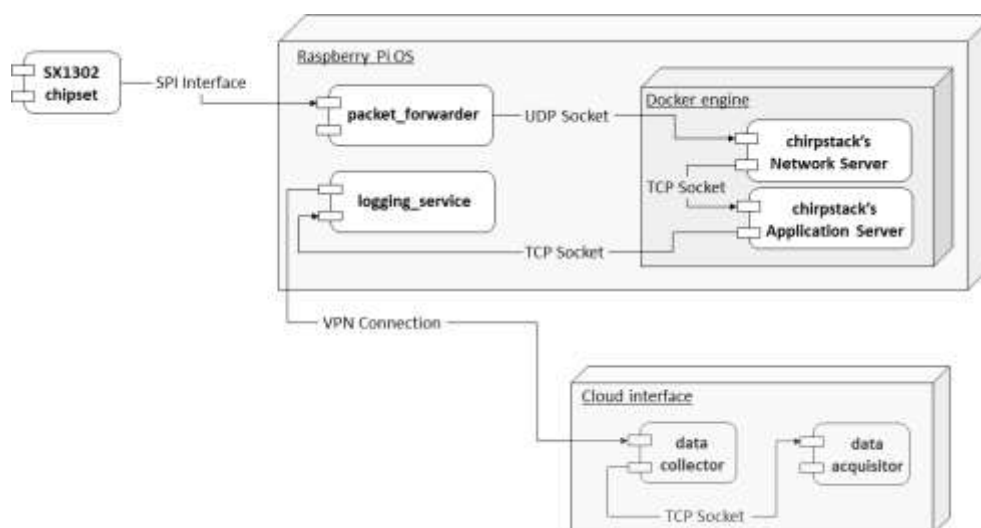


Rysunek 4: Diagram stanów bramki pokładowej (mobilnej) BSP

3.2. Budowa

Diagram wdrożenia komponentów oprogramowania bramki pokładowej przedstawia Rysunek 5. Wszystkie wykazane komponenty oprogramowania obsługują tor połączenia od odebrania danych z czujnika po archiwizację w postaci gotowej do zapisania w repozytorium chmury, jak przedstawiono wcześniej na Rysunku 4. Całość tego oprogramowania chodzi pod kontrolą systemu operacyjnego Raspbian/Raspberry Pi RaspberryPi, opartego na Debianie, opracowanego specjalnie dla komputerów jednopłytkowych Raspberry Pi.

Do użytku wewnętrznego zespołu wykonawców projektu



Rysunek 5: Diagram wdrożenia komponentów oprogramowania bramki pokładowej

Zadania poszczególnych komponentów bramki pokładowej opisano poniżej:

- **SX1302** (układ scalony) umożliwia komunikację pomiędzy urządzeniami LoRaWAN a systemem operacyjnym Raspberry Pi OS za pośrednictwem interfejsu SPI (ang. *Serial Peripheral Interface*), standardowego szeregowego interfejsu komunikacyjnego pomiędzy systemami mikroprocesorowymi a układami peryferyjnymi.
- **Packet_forwarder**, jako daemon systemowy odpowiada za implementację logiki działania bramki. Odebrane przez niego pakiety są przesyłane z wykorzystaniem protokołu UDP do serwera **Network Server**.
- **Network_Server** odbiera pakiety przesłane przez **Packet_forwarder** i dokonuje ich zdekodowania i przekazuje do serwera **Application Server**.
- **Application_Server** odpowiada za końcowe zdekodowanie danych przesłanych przez czujniki i przekazanie ich do **Logging_service**. **Network_Server** i **Application_Server** uruchomione są w swoich dedykowanych kontenerach.
- **Logging_service**, którego implementacja bazuje na serwerze WWW **nginx**, zapisuje wszystkie dane odebrane od czujników, a gdy wykryje istnienie połączenia internetowego przesyła wszystkie zgromadzone dane do segmentu chmurowego (**Cloud interface**).
- **Data_collector** pobiera dane odebrane przez bramkę i przekazuje je do **Data_acquisitor**.
- **Data_acquisitor** archiwizuje odebrane dane w dedykowanym wolumenie.

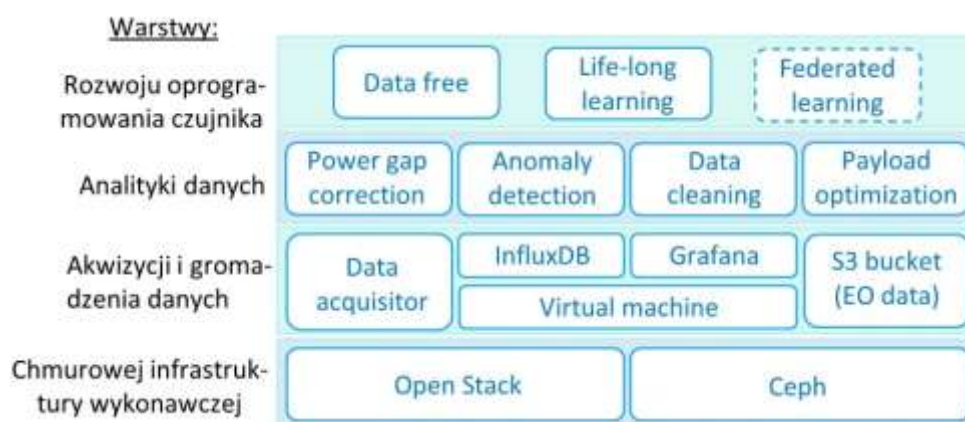
4. Chmura

Rysunek 6 przedstawia kluczowy komponent demonstratora ekosystemu Rural IoT, którym jest stos technologiczny **Collaborative Learning as a Service (CLaaS)** [8]. Jego poszczególne warstwy dostarczają:

- Bazową infrastrukturę wykonawczą (ang. *cloud instance layer*), zapewniającą niezbędne instancje obliczeniowe, połączenia sieciowe i pamięć masową;
- Usługi akwizycji i gromadzenia danych (ang. *data collection layer*) do zarządzania repozytoriami danych pomiarowych in-situ z czujników, danych pogodowych i danych obrazowych;
- Usługi analityki danych (ang. *data analytics layer*) do profilowania i czyszczenia szeregów czasowych próbek pomiarowych;

Do użytku wewnętrznego zespołu wykonawców projektu

- Usługi ustawicznego rozwoju modeli uczenia maszynowego (ang. *model development layer*) umożliwiających rozwój i doskonalenia oprogramowania inteligentnych czujników ekosystemu w miarę gromadzenia w nim wiedzy o monitorowanych zjawiskach.



Rysunek 6: Stos technologiczny CLaaS chmury ekosystemu Rural IoT

4.1. Infrastruktura bazowa

Warstwę bazową infrastruktury wykonawczej stanowi rozwijana w CI TASK od prawie 10 lat chmura obliczeniowa TASKcloud. Została ona zbudowana w pełni w oparciu o rozwiązania open source zgodnie z pierwotnymi założeniami unikania uzależnienia świadczonych usług od dostawców sprzętu i oprogramowania (ang. *vendor lock-in*).

Podstawową warstwą chmury TASKcloud jest warstwa systemów operacyjnych z rodziny Linux a konkretnie system operacyjny Ubuntu. Kolejną warstwą jest narzędzie konteneryzacji Docker zainstalowane na wszystkich serwerach, które zwiększa poziom abstrakcji i tworzy uniwersalną warstwę, na której wdrażane jest oprogramowanie zapewniające funkcjonalności oferowane w TASKcloud.

4.1.1. Open Stack

Bazowym oprogramowaniem wykorzystywanym w chmurze TASKcloud jest oprogramowanie Open Stack [27]. Jest to produkt istniejący na rynku od 2010 rozwijany przez społeczność, którą tworzą zarówno zespoły z firm liderów branży IT jakimi są, np. OVH, Rackspace, Intel jak również indywidualni deweloperzy.

Oprogramowanie zapewnia funkcjonalność chmury obliczeniowej w modelu IaaS (Infrastructure as a Service). Model ten zapewnia użytkownikowi warstwę wirtualizacji zasobów i największą elastyczność w porównaniu z modelami SaaS (Software as a Service) czy PaaS (Platform as a Service). Użytkownik chmury dzięki dostarczonym funkcjonalnościom może tworzyć wirtualne sieci, konfigurować je na uruchomionych przez niego wirtualnych maszynach, do których może dołączyć wirtualne dyski. Dodatkowo OpenStack zapewnia szereg funkcji pozwalających na zabezpieczenie i sterowanie ruchem sieciowym co podnosi bezpieczeństwo uruchamianej infrastruktury obliczeniowej.

W chmurze ekosystemu Rural IoT wykorzystano warstwę IaaS do przygotowanie wirtualnej warstwy obliczeniowej i gromadzenia danych. Udostępniono maszyny wirtualne, na których zainstalowano warstwę akwizycji danych i uruchomiono opracowane skrypty gromadzenia i przetwarzania danych.

4.1.2. Ceph

Istotnym elementem warstwy chmurowej jest dostarczenie wolumenów (wirtualnych dysków do maszyn wirtualnych). Dane są najważniejszym zasobem, który w szczególny sposób musi podlegać zabezpieczeniu przed ich utratą. W tym celu OpenStack wykorzystuje integrację z systemem magazynu danych **Ceph**.

Ceph, jako oprogramowanie otwartoźródłowe, umożliwia przechowywanie danych w kilku kopiach rozproszonych pomiędzy różne dyski na różnych serwerach. Dzięki takim założeniom dane są odporne na awarie pojedynczych dysków jak też całego serwera, a w przypadku wystąpienia którejś z nich Ceph potrafi sam przeprowadzić tzw. rebalancing na inne zasoby (alternatywny dysk lub serwer), tak by dane zawsze miały określoną liczbę kopii.

Zasoby magazynu danych są udostępniane przez Ceph na dwa sposoby. Możliwe jest udostępnianie wolumenów danych, które następnie są montowane przez maszyny wirtualne i widziane przez system operacyjny jako urządzenia blokowe, jak również jako tzw. magazyn obiektowy, czyli zasób dostępny po protokole http ze zdefiniowanym API, dzięki czemu możliwe są operacje na danych (ich gromadzenie i odczytywanie) z dowolnego miejsca. Magazyn blokowy został wykorzystany do gromadzenia danych satelitarnych (por. p.4.2.2).

4.2. Akwizycja danych

Usługi realizowane przez centralną chmurę obliczeniową ekosystemu Rural IoT (wymienione m.i. na Rysunku 1) wymagają pozyskiwania danych z wielu źródeł. Podstawowym źródłem są oczywiście *dane pomiarowe* zbierane przez BSP z czujników rozmieszczonych na monitorowanym obszarze. Po ich dostarczeniu do chmury w pierwszej kolejności przeprowadzana jest fuzja danych (szeregów czasowych) pochodzących z wielu czujników w celu usunięcia z każdego pojedynczego szeregu wszystkich tych anomalii, których dany pojedynczy czujnik nie był w stanie wykryć, ani tym bardziej wyeliminować. W następnej kolejności, oczyszczone dane są wykorzystywane do douczania modelu bazowego detektora zaimplementowanego przez każdy czujnik. Dzięki temu douczaniu, każda udoskonalona wersja oprogramowania czujnika kolejnej (nowej) generacji będzie charakteryzowała się wyższą skutecznością detekcji anomalii. W przypadku zaimplementowania w oprogramowaniu czujnika także możliwości odczytu danych z bramki pokładowej BSP jest możliwa okresowa aktualizacja oprogramowania czujników już zainstalowanych w ekosystemie. Wreszcie dla realizacji najbardziej zaawansowanych scenariuszy użytkowych (np. predykcji plonów) w ekosystemie Rural IoT możliwa jest fuzja danych pomiarowych z *danymi meteorologicznymi*, lub nawet z *danymi satelitarnej* obserwacji Ziemi. W opisaney w tym raporcie implementacji ekosystemu Rural IoT zbadano możliwość pozyskiwania danych meteorologicznych i danych satelitarnych, jednak wspomnianych scenariuszy predykcyjnych nie realizowano.

4.2.1. Baza danych pomiarowych

Niezbędnym elementem komponentu chmurowego ekosystemu Rural IoT jest warstwa gromadzenia danych, które mogą być wykorzystywane do różnych analiz. Dostarcza ona rozwiązanie, które jest łatwe do integracji z oprogramowaniem odpowiedzialnym za wysyłanie danych, pozwala w wygodny sposób prezentować metryki w postaci wykresów, jak też zapewnia programistyczny dostęp na potrzeby automatycznej analizy danych. Powyższe wymagania spełniają z nadmiarem dwa wykorzystane narzędzia otwartoźródłowe:

- **InfluxDB** - baza danych do gromadzenia, zarządzania i udostępniania wielkich wolumenów danych w postaci szeregów czasowych (ang. *time series data base*, TSDB), takich jak metryki monitorowania, dane z czujników IoT i dane dotyczące wydajności aplikacji [28];
- **Grafana** - platforma do wyszukiwania, wizualizacji i analizowania logów, umożliwiająca tworzenie niestandardowych pulpitów nawigacyjnych z wykresami do wizualnej analizy danych w czasie rzeczywistym [29].

Oba narzędzia dobrze się ze sobą integrują i współpracują. Punktem wyjścia do przygotowania bazy danych pomiarowych jest maszyna wirtualna uruchomiona na chmurze TASKcloud z publicznym adresem IP, zamontowanym wolumenem danych z replikacją (**/data**), przeznaczonym do zapisywania danych oraz zainstalowanym systemem operacyjnym Debian wraz z oprogramowaniem Docker [31]. W celu uruchomienia oprogramowania InfluxDB na serwerze należy pobrać jego kontener i przy uruchamianiu podać zmienne startowe, które zdefiniują, gdzie na maszynie wirtualnej mają być składowane dane oraz przygotują konto administratora i ustawią hasło. Przykładowe polecenie do uruchomienia InfluxDB na porcie 8086 i składającego dane na maszynie w zamontowanym wolumenie **/data** prezentuje listing poniżej:

```
docker run -p 8086:8086 \
-v $PWD/data:/data/influxdb \
-e INFLUXDB_ADMIN_USER=user \
-e INFLUXDB_ADMIN_PASSWORD=password \
-e INFLUXDB_DB=ruraliot \
influxdb:1
```

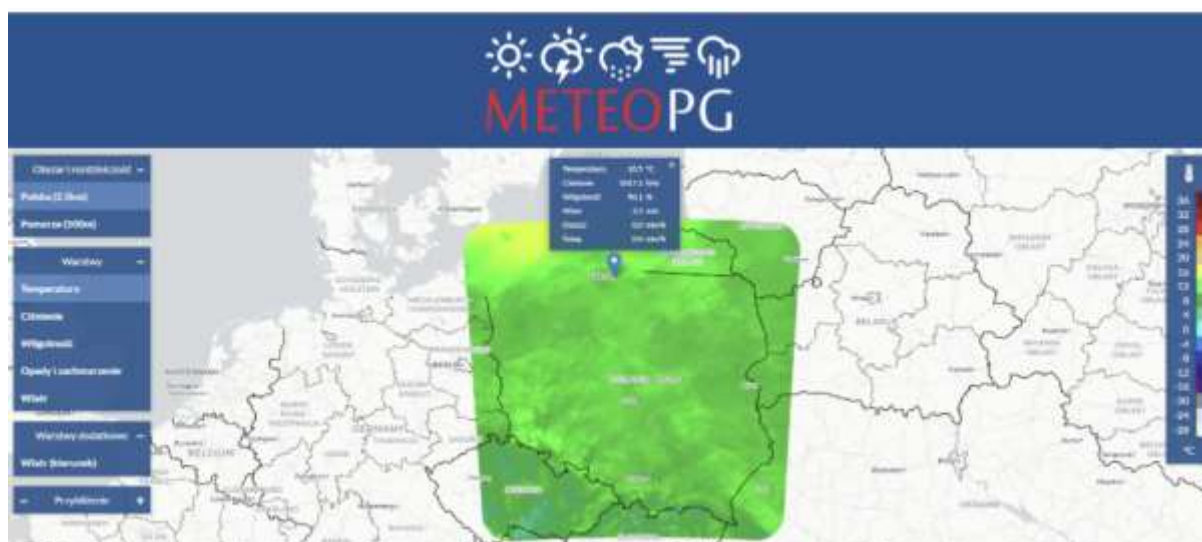
W podobny sposób uruchamiany jest kontener z oprogramowaniem Grafana, które będzie działało na porcie 3000 i przechowywało dane na lokalnym dysku maszyny wirtualnej w katalogu **/data**:

```
docker run -p 3000:3000 \
-v $PWD/data:/data/grafana \
grafana/grafana
```

Po uruchomieniu kontenerów należy wejść z przeglądarki na komputerze posiadającym dostęp do maszyny wirtualnej, wpisać adres IP maszyny (np. 192.168.1.1) i podać port Grafany, np. <http://192.168.1.1:3000>. W przeglądarce pojawi się panel logowania do grafany, gdzie można zalogować się jako użytkownik **admin** z hasłem **admin**.

4.2.2. Dane meteorologiczne

Możliwość systematycznego gromadzenia danych meteorologicznych razem z danymi pomiarowymi zbieranymi z czujników gleby może być istotna z punktu widzenia realizacji różnych scenariuszy predykcyjnych, związanych planowaniem działań na monitorowanym obszarze. Stąd pojawia się konieczność wykorzystania dostępnych serwisów pogodowych. W ekosystemie Rural IoT wykorzystano system MeteoPG, uruchomiony na zasobach obliczeniowych CI-TASK PG (por. Rysunek 7).



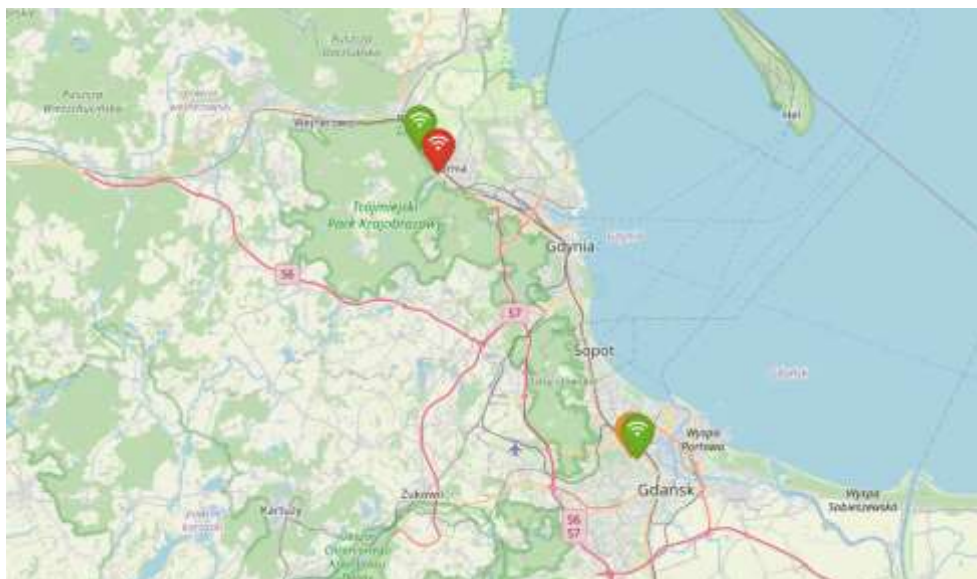
Rysunek 7: Zrzut ekranu strony internetowej portalu pogodowego MeteopG <https://meteopg.pl>

Prognozy wyliczane przez system MeteopG opierają się na modelu krótkoterminowym (72h), którego dokładność i wysoka gęstość siatki predestynuje go do wykorzystania w zastosowaniach rolniczych. Dane w tym modelu pochodzą z wysokorozdzielczej wersji mezoskalowego modelu HRWRF 3.8.1/3.9.1.1 (ang. High Resolution Weather Research and Forecasting). W HRWRF zastosowano zoptymalizowaną parametryzację fizyki pogody dla terenu Europy Środkowej. Wykorzystano również wysokorozdzielcze dane przestrzenne zawierające topografię, szorstkość podłoża, zagospodarowanie i pokrycie terenu, które pozyskano z krajowych i zagranicznych baz danych. Zgromadzone obecnie wysokorozdzielcze dane pozwalają prowadzić obliczenia metodą siatek zanurzonych do rozdzielczości nawet 100m [30]. Do prognozowania pogody wykorzystywane są dane z globalnego modelu GFS (ang. Global Forecast System) o rozdzielczości 0.25°, który zawiera w sobie sprzężone modele atmosfery, gleby, oceanu i lodu morskiego. Model w ciągu doby uruchamiany jest dla czterech głównych terminów synoptycznych 00, 06, 12 i 18 z czasem prognozy wynoszącym 60 godzin. System obliczeniowy prognozowania pogody pracuje na superkomputerze "TRYTON" w CI TASK, wykorzystując kilkaset rdzeni obliczeniowych procesorów Haswell.¹

Dostęp do danych; dostęp do danych meteorologicznych z modelu MeteopG nie jest dostępem otwartym i nie ma możliwości pobierania danych po API z udostępnionego ogólnie endpointu. Formę wymiany danych można jednak uzgodnić z zespołem MeteopG². W opisywanej w tym raporcie implementacji ekosystemu Rural IoT bazowano na plikach źródłowych z prognozami pogody w formacie **.wrf** (wynik działania modelu WRF na klastrze obliczeniowym). Dane dla pojedynczego obszaru były spakowane w postaci **.tar** – każda z prognoz znajdowała się w oddzielnym pliku. Interesującym obszarem był obszar, gdzie zostały zainstalowane czujniki LoRaWAN tj. obszar Rumii i Tczewa. Dla zadanego obszaru model wyliczany jest z bardzo wysoką dokładnością tj. z siatką 500 m (por. Rysunek 8).

¹ <https://meteopg.pl/#/description> (dostęp: 14.07.2025 r.)

² meteopg@pg.edu.pl



Rysunek 8: Zrzut ekranu mapy systemu Chirpstack (<https://lorawan.apl.task.gda.pl>) z zaznaczonymi czujnikami

Wybór i gromadzenie danych; Proces ekstrakcji danych z prognoz opartych o model WRF przebiega przy założeniu, że dane zostały zapisane w formacie **NetCDF4**. Poniższy przykład ilustruje jego przebieg z wykorzystaniem języka Python. Aby odczytać interesujące dane z otrzymanych plików należy wykorzystać bibliotekę **netcdf4-python**³, a następnie za pomocą biblioteki **wrf**⁴ znaleźć elementy siatki, które odpowiadają interesującym nas współrzędnym geograficznym.

```
import netCDF4
from wrf import ll_to_xy

def read_forecast_data(ncfile, in_lat, in_lon):
    times = wrf.extract_times(ncfile, wrf.ALL_TIMES, meta=False)
    x_y = ll_to_xy(ncfile, in_lat, in_lon, wrf.ALL_TIMES, meta=False)
    (...)

coordinates = [
    (54.5765651, 18.3303489),
    (54.160255, 18.814006)
]

ncfile = netCDF4.Dataset('/data/wrf-forecast/wrfout_d02_2023-04-01_00:00:00')

for point in coordinates:
    read_forecast_data(client, ncfile, point[0], point[1])
```

Posiadając informacje (obiekt `x_y`) o elementach siatki zawierających prognozy dla interesujących nas punktów (zawartych w tablicy `coordinates`) możemy odczytać wartości poszczególnych parametrów prognozy. Poniżej zaprezentowano przykład dla odczytu temperatury, ciśnienia i zachmurzenia niskiej warstwy.

³ <https://unidata.github.io/netcdf4-python/> (wer. 1.7.1.post1)

⁴ <https://wrf-python.readthedocs.io/en/latest/> (wer. 1.3.4.1)

```

from wrf import getvar

def read_temperature(ncfile, x_y):
    temperature = getvar(ncfile, "tc", timeidx=wrf.ALL_TIMES,
meta=False)
    return temperature[:, 0, x_y[0], x_y[1]]

def read_pressure(ncfile, x_y):
    pressure = getvar(ncfile, "pressure", timeidx=wrf.ALL_TIMES,
meta=False)
    return pressure[:, 0, x_y[0], x_y[1]]

def read_cloudfrac(ncfile, x_y, level):
    cloudfrac = getvar(ncfile, "cloudfrac", timeidx=wrf.ALL_TIMES,
meta=False)
    return cloudfrac[level, :, x_y[0], x_y[1]]

```

Do zapisania danych w bazie wykorzystujemy obiekt pomocniczy reprezentujący pojedyncze prognozy o nazwie **Sample**.

```

from wrf import getvar

class Sample(object):
    name = None
    lat = None
    long = None
    temperature = None
    pressure = None
    timestamp = None
    cloudfrac_low = None

    def json(self):
        return [
            {
                "measurement": "weather_forecast",
                "tags": {
                    "latitude": self.lat,
                    "longitude": self.long,
                    "sensor_name": self.name
                },
                "time": str(self.timestamp),
                "fields": {
                    "temperature": self.temperature,
                    "pressure": self.pressure,
                    "cloudfrac_low": self.cloudfrac_low,
                    "rainc": self.rainc
                }
            }
        ]

```

Następnie należy odczytać godziny, dla których w danym pliku zapisane są prognozy oraz utworzyć dla każdej prognozy obiekt **Sample**, zapisywany do bazy w formacie JSON.

```

times = wrf.extract_times(ncfile, wrf.ALL_TIMES, meta=False)

with InfluxDBClient('10.0.0.1', 8086, 'user', 'password', 'database')
as influxClient:
    i = 0
    for time in times:

        s = Sample()
        s.name = point_name
        s.timestamp = time
        s.lat = in_lat
        s.long = in_lon
        s.temperature = temperature[i]
        s.pressure = pressure[i]
        s.cloudfrac_low = cloudfrac_low[i]

    influxClient.write_points(s.json())
    i+=1

```

W ten sposób zebrano w bazie danych następujące parametry pogody: czas (ang. timestamp), zachmurzenie w 3 warstwach chmur: niskiej, średniej i wysokiej (w %), szerokość i długość geograficzną punktu, nazwę punktu, ciśnienie, opady i temperaturę. Przykładowo rekord ten może wyglądać:

time	cloudfrac_high	cloudfrac_low	cloudfrac_medium	latitude
	longitude	pressure	rainc sensor_name temperature	
----	-----	-----	-----	-----
1680307200000000000	0.17411097884178162	0.8777816891670227	0.8470022678375244	54.120755

W okresie od 1. kwietnia do 31. grudnia 2023 r. pozyskano i wprowadzono do bazy danych dane surowe (format RAW), stanowiące bezpośredni wynik działania modelu dla obszaru na Rysunku 8; w postaci spakowanej miały rozmiar 227 GB. Dane te były dostępne do badań na dwa sposoby: za pomocą API i bibliotek, którymi możliwe było pobranie danych w skryptach analizy danych, a także wizualnie za pomocą narzędzia Grafana umożliwiającego ich podgląd (por. Rysunek 6). Przykładowy wykres dla wybranych parametrów pogody za okres kwiecień-czerwiec 2023 r. przedstawia Rysunek 9.



Rysunek 9: Wizualizacja parametrów pogody za pomocą narzędzia Grafana

4.2.3. Dane satelitarne

Modele predykcyjne realizowane na chmurze obliczeniowej ekosystemu Rural IoT zasilane być mogą także danymi obrazowymi, pochodzącymi z misji Sentinel-2 programu Copernicus,

prowadzonego przez Europejską Agencję Kosmiczną (ESA) we współpracy z Komisją Europejską. Satelita tego programu jest wyposażony w czujnik multispektralny dostarczający zobrazowania w 13 pasmach widmowych o rozdzielczości przestrzennej od 10 m do 60 m na piksel, a dostarczane przez niego dane można pozyskiwać z europejskiej platformy Copernicus Data Space Ecosystem (CDSE) [15]. Na potrzeby projektu Rural IoT utworzono repozytorium danych satelitarnych z misji Sentinel-2, zawierające zobrazowania przedstawiające miejsca umieszczenia czujników (okolice miast Rumia i Tczew). Założono, że repozytorium tym znajdują się dane:

- Surowe na poziomie 2A, BOA (ang. Bottom-Of-Atmosphere) czyli po korekcji atmosferycznej.
- Zdjęcia RGB utworzone z pasma widzialnego (pasma: B4, B3, B2) całej sceny.
- Zdjęcia RGB utworzone z pasma widzialnego (pasma: B4, B3, B2) obszaru Rumi i Tczewa, jeśli nie są zasłonięte chmurami, wycięte z w/w sceny o wielkości 1000x1000 pikseli.

Utworzenie takiego repozytorium obejmuje następujące etapy:

1. **Dostęp do danych;** pobranie danych z CDSE wymaga pozyskania danych dostępowych:

- konto użytkownika w CDSE, który umożliwia dostęp do systemu przez przeglądarkę;
- rejestracja klienta protokołu delegowania zarządzania dostępem do zasobów OAuth powiązanego z kontem użytkownika (p.1a). Operacja ta jest konieczna w celu selekcji danych do pobrania. Szczegółowy opis tego procesu jest dostępny pod <https://documentation.dataspace.copernicus.eu/APIs/SentinelHub/Overview/Authentication.html>;
- klucze API obiektowej pamięci masowej zgodnej ze standardem S3, służące do uwierzytelniania do repozytorium CDSE. Szczegółowy opis procedury dostępu można znaleźć pod <https://documentation.dataspace.copernicus.eu/APIs/S3.html>;

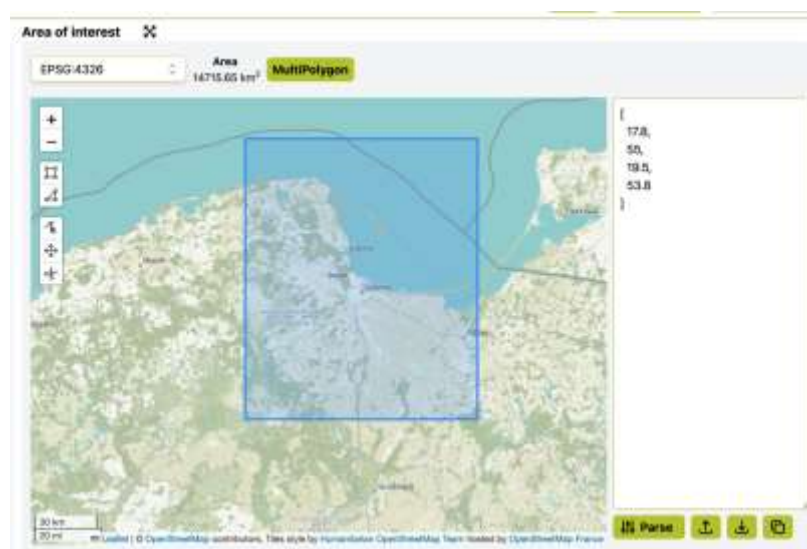
2. **Wyszukanie danych;** CDSE udostępnia szereg API dedykowanych do przeszukiwania dostępnych zbiorów danych, oparte na otwartych standardach OData [16], STAC API [17] i OpenSearch [18]. Sentinel Hub Catalog (zwany w skrócie „Catalog”) dostarcza interfejs API [19] implementujący specyfikację STAC i umożliwia wyszukiwanie i pobieranie informacji geoprzestrzennej z Sentinel Hub na podstawie szeregu kryteriów, w tym obszaru widocznego na zdjęciu, czasu wykonania obrazu czy obszaru podlegającego obserwacji. Procedura wyszukiwania danych wykorzystująca Catalog API wygląda następująco:

- rozpoczęcie sesji, uwierzytelnienie OAuth,
- przesłanie kwerendy do API,
- odebranie w odpowiedzi adresów do pobrania spełniających kwerendę zobrazowań. Zważywszy na to, że dane są stronicowane, konieczne jest wielokrotne odpytanie API aż do otrzymania wszystkich wyników.

3. Przykładowa konfiguracja zapytania o dane z misji Sentinel-2 za okres od kwietnia do października 2023 r. przedstawiających Trójmiasto i okolice, gdzie pokrycie chmurami jest mniejsze niż 50% wygląda następująco:

```
data = {
  "bbox": [17.8, 55, 19.5, 53.8],
  "datetime": "2023-04-01T00:00:00Z/2023-10-30T23:59:59Z",
  "collections": ["sentinel-2-l2a"],
  "limit": 10,
  "filter": "eo:cloud_cover < 50",
}
```

Zapytanie zwraca dane dla obszaru przedstawionego na Rysunku 10.



Rysunek 10: Obszar dla którego przechowywane są dane obrazowe w repozytorium Rural IoT

Dokumentacja Sentinel Hub Catalog API jest dostępna pod adresem: <https://documentation.dataspace.copernicus.eu/APIs/SentinelHub/Catalog.html>. Pobranie danych jest możliwe za pomocą dowolnego narzędzia wspierającego format S3. Ze względu na znaczną wielkość poszczególnych produktów satelitarnych (zwykle około 0,5-1GB) oraz konieczność pobrania szeregu takich produktów zalecane jest przygotowanie skryptu, który wykona pobranie danych bez potrzeby ciągłej interakcji z użytkownikiem, przy pomocy takich poleceń jak *aws-cli* czy *s3cmd*.

4. **Tworzenie obrazów RGB;** ESA udostępnia oprogramowanie SNAP (Sentinel Application Platform) służące do przetwarzania, analizy i wizualizacji danych satelitarnych, szczególnie z misji programu Copernicus, takich jak Sentinel-1, Sentinel-2 i Sentinel-3. SNAP udostępnia środowisko programistyczne (IDE) oraz API w języku Java. *Esa-snappy* to moduł Python, który umożliwia korzystanie z funkcji oprogramowania ESA SNAP bezpośrednio z poziomu skryptów tego języka dzięki dedykowanemu interfejsowi (ang. *Java-Python bridge*) opartemu o moduł *jpy*. Procedura tworzenia obrazu RGB dla sceny satelitarnej wymaga wykonania następujących operacji:

- pobranie referencji do interfejsu Java poprzez pole *snappy.jpy*
- import klas Java środowiska SNAP poprzez *snappy.jpy.get_type()*
- odczyt obrazu poprzez klasę *ProductIO*
- pobranie pozycji piksela dla zadanej pozycji geograficznej poprzez metody: *getSceneGeoCoding()* i *getPixelPos()*
- pobranie pasma metodą *getBand()*
- pobranie wartości piksela dla zadanej pozycji metodą *readPixels()*
- wytworzenie obrazu na podstawie pasm B4, B3, B2 przy użyciu implementacji JAI SNAP (and Java Advanced Imaging) poprzez *ImageManager.createColoredBandImage()*
- wycięcie obrazu poprzez *Image.crop()*

Przykłady obrazów wygenerowanych powyższą metodą przedstawiają Rysunki 11 i 12.



Rysunek 11: Obraz w paśmie widzialnym dla miasta Tczew, produkt źródłowy: S2A_MSIL2A_20230530T100031_N0509_R122_T34UCF_20230530T134700.SAFE



Rysunek 12: Obraz w paśmie widzialnym dla miasta Rumia, produkt źródłowy: S2A_MSIL2A_20230927T100031_N0509_R122_T34UCF_20230927T141059.SAFE

5. **Uruchomienie repozytorium;** Dostarczone w projekcie repozytorium danych satelitarnych zawiera 471 produktów satelitarnych z misji Sentinel-2 wykonanych pomiędzy kwietniem i październikiem 2023 r. i przedstawiających Trójmiasto i okolicę, gdzie pokrycie chmurami jest mniejsze niż 50%. Produkty zostały wzbogacone o obrazy w formacie PNG wykonane na podstawie warstw w paśmie widzialnym (B4, B3, B2):
 - plik **rgb.png**: cała scena na jednym obrazie,
 - pliki **tczew.png** i/lub **rumia.png**, jeśli na produkcie źródłowym dla danego obszaru możliwe było wycięcie zdjęcia o wielkości 1000x1000 pikseli dla obszaru miast Tczew i/lub Rumia.

Całkowity rozmiar danych to 428GB. Dane te są przechowywane są w chmurze obliczeniowej TASKcloud (dostarczanej przez CI TASK) i udostępniane za pomocą interfejsu S3 pod adresem:

- bucket: `s3://rural-iot/`
- endpoint: <https://s3.storage.task.gda.pl>

Dane są udostępnione publicznie tj. ich pobranie nie wymaga logowania. Poniżej zamieszczono przykład skryptu⁵, który wypisze listę dostępnych w repozytorium produktów. Skrypt zakłada uruchomienie w środowisku Linux, z konsolą BASH i zainstalowanym poleceniem *docker*:

```
#!/usr/bin/env bash
function docker_aws() {
    COMMAND=${1}
    docker run \
        --name "docker_awscli" \
        --rm \
        "public.ecr.aws/aws-cli/aws-cli" \
        --endpoint-url="https://s3.storage.task.gda.pl" \
        ${COMMAND}
}
# pokaż listę produktów
docker_aws "--no-sign-request s3 ls s3://rural-iot/sentinel2/"
```

4.3. Analiza danych pomiarowych

Dane in-situ z sensorów umieszczonych w glebie są rejestrowane z założonym stałym krokiem czasowym i zapisywane w pamięci czujnika inteligentnego. Tworzone w ten sposób ciągi wartości uporządkowanych w czasie stanowią szeregi czasowe próbek sygnałów pomiarowych gleby. Przed przestaniem tych danych na pokład BSP w celu dalszego wykorzystania w chmurze obliczeniowej niezbędne jest opracowanie i doskonalenie oprogramowania czujnika zdolnego do optymalizacji jakościowej i objętościowej danych surowych poprzez uzupełnienie próbek brakujących, korektę próbek niepoprawnych oraz usunięcie próbek nadmiarowych. Zadanie to jest realizowane za pomocą zestawu narzędzi udostępnionych warstwie analizy danych stosu technologicznego CLaaS z uwzględnieniem następujących anomalii jakie mogą wystąpić szeregach czasowych sygnałów pH, T, M lub PV [6]:

1. Wartości próbek sygnału zapisane w pamięci czujnika są przesunięte (opóźnione) w czasie względem przebiegu oryginalnego sygnału (ang. *power gaps*). Sytuacja tego typu wystąpi, gdy w trakcie odczytu kolejnych próbek przez MCU z sensorów umieszczonych w glebie nastąpi zatrzymanie licznika czasu systemowego spowodowane całkowitym brakiem zasilania, np. po rozładowaniu akumulatora zasilającego w nocy.
2. Wartość pojedynczej próbki sygnału wykracza poza dozwolony zakres jego zmienności (ang. *absolute error*). Sytuacja tego typu wystąpi, gdy w trakcie odczytu zaszło krótkotrwałe zdarzenie w środowisku fizycznym sensora umieszonego w glebie powodującego błędny odczyt danej wartości.
3. Wartość kilku sąsiednich próbek sygnału pozostaje w poprawnym zakresie zmienności sygnału, ale znacząco przekracza wartość średnią próbek zarówno w ich lewym jak i prawym sąsiedztwie

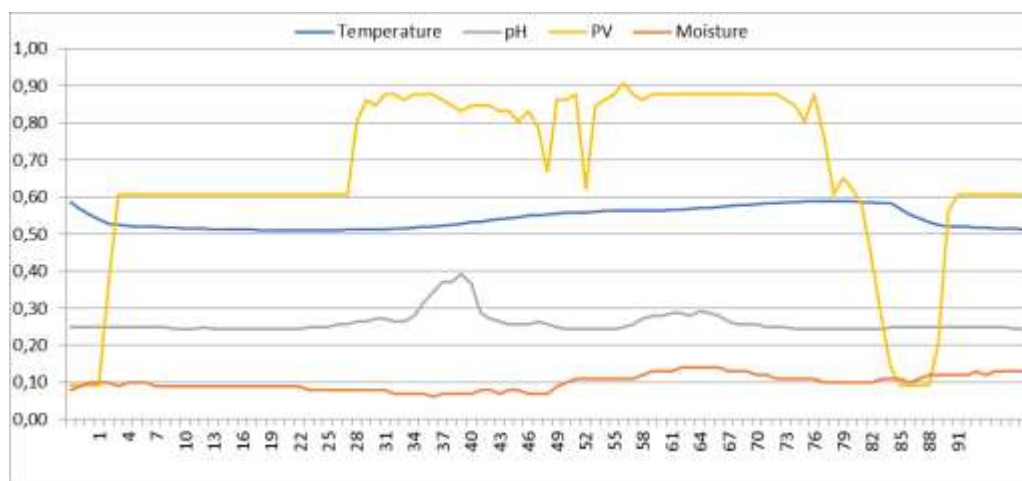
⁵ ze względu na planowaną na trzeci kwartał 2025 r. migrację usług TASKcloud, wystąpi zmiana adres dostępowego z <https://s3.storage.task.gda.pl> na <https://public.s3.storage.task.gda.pl>

- (ang. *peak*). Sytuacja tego typu wystąpi, gdy w trakcie odczytu zaszło krótkotrwałe zdarzenie w środowisku fizycznym sensora umieszczonego w glebie zakłócające odczytywany sygnał.
4. Wartość kilku sąsiednich próbek sygnału pozostaje w poprawnym zakresie zmienności sygnału, ale znacząco przekracza wartość średnią próbek tylko w ich lewym albo prawym sąsiedztwie (ang. *jump*). Sytuacja tego typu wystąpi, gdy w trakcie odczytu danych doszło do trwałej zmiany stanu sensora umieszczonego w glebie, poprzez np. załączenie lub odłączenie jego zasilania (sensor analogowy) czy wyzerowanie jego rejestru odczytu (sensor cyfrowy).
 5. Wartość ciągu próbek sygnału pozostających w poprawnym zakresie zmienności sygnału łagodnie odbija od jego oczekiwanego trendu w dłuższym przedziale (np. dobowym) czasu obserwacji. Sytuacja tego typu wystąpi, gdy w trakcie odczytu kolejnych próbek zaszło zdarzenie powodujące długotrwałe zakłócenie pracy sensora, np. zalanie sensora wilgotności gleby.
 6. Wartości pewnego ciągu próbek oscylują z niewielkim odchyleniem względem siebie wzdłuż linii trendu sygnału (ang. *instabilities*). Sytuacja tego typu wystąpi, gdy w trakcie odczytu kolejnych próbek zasilanie sensora umieszczonego w glebie będzie niestabilne, np. gdy przy niewystarczającym naładowaniu akumulatora zasilającego ogniwo PV zostanie obciążone zbyt mocno przez obliczenia MCU.

Powyższa klasyfikacja anomalii jest wynikiem analizy ponad 175 000 próbek sygnałów pH, M, T i PV zebranych z 7 czujników rozmieszczonych na obszarze o promieniu ok. 35 km w okresie jednego pełnego sezonu wegetacyjnego od kwietnia do października 2023 r. [11]. Analiza ta została przeprowadzona z wykorzystaniem trzech alternatywnych podejść, odpowiednio: dekompozycji statystycznej każdego sygnału na trend, wahania sezonowe i szum, interpretacji wizualnej jego obwiedni oraz interpretacji fizyki jego procesu pomiarowego (por. Tabela 1). Na tej podstawie zostały opracowane narzędzia do uzupełniania brakujących próbek (ang. *power gap correction*), detekcji anomalii (ang. *anomaly detection*) i czyszczenia danych (ang. *data cleaning*) oraz optymalizacji rozmiaru pakietu danych do transmisji z czujnika na pokład BSP (ang. *payload optimization*).

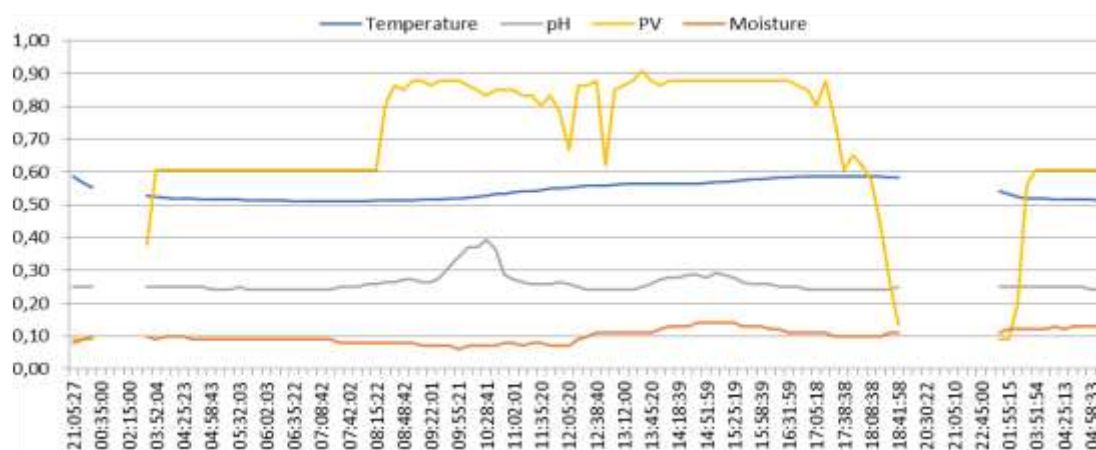
4.3.1. Uzupełnianie brakujących próbek

Konstrukcja czujnika pomiarowego (por. Rysunek 2) nie przewiduje zegara czasu rzeczywistego zasilanego z odrębnej baterii, który by odliczał czas także podczas wyłączenia jednostki MCU (np. po rozładowaniu akumulatora przy braku wystarczającej ilości światła słonecznego). Stąd, gdy dojdzie do zatrzymania pomiarów, ciągi próbek poszczególnych sygnałów mogą być niekompletne. W przypadku odbioru takich danych przez bramkę pokładową BSP – odwiedzającego czujniki nieregularnie i pobierającego dane w porcjach od jedno- do kilkudobowych, stemple czasowe poszczególnych próbek będą tylko kolejnymi numerami próbek w odbieranej serii. Fakt wystąpienia przerw w ich rejestracji przez czujnik nie będzie widoczny dla bramki pokładowej, bowiem nie ma możliwości zsynchronizowania *post factum* jej zegara systemowego z licznikiem próbek w czujniku. Sytuację tę ilustruje Rysunek 13.



Rysunek 13: Niekompletne ciągi próbek sygnałów odebrane przez bramkę pokładową BSP [6]

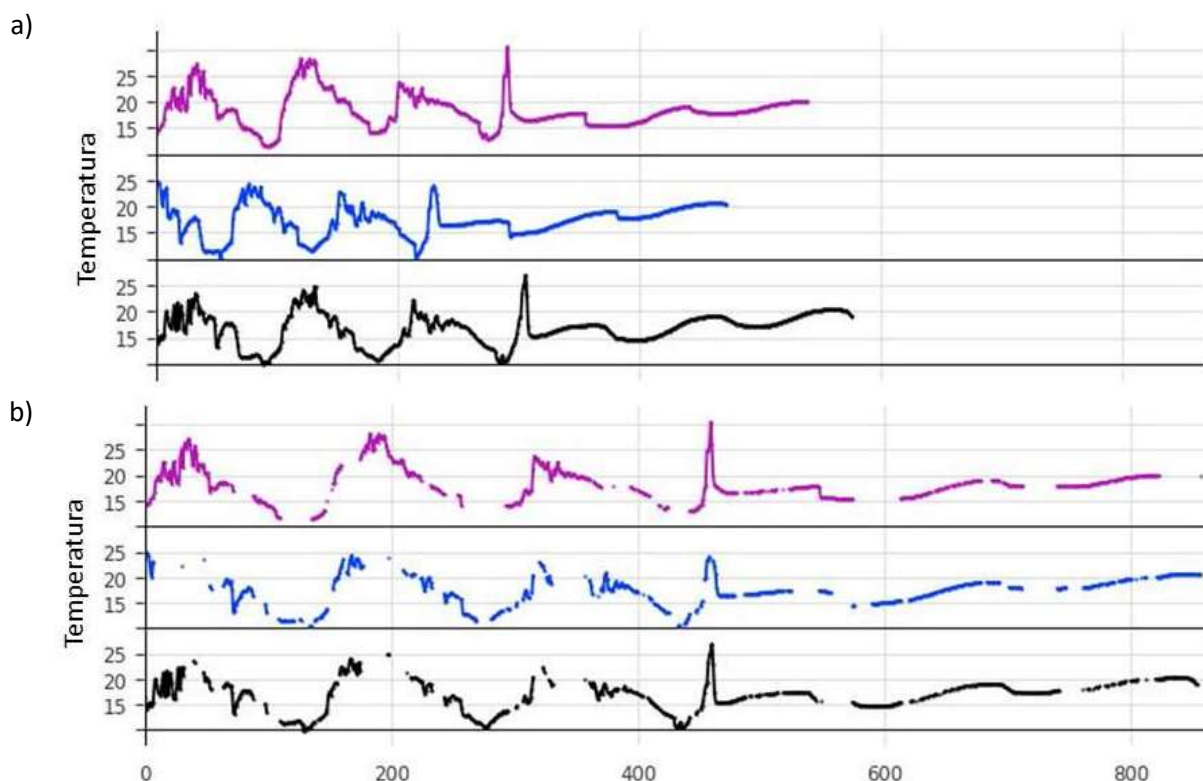
Gdy zamiast bramki pokładowej zostaje użyta naziemna bramka stacjonarna, odbierająca próbki od czujnika na bieżąco i posiadająca własny zegar systemowy, wszystkie przerwy w jego działaniu będą widoczne. Wynika to z faktu, iż wszystkie kolejno odbierane próbki mogą być stemplowane wartościami zegara bramki stacjonarnej. Sytuację tę ilustruje Rysunek 14.



Rysunek 14: Niekompletne ciągi próbek sygnałów odebrane przez naziemną bramkę stacjonarną [6]

Wprawdzie czujnik może samodzielnie wykryć, że ciąg zarejestrowanych próbek jest niekompletny poprzez zliczanie próbek między dwoma kolejnymi przednimi zboczami przedniego sygnału PV (tj. między dwoma kolejnymi wschodami Słońca), to jednak samodzielne ustalenie przez niego miejsc w których brakuje pomiarów jest praktycznie niemożliwe. Stąd próbki w niekompletnych fragmentach dobowych sygnału, przed ich wysłaniem na pokład BSP zostają tylko odpowiednio zaetykietowane jako „niezsynchronizowane” (ang. *time shifted*). Po ich dostarczeniu do chmury obliczeniowej przeprowadzana jest ich właściwa analiza i korekcja za pomocą narzędzia *power gap correction*, wykorzystującego fuzję danych z wielu czujników [7]. Odbывается ona przy założeniu, że chociaż podobne sygnały z poszczególnych czujników rozmieszczonych w pewnej odległości od siebie (sąsiadujących) mogą być niekompletne z powodu przerw w zasilaniu, to jest bardzo mało prawdopodobne by wystąpiły one we wszystkich czujnikach równocześnie. Stąd, jeśli tylko istnieje korelacja między poszczególnymi fragmentami próbek sygnału z różnych czujników, można ją

wykorzystać do detekcji przerw w zasilaniu poprzez przesuwanie i dopasowanie tych fragmentów. Przykład tego działania przedstawia Rysunek 15.



Rysunek 15: Próbkę sygnału temperatury z trzech czujników: (a) z bramki pokładowej, (b) po detekcji przerw [7]

Fuzja danych pomiarowych pobranych z czujników sąsiadujących wykorzystuje specjalnie opracowaną heurystyczną metodę wykrywania przerw w ich zasilaniu [7]. Polega ona na stopniowym dopasowywaniu kolejnych przebiegów do siebie tak długo, aż wszystkie zostaną zsynchronizowane. Kolejność dopasowywania sygnałów zależy od przyjętej metryki odległości między nimi. Jako metrykę wykorzystano DTW (ang. *dynamic time warping*) oraz DDTW (ang. *derivative dynamic time warping*) [13]; ich wspólną zaletą jest możliwość porównywania sygnałów o różnych długościach. W ramach opracowanej funkcjonalności istnieją dwie opcje ustalania kolejności dopasowywania sygnałów. Pierwsza z nich wykorzystuje aglomeracyjny algorytm grupowania hierarchicznego przyjmując za odległość między grupami przebiegów odległość między dwoma najbliższymi sygnałami (ang. *single-linkage*). Przebiegi są dopasowywane w kolejności zgodnej z dendrogramem otrzymanym w wyniku działania algorytmu grupowania. Druga opcja polega na wyborze w każdym kroku sygnału najbliższego grupie sygnałów już zsynchronizowanych i dopasowaniu go do grupy. W obu przypadkach działanie rozpoczyna się od wyznaczenia macierzy odległości między parami sygnałów, a następnie zsynchronizowania dwóch najbliższych sobie. Synchronizacja dwóch przebiegów odbywa się na podstawie ścieżki wyznaczonej przez metodę DTW/DDTW. Jeżeli próbką jednego przebiegu jest dopasowano do dłuższego fragmentu w przebiegu drugim, to metoda zakłada, że w tym miejscu w przebiegu pierwszym jest nieciągłość. Synchronizacja sygnałów z grupą wcześniej zsynchronizowanych lub dwóch grup sygnałów jest bardziej złożona i polega na dopasowaniu dwóch najbliższych przebiegów pochodzących z dwóch grup. Priorytetem jest jak najwierniejsze ich dopasowanie przy jednoczesnym zachowaniu dopasowania dokonanego we wcześniejszych krokach.

4.3.2. Detekcja anomalii

Komponent detekcji anomalii warstwy analityki danych stosu technologicznego Rural IoT, oprócz narzędzi do klasycznej statystycznej analizy szeregów czasowych danych pH, T, M i PV w celu ich dekompozycji na składniki w postaci trendu, wahań sezonowych i szumu, dostarcza usługi umożliwiające profilowanie danych, tj. analizę połączoną ze zrozumieniem fizyki związanych z nimi procesów zachodzących w glebie (por. Tabela 2). W działaniach tych bardzo przydatnym okazuje się narzędzie Grafana [29], umożliwiające bezpośrednią wizualną interpretację obwiedni analizowanego sygnału pomagającą uchwycić wszelkie dające się racjonalnie wytłumaczyć wzorce ich przebiegu.

Tabela 2: Interpretacja fizyczna mierzonych sygnałów gleby

Sygnał	Charakterystyka przebiegu			Interpretacja fizyczna
	Trend	Wahania sezonowe	Szum	
pH	Brak	Trudne do uchwycenia	Łagodne fluktuacje, sporadyczne skoki	Minerały zawarte w glebie mogą rozpuszczać się w wodzie w niej zawartej tylko do pewnego stężenia granicznego (roztwór nasycony)
T	Dzienny (silny)	Dzienne (silne)	Słabe fluktuacje	Ciepło słoneczne jest stopniowo gromadzone w glebie od wschodu Słońca i emitowane po jego zachodzie.
M	Trudne do uchwycenia	Trudne do uchwycenia	Słabe fluktuacje, sporadyczne skoki	Silne opady deszczu w połączeniu z warunkami glebowymi/terenowymi i lokalizacją czujnika mogą spowodować zalanie sondy pomiarowej
PV	Brak	Dzienne (silne)	Silne fluktuacje, sporadyczne skoki	Ogniwo fotowoltaiczne wytwarza maksymalne napięcie znamionowe, które natychmiast spada po dodaniu obciążenia zewnętrznego (otwarte źródło napięciowe)

Opisana wyżej kompleksowa analiza danych pomiarowych nie tylko pozwala sklasyfikować wspomniane wcześniej anomalie, wytłumaczalne na gruncie fizyki związanych z nimi procesów zachodzących w glebie, ale także scharakteryzować każdą z nich formalnie za pomocą zbioru mierzalnych parametrów:

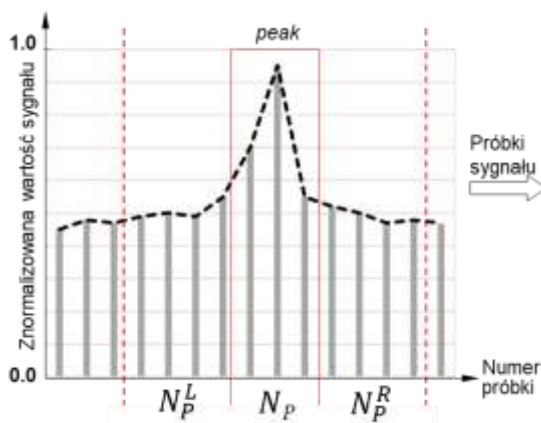
1. **Absolute error;** wartość pojedynczej próbki sygnału wykracza poza dozwolony zakres jego zmienności; typowe (prawidłowe) zakresy dla mierzonych sygnałów pH, T, M i PV na obszarze Polski przedstawia Tabela 3.

Tabela 3: Prawidłowe zakresy zmienności mierzonych sygnałów gleby

Sygnał	Jedn. miary	Zakres typowy	Komentarz
pH	–	[3,0 – 9,0]	Niska wartość pH jest charakterystyczna dla gleby leśnej, średnie wartości dla upraw zbóż, wyższe wartości osiągają gleby upraw warzywnych i sadowniczych.
T	°C	[0 – 40]	Temperatura gleby na głębokości 0,5-1,0 m w okresie wegetacyjnym (marzec-październik) nie osiąga wartości poniżej zera nawet w przy okresowych przymrozkach. Z kolei 40°C to najwyższa średniodobowa temperatura zarejestrowana w Polsce od 1928 roku.
M	%	[10 – 80]	Wilgotność gleby uprawnej w polskiej strefie klimatycznej waha się zazwyczaj w przedziale 20-70%, szczególnie w warstwie przypowierzchniowej (0-7 cm). Przyjęcie nieco szerszego zakresu tego przedziału pozwala wychwycić sytuacje związane z przesychaniem i podmakaniem gleby.

PV	V	[0,0 – 6,6]	Wartość napięcia znamionowego ogniwa PV jest określana przez producenta.
----	---	-------------	--

2. **Peak;** Rysunek 16 przedstawia nagły i krótkotrwały wzrost wartości próbek mierzonego sygnału.



Rysunek 16: Nagły i krótkotrwały wzrost wartości sygnału [10]

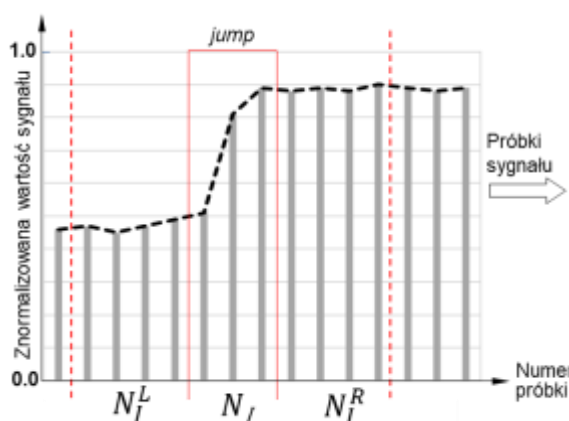
Anomalia tego typu charakteryzuje się pewnym niewielkim fragmentem N_P próbek w poprawnym zakresie zmienności sygnału, ale o wartościach znacząco przekraczających wartość średnią próbek w ich lewym N_P^L i prawym N_P^R sąsiedztwie. Formalnie, tą właściwość fragmentu sygnału $N_P^L \cup N_P \cup N_P^R$ opisujemy za pomocą dwóch parametrów [8], *względnej wysokości* anomalii P (peak):

$$\alpha_P = \frac{\max N_P}{\text{avg } N_P^L \cup N_P \cup N_P^R} \quad (1)$$

oraz *wzłędnej poziomu podstawy* anomalii P (peak):

$$\beta_P = \frac{|\text{avg } N_P^L - \text{avg } N_P^R|}{\text{avg } N_P^L \cup N_P \cup N_P^R} \quad (2)$$

3. **Jump;** nagła i długotrwała zmiana wartości próbek w dłuższym fragmencie sygnału przedstawia Rysunek 17.



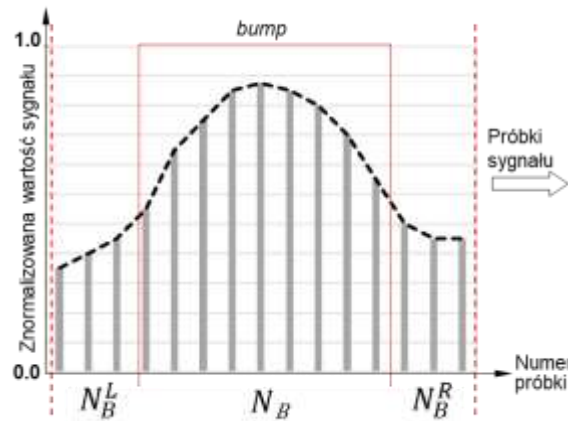
Rysunek 17: Nagła i długotrwała zmiana wartości próbek sygnału [10]

4. Anomalia tego typu charakteryzuje się pewnym niewielkim fragmentem N_J próbek w poprawnym zakresie zmienności sygnału, oddzielającym od siebie dwa fragmenty sygnału o

znacznie różniących się wartościach średnich ich próbek w odpowiednio lewym N_J^L i prawym N_J^R sąsiedztwie. Formalnie, tą właściwość fragmentu sygnału $N_J^L \cup N_J \cup N_J^R$ opisujemy za pomocą parametru *względnej wysokości anomalii J (jump)* [8]:

$$\alpha_J = \frac{\text{avg } N_J}{\min(\text{avg } N_J^L, \text{avg } N_J^R)} \quad (3)$$

5. **Bump**; łagodne odbicie sygnału od jego trendu przedstawia Rysunek 18.

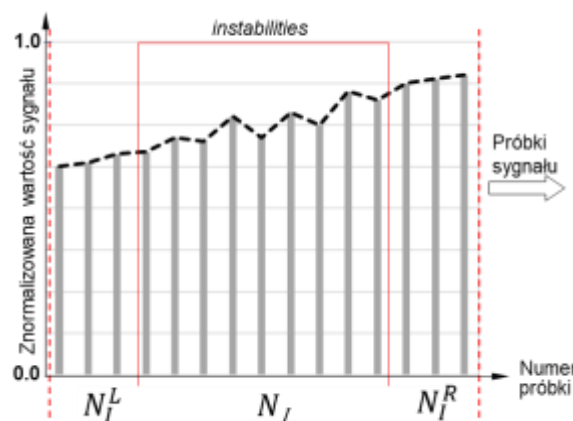


Rysunek 18: Łagodne okresowe odbicie sygnału od przewidywanego trendu [10]

Anomalia tego typu zawiera pewien dłuższy fragment N_B próbek, których wartości stopniowo odchylają się i powracają do linii trendu przewidywanego na podstawie zmian wartości w jego lewym N_B^L i prawym N_B^R sąsiedztwie. Formalnie, tą właściwość fragmentu sygnału $N_B^L \cup N_B \cup N_B^R$ opisujemy za pomocą parametru *względnej wysokości anomalii B (bump)* [8]:

$$\alpha_B = \frac{|\text{avg } N_B^L - \text{avg } N_B^R|}{\text{avg } N_B^L \cup N_B \cup N_B^R} \quad (4)$$

6. **Instabilities**; Rysunek 19 przedstawia oscylację ciągu próbek względem linii trendu sygnału.



Rysunek 19: Oscylacja ciągu próbek względem linii trendu sygnału [10]

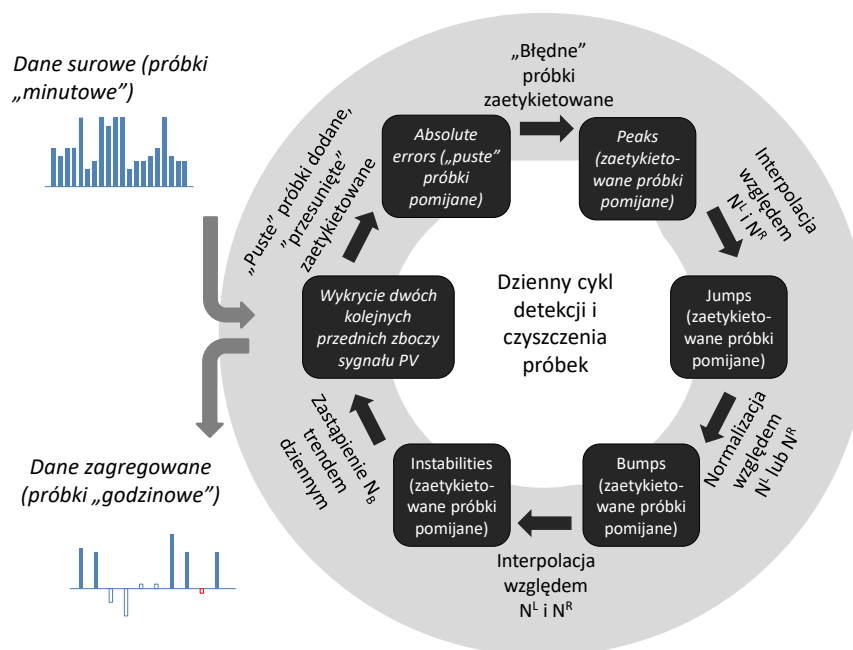
Anomalia tego typu zawiera pewien wyraźny fragment N_I próbek sygnału, których wartości oscylują wzdłuż linii trendu większego fragmentu sygnału $N_I^L \cup N_I \cup N_I^R$. Formalnie opisujemy tą właściwość za pomocą parametru *względnego zniekształcenia obwiedni sygnału w obrębie anomalii I (instabilities)* [7]:

$$\alpha_B = \frac{|\text{avg}(\max N_I^L - \max N_I^R)|}{\text{avg}(\max N_I^L \cup N_I \cup N_I^R - \min N_I^L \cup N_I \cup N_I^R)} \quad (5)$$

Szczegółowe omówienie alternatywnych sposobów detekcji wspomnianych anomalii oraz ocena ich skuteczności znajduje się w Rozdziale 5.

4.3.3. Czyszczenie anomalii

Czujnik pomiarowy, którego schemat blokowy przedstawia Rysunek 2, jest ograniczonym urządzeniem końcowym (ang. *constrained end device*). Ma ono pewien maksymalny dostępny dzienny zapas energii elektrycznej potrzebny do odczytu danych z podłączonych do niego sond pomiarowych i przeliczenia ich na potrzeby transmisji. Ponadto musi ono minimalizować liczbę ramek do wysłania, by optymalnie wykorzystać dostępne okienko czasowe Δt (por. Rysunek 1). Z tego względu czyszczenie danych realizowane przez oprogramowanie czujnika powinno z jednej strony zapewniać minimalizację rozmiaru pakietów danych przesyłanych na pokład BSP, a jednocześnie zachować w przesyłanych danych wszystkie niezbędne informacje potrzebne do fuzji w instancji chmury z danymi z innych czujników. Czujnik musi zatem sam być w stanie zdecydować, czy podejrzane fragmenty sygnału powinny zostać skorygowane, czy tylko odpowiednio zaetykietowane przed wysłaniem. Lokalne naprawy danych powinny być przeprowadzane tak by nie zakłócały fuzji z danymi z innych czujników po dostarczeniu ich do chmury. Innymi słowy, dzienne porcje próbek powinny być agregowane w celu wyeliminowania powtarzających się danych, a nieprawidłowe wartości powinny być w miarę możliwości naprawiane lub tylko oznaczane na podstawie analizy ich lokalnego kontekstu opisanego wzorami (1)-(5). Kolejność i zakres związanych z tym operacji ma znaczenie, jak przedstawia to Rysunek 20.



Rysunek 20: Czyszczenie szeregów czasowych mierzonych sygnałów gleby [6]

Celem operacji przedstawionych na Rysunku 20 jest zagregowanie surowych "minutowych" próbek sygnałów, mierzonych w relatywnie krótkich odstępach czasu (np. $\tau_m=10\text{min}$), do docelowej postaci próbek "godzinowych", reprezentujących dłuższe okresy czasu (np. $\tau_h=60\text{min}$), nie obciążonych

błędami wynikającymi z występowania anomalii w próbkach "minutowych". Wartości τ_m dobierane są z punktu widzenia dostępnych zasobów pamięciowych czujnika, natomiast τ_h z punktu widzenia wielkości dostępnego okienka czasowego Δt (por. Rysunek 1), ograniczającego z góry liczbę bajtów (ang. *data payload*) w ramce wysyłanej każdorazowo na pokład BSP. Ze względu na różną dynamikę mierzonych sygnałów gleby, wartości τ_h powinny być ustalane indywidualnie dla każdego sygnału; metoda ustalania tych wartości została opisana dalej w p.4.3.4.

Operacje cyklu dziennego detekcji i czyszczenia próbek obejmują następujące kroki:

1. Próbkę opóźnioną w wyniku zatrzymania licznika czasu systemowego (ang. *power gap*) mogą mieć prawidłowe wartości, zatem powinny zostać zachowane, ale oznaczone jako „przesunięte”. Najprościej można to zrealizować poprzez odwrócenie bitu znaku wartości każdej takiej próbki. Wykrycie przesuniętych próbek wymaga wyliczenia dobowej liczby N^d zarejestrowanych próbek sygnału pH, T lub M znajdujących się pomiędzy dwoma kolejnymi przednimi zboczami sygnału PV. Liczba ta powinna spełniać warunek [6]:

$$\left| \frac{N^d - N_{avg}^d}{N_{avg}^d} \right| < th_{\Delta N^d} \quad (6)$$

gdzie $th_{\Delta N^d}$ oznacza wartość progową odróżniania próbek przesuniętych od prawidłowych. Na przykład, dla $\tau_m=10\text{min}$ i sporadycznych 6-15min wahań tempa zliczania próbek (ang. *timer drift*) wartość $N^d \approx 133 \pm 25$ i $th_{\Delta N^d}=19\%$. W przypadku wykrycia przesuniętych próbek, brakujące próbki powinny zostać uzupełnione próbkami bez wartości, oznaczonymi jako „puste”.

2. Po wykryciu przesuniętych i brakujących próbek w dobowej porcji danych wszystkich sygnałów czujnik inteligentny kontynuuje wykrywanie anomalii osobno, już tylko w sygnałach pH, T i M. Jest tak dlatego, że jak wskazano w Tabeli 2, zmienność sygnału PV jest związana jedynie z faktem ładowania akumulatora zasilającego czujnik. Sygnał PV nie wykazuje żadnych anomalii wartych analizy i korygowania – z wyjątkiem prawidłowego wykrycia przerw w zasilaniu wpływających na trzy pozostałe sygnały. Zatem następnym krokiem na Rysunku 20 jest wykrycie próbek, których wartości wykraczają poza dopuszczalne zakresy określone w Tabeli 3. Próbkę takie nie mogą być brane pod uwagę przy agregowaniu próbek „minutowych” w próbki „godzinowe” i muszą być oznaczone jako próbki „błędne”. Analogicznie, wstawione w poprzednim kroku próbki „puste” są także pomijane podczas tej agregacji. Zauważmy, że oznaczenie próbek jako „błędne” nie ma wpływu na próbki „przesunięte” z prawidłowymi wartościami. Podczas późniejszej fuzji danych z wielu czujników (por. Rysunek 15) w chmurze próbki „przesunięte i błędne” mogą zostać ostatecznie odpowiednio oznaczone znacznikiem czasu i uzyskać skorygowaną (poprawną) wartość.
3. W kolejnym kroku powinny być wykrywanie nagłe zmiany przebiegu sygnału o krótkotrwałym (peak) lub długotrwałym (jump) charakterze. Kolejność ta wynika z faktu, że zgodnie z interpretacją fizyczną sygnałów pomiarowych gleby, przedstawioną w Tabeli 2, ich zmiany powinny być raczej łagodne i płynne. Z tego względu wykrycie nagłych zmian w ich przebiegu oznacza wystąpienie anomalii w samym procesie pomiarowym, a nie w procesie fizycznym zachodzącym w glebie. Wszystkie zatem próbki w obrębie nagłych zmian (por. Rysunki 16 i 17), które wcześniej nie zostały oznaczone jako „puste”, „przesunięte” lub „błędne” powinny mieć wartości skorygowane względem swoich sąsiadów, np. w drodze interpolacji.
4. Dopiero po skorygowaniu wartości próbek związanych z nagłymi zmianami obwiedni sygnału możliwa jest analiza łagodnych (bump) zmian w jego przebiegu oraz innych, nieznacznie zaburzonych przebiegów względem trendu (instabilities). Anomalie tego typu charakteryzują się występowaniem lokalnych maksimów w stosunkowo dłuższej, dobowej sekwencji próbek.

Korekta łagodnych zmian (bump) polega na wygładzeniu przebiegu sygnału w ich obszarze względem wartości jego sąsiednich fragmentów. Z kolei korekta nieznacznie zaburzonego fragmentu sygnału (instabilities) polega na zastąpieniu go próbkami trendu sygnału obliczonymi jako dobowe średnie ruchome. W obu przypadkach próbki oznaczone wcześniej jako „puste”, „przesunięte” lub „błędne” nie są brane pod uwagę.

Agregowanie próbek „minutowych” (okres próbkowania τ_m) na „godzinowe” (okres próbkowania $T_h > T_m$) jest proste. Dla każdego oczyszczonego sygnału w całodobowej serii N^d próbek każdego sygnału pH, T i M należy wyznaczyć fragmenty po

$$N_h = N_{avg}^d \div \left\lfloor \frac{\tau_m \cdot N_{avg}^d}{\tau_h} \right\rfloor \quad (7)$$

próbek do zagregowania do postaci jednej próbki „godzinowej”; możliwe są następujące przypadki:

- Dany fragment N_h zawiera zarówno prawidłowe (niezaetykietowane) próbki, jak i próbki oznaczone jako „błędne”. Jeżeli przynajmniej połowa z nich jest poprawna, zagregowana próbka godzinowa jest wyznaczana jako ich średnia lub mediana; w przeciwnym razie jest oznaczana jako „błędna”.
- Dany fragment N_h zawiera próbki oznaczone jako „przesunięte”, z których część może być dodatkowo oznakowana jak „błędne”. Jeżeli przynajmniej połowa próbek w tym fragmencie jest oznaczona tylko jako „przesunięte” zagregowana próbka godzinowa jest wyznaczana jako ich średnia lub mediana; w przeciwnym razie jest oznaczana jako „przesunięta i błędna”. Należy pamiętać, że wartości bezwzględne „przesuniętych” próbek są na tym etapie analizy uważane za prawidłowe i będą później potrzebne do fuzji danych z wielu czujników w chmurze.
- Jeśli dany fragment N_h zawiera próbki oznaczone jako „puste” zagregowana próbka godzinowa jest również oznaczana jako „pusta”; w przeciwnym razie zagregowana próbka godzinowa jest wyznaczana jako średnia lub mediana pozostałych próbek tego fragmentu, albo oznaczana jako „błędna” – w zależności od tego, czy te próbki zostały wcześniej oznaczone tylko jako próbki „przesunięte” czy „przesunięte i błędne”.

Opisane wyżej podejście "zachowawcze" do agregacji fragmentów próbek "minutowych" na próbki "godzinowe" jest podyktowane z jednej strony dążeniem do minimalizacji obciążenia obliczeniowego czujnika, a z drugiej założeniem, że nawet w przypadku agregacji obciążonej błędem, fuzja danych z wielu czujników pozwoli na jego skorygowanie.

4.3.4. Optymalizacja rozmiaru danych

W systemach zdalnego monitoringu bardzo istotna jest minimalizacja ilości przesyłanych danych, szczególnie tam, gdzie urządzenia działają w trudnych warunkach terenowych i muszą być energooszczędne. Rzadsze próbkowanie i mniejsze paczki danych pozwalają znacząco zmniejszyć zużycie energii, co przekłada się na dłuższą żywotność czujników. Mniejszy wolumen danych to także niższe koszty transmisji i większa niezawodność. Dodatkowo, uproszczone przesyłanie i przechowywanie danych przyspiesza analizę w chmurze, pozwalając na szybsze podejmowanie decyzji i budowanie bardziej efektywnych modeli predykcyjnych. Jednym ze sposobów na zminimalizowanie ilości przesyłanych danych jest odpowiedni dobór częstotliwości próbkowania. Celem jest zminimalizowanie ilości przesyłanych danych przy zachowaniu odpowiedniej jakości sygnału pomiarowego i skuteczności modeli prognozujących. Procedura doboru częstotliwości próbkowania obejmuje następujące kroki:

1. W pierwszym kroku należy ustalić domyślną bazową częstotliwość, jednakową dla każdego rodzaju sygnału. W projekcie Rural IoT dane standardowo były zbierane co $T_m = 10$ min.
2. Każdy sygnał należy przedstawić przyjmując różne okresy próbkowania od 20 do 300 minut zwiększając okres co 10 minut.
3. W kolejnym kroku należy porównać jakość sygnałów próbkowanych z różnymi okresami próbkowania T_m . Aby ocenić, jak mocno spada jakość sygnału przy rzadszym próbkowaniu, sygnały można odtwarzać przy pomocy interpolacji wielomianowej i wyznaczać błąd rekonstrukcji MAPE (średni bezwzględny błąd procentowy), przyjmujący tym większe wartości im rzadziej zbierane są dane; zdefiniowany jest zdefiniowany jako:

$$MAPE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} \frac{|y_i - \hat{y}_i|}{\max(e, |y_i|)}, \quad (8)$$

gdzie n oznacza liczbę próbek (długość ciągu), $\hat{y}_i$ przewidywaną wartość i -tej próbki, y_i jej rzeczywistą wartość oraz e dowolnie małą liczbę dodatnią, pozwalającą uniknąć nieokreślonych wyników, gdy y_i przyjmuje wartość równą zero. Błąd ten może się różnić nawet dla tego samego typu pomiaru, w zależności od lokalizacji czujnika.

4. Na podstawie wyników otrzymanych w poprzednim kroku należy dla każdego rodzaju sygnału ustalić częstotliwość, przy której błąd MAPE nie przekracza ustalonej wartości progowej wynoszącej, np. 1%. Jeżeli wartości błędów dla danych z pewnego czujnika wyraźnie odstają od pozostałych czujników mierzących ten sam rodzaj sygnału, to należy go pominąć podczas ustalania częstotliwości.
5. Warto również sprawdzić, jak rzadsze próbkowanie wpływa na działanie modeli predykcyjnych, które mogą być w dalszej kolejności uruchamiane w chmurze dla różnych scenariuszy użytkowych, wymagających przewidywania przyszłych wartości sygnału, np. z wykorzystaniem metody lasu losowego [14]. Do jego uczenia wykorzystuje się odtworzone za pomocą interpolacji sygnały. Dla mierzonych parametrów gleby typowo prognoza może obejmować 2 dni (czyli np. $2 \cdot N^d \approx 288$ kolejnych wartości). Każdy punkt w serii przewidywany jest krok po kroku, na podstawie poprzedniego. Jako miarę jakości modelu prognozującego można ponownie wybrać błąd MAPE liczony między ciągiem prognozowanym a prawdziwym. Eksperyment należy powtórzyć dla wszystkich testowanych częstotliwości i przeanalizować jaki ma ona wpływ na dokładność modelu predykcyjnego. Eksperymenty przeprowadzone na danych w projekcie nie wykazały jedno-znacznie wpływu częstotliwości próbkowania na skuteczność modelu.
6. Jeśli nie daje się jednoznacznie powiązać skuteczności modelu uczonego w kroku 5 z okresem próbkowania, to jako ostateczne wartości częstotliwości próbkowania można przyjąć otrzymane w kroku 4 dla progu błędu MAPE = 1%.

W wyniku opisanych wyżej działań dla każdego sygnału, w zależności od dynamiki jego zmian, otrzymujemy różne graniczne wartości okresów próbkowania przy zachowaniu odpowiedniej jakości sygnału pomiarowego. Im większy ten okres tym mniejsza będzie minimalna łączna liczba dobowych próbek do przetransmitowania na pokład BSP. Zakładając, że wartości próbek sygnałów wymienionych w Tabeli 3 zapisywane są na dwóch bajtach łączny dobowy rozmiar pakietu danych do wysłania nie przekracza 120 B; szczegółowe wartości wyznaczone eksperymentalnie w projekcie Rural IoT [9] przedstawia Tabela 4.

Tabela 4: Okresy próbkowania i odpowiadające im rozmiary danych

Sygnał	Okres próbkowania [min]	Liczba bajtów na dobę
pH	220	14
T	90	32

M	70	42
PV	90	32

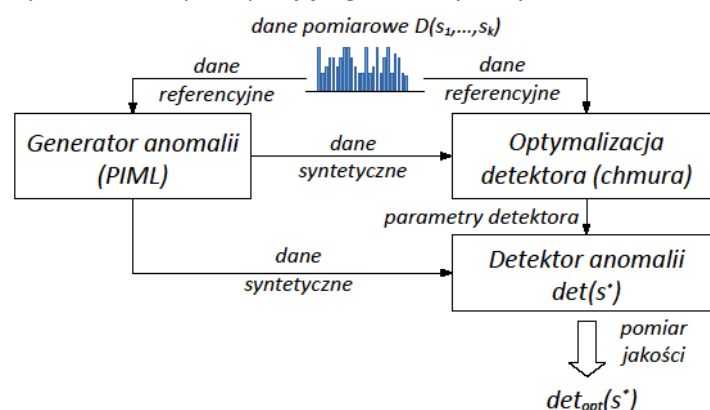
Rozmiar pakietu przesyłanych danych może być mniejszy dla różnych klas scenariuszy, jak opisano dalej w p.6.5.

4.4. Rozwój oprogramowania czujników

Ze względu na ograniczone zasoby obliczeniowe czujnika inteligentnego wytwarzanie i rozwój jego oprogramowania wymaga wsparcia ze strony podmiotu zewnętrznego, takiego jak np. chmura obliczeniowa. Dotyczy to szczególnie sytuacji, gdy zakładane jest stosowanie modeli sztucznej inteligencji w celu ustawicznego doskonalenia detektora anomalii, umożliwiającego coraz lepsze czyszczenie serii danych pomiarowych w miarę stopniowego gromadzenia wiedzy na temat monitorowanych zjawisk w całym ekosystemie pomiarowym. Wiedza ta powinna podlegać okresowej redystrybucji, poczynając od początkowej konfiguracji parametrów detektora wykorzystywanego przez czujniki w procesie czyszczenia surowych danych pomiarowych w oparciu o wcześniej pozyskaną wiedzę nt. monitorowanych zjawisk (np. na podstawie Tabel 2 i 3), a następnie ich regularnej aktualizacji podczas fuzji danych z wielu czujników w kolejnych cyklach. Tak zaktualizowane parametry detektora mogą być zarówno przekazywane zwrótnie do już wcześniej zainstalowanych czujników pomiarowych (gdy oprócz nadawania ich moduł RAT umożliwia także odbiór danych z bramki nomadycznej), jak i wykorzystane przez oprogramowanie kolejnej generacji nowych czujników.

4.4.1. Wersja początkowa detektora anomalii

Wytworzenie początkowej wersji oprogramowania detektora anomalii wymaga znajomości konstrukcji pomiarowego urządzenia końcowego, metod pomiaru wybranych parametrów środowiskowych oraz fizyki monitorowanych procesów. Na tej podstawie możliwe jest formalne scharakteryzowanie poszczególnych anomalii za pomocą zbioru mierzalnych parametrów, np. jak przedstawiono to wcześniej za pomocą równań (1) – (5). Kluczowym zagadnieniem na tym etapie rozwoju oprogramowania detektora jest określenie optymalnych wartości tych parametrów, co z kolei wymaga dysponowania odpowiednio dużym wolumenem surowych danych pomiarowych. Gdy takim zbiorem nie dysponujemy pomocnym może okazać się wykorzystanie schematu *Data Free* przedstawionego na Rysunku 21, wykorzystującego *dane syntetyczne*.



Rysunek 21: Schemat *Data Free* rozwoju oprogramowania detektora [8]

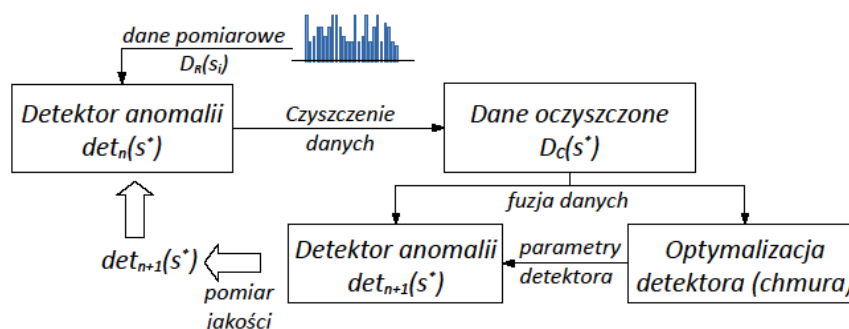
Schemat ten stanowi adaptację jednego z modeli uczenia sfederowanego [20], w którym zaawansowany obliczeniowo moduł „nauczyciela” poszukuje optymalnych wartości parametrów poszczególnych anomalii, a moduł „ucznia” stanowi oprogramowanie właściwego detektora anomalii

przeznaczone do zainstalowania na urządzeniu końcowym. Oba moduły, „nauczyciel” i „uczeń”, zasilane są danymi zawierającymi anomalie, a cały proces kontynuowany jest aż do osiągnięcia pożądanego poziomu jakości detekcji przez „ucznia”. Wynikiem tego działania jest zestaw parametrów detektora $det_{opt}(s^*)$, taki sam dla każdego czujnika $s^* \in \{s_1, \dots, s_k\}$. Moduł „nauczyciela” może wykorzystywać różne modele, opisane szczegółowo dalej w p.5.1

Dane syntetyczne zawierające anomalie, choć generowane automatycznie, powinny w realistyczny sposób odzwierciedlać dane rzeczywiste, zbierane przez czujniki pomiarowe nie wyposażone w detektor anomalii. Można to zrealizować w sposób opisany w [6]. Zbiór danych D zarejestrowanych przez czujniki $\{s_1, \dots, s_k\}$ powinny zostać w pierwszej kolejności przeanalizowane przez eksperta z wykorzystaniem narzędzia do wizualizacji szeregów czasowych, np. grafana opisanego wcześniej, a wszelkie zauważone anomalie usunięte w drodze edycji. W ten sposób wytworzony zostaje pewien zasób danych referencyjnych (idealnych szeregów czasowych), niezawierających anomalii. Następnie do danych referencyjnych wstrzykiwane są pojedyncze anomalie o losowo zmieniających się wartościach wybranych parametrów. Fragmenty próbek sygnału, w których wstrzykiwane są poszczególne anomalie powinny być wybierane realistycznie, np. powinny pozostawać w zakresie cykli dziennych sygnału (pomiędzy dwoma kolejnymi przednimi zboczami sygnału PV), występować sporadycznie w okresach wielodniowych (np. tygodniowych) i się wzajemnie nie nakładać (kumulować).

4.4.2. Wersje rozwojowe detektora anomalii

W trakcie działania czujników ekosystemu w dłuższym okresie czasu wolumen oczyszczonych danych $D_c(s^*)$ pobranych z czujników $s_1, \dots, s_k$ systematycznie rośnie. Można go wówczas wykorzystać do bieżącej oceny jakości aktualnej wersji detektora $det_n(s^*)$ zainstalowanego w czujnikach odczytujących poszczególne dane surowe $D_R(s_i)$, $i=1, \dots, k$. Do tego celu nadaje się schemat uczenia sfederowanego [20] *Life-Long Learning*, pod adaptacji przedstawionej na Rysunku 22.



Rysunek 22: Schemat *Life-Long Learning* rozwoju oprogramowania detektora [8]

Wszystkie dotychczas zebrane w chmurze dane pomiarowe dostarczane są na wejście modułu, który realizuje wybrany model uczenia maszynowego, systematycznie douczany danymi $D_c(s^*)$. W wyniku tego procesu wyznaczany jest zestaw nowych wartości parametrów detektora $det_{n+1}(s^*)$. Jeśli wyniki detekcji anomalii otrzymane dla nowszej wersji są lepsze od wyników otrzymywanych dotychczas, nowy zestaw parametrów $det_{n+1}(s^*)$ jest w miarę możliwości ładowany do każdego istniejącego czujnika $s_1, \dots, s_k$ ekosystemu oraz ustawiany we wszystkich nowych instalowanych w nim czujnikach. Tak w przypadku schematu *Data Free* jak i *Life-Long Learning* pomiar jakości detekcji anomalii odbywa się z wykorzystaniem sygnałów referencyjnych, w sposób opisany dalej.

5. Jakość danych pomiarowych

Jak przedstawiono wcześniej na Rysunku 1, podział zadań pomiędzy komponentami ekosystemu pomiarowego Rural IoT jest następujący:

- Inteligentne czujniki (urządzenia końcowe) odczytują dane z sensorów z krótkim (minutowym) okresem próbkowania, zapisują je w pamięci lokalnej, raz na dobę dokonują ich czyszczenia zgodnie z cyklem przedstawionym na Rysunku 20 oraz agregują w pakiety gotowe do przesłania na pokład BSP; agregacja polega na ustaleniu dla każdego oczyszczonego sygnału odpowiednio dłuższego (godzinowego) okresu próbkowania, zgodnego z wartościami przedstawionymi w Tabeli 4.
- Bramka mobilna odbiera pakiety danych z każdego czujnika, który znajdzie się w jej zasięgu, dokonuje ich zdekodowania oraz zapisuje dedykowanym wolumenie, kopiowanym po wyłączeniu BSP do repozytorium w chmurze.
- Chmura obliczeniowa gromadzi dane ze wszystkich czujników, przeprowadza ich fuzję w celu uzupełniania brakujących próbek (ang. *power gaps*), jak opisano na Rysunku 15, dotrenowuje modele detekcji anomalii wykorzystywane do czyszczenia danych i dokonuje pomiaru jakości oczyszczonych danych; w przypadku uzyskania przez dotrenowane modele wyższej jakości procesu czyszczenia, ulepszone (zoptymalizowane) wartości parametrów anomalii i modeli służą jako podstawa do wygenerowania ulepszonej wersji oprogramowania czujników, jak opisano wcześniej na Rysunkach 21 i 22.

Dla powyższego podziału kluczowym zagadnieniem jest zapewnienie metody czyszczenia danych „użytecznej” z punktu widzenia jej praktycznej realizacji na urządzeniu końcowym, ograniczonym pod względem możliwości obliczeniowych i zapotrzebowania na energię elektryczną.

5.1. Metody detekcji anomalii

W przypadku szeregów czasowych sygnałów pH, T, M i PV analiza statystyczna polegająca na wydzieleniu z nich trendu, wahań sezonowych i szumu nie pozwala jednoznacznie zidentyfikować i zlokalizować powtarzających się, charakterystycznych wzorców w ich przebiegu. Jako opisano wcześniej w Tabeli 2 właściwości fizyczne tych sygnałów skutkują niestacjonarnością średnich i wariancji ich parametrów oraz sezonowością i nieregularnymi wahaniami trendów w przedziałach o długości trudnej do uchwycenia [6]. Praktycznie tylko sygnał temperatury gleby T wykazuje jakiś wyraźny trend w ciągu dnia, powtarzając się regularnie między dwoma kolejnymi wschodami słońca. Z kolei statystyczna dekompozycja sygnału wilgotności gleby M nie wykazuje żadnych istotnych powtarzających się wzorców w całym sezonie wegetacyjnym – ani w trendzie, ani w sezonowości. Wreszcie sygnał kwasowości pH wykazuje praktycznie całkowity brak powyższych zależności.

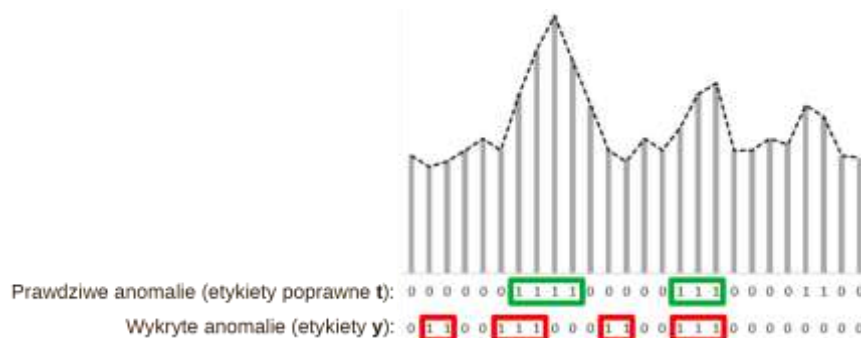
Z tego względu, detektor anomalii w każdym pojedynczym czujniku powinien wykorzystywać możliwie całą wiedzę nt. zjawisk zachodzących w procesach pomiarowych sygnałów pH, T, M i PV w całym ekosystemie Rural IoT. Stąd wydobywanie i agregacja tej wiedzy poprzez fuzję danych z wielu czujników realizowana jest przez komponent chmurowy ekosystemu, jak opisują schematy na Rysunkach 21 i 22, natomiast jej bezpośrednie wykorzystanie następuje w module detektora anomalii każdego czujnika. W ten sposób problem detekcji określonych klas anomalii w szeregach próbek pomiarowych można sprowadzić do zadania optymalizacji wartości parametrów charakteryzujących daną anomalię (podejście heurystyczne) lub do zadania szacowania prawdopodobieństwa przynależności poszczególnych próbek szeregu do anomalnego fragmentu sygnału (podejście neuronowe). Poniżej przedstawiono cztery alternatywne metody budowy i optymalizacji parametrów detektora anomalii, poczynawszy od metody heurystycznej wykorzystującej symulowane wyżarzanie,

poprzez mieszaną metodę heurystyczno-neuronową, aż po dwie opcjonalne metody wykorzystujące autokodery. Wszystkie cztery metody zostały przebadane dla rzeczywistych danych pomiarowych udostępnionych w [11], a ich szczegółową ocenę można znaleźć w [6] i [7].

Implementacja powyższych metod wymaga uprzedniej normalizacji wartości wszystkich analizowanych próbek w celu usunięcia części wariancji sygnału oraz ułatwienia wykrywania anomalii. Brak normalizacji skutkowałby koniecznością wykorzystania znacznie większej ilości danych do uczenia modeli neuronowych. Możliwe są dwa warianty normalizacji próbek sygnału: poprzez obliczanie wartości każdej próbki x w zadanym fragmencie N próbek znormalizowanej względem średniej wartości μ_N wszystkich próbek w tym fragmencie jako $x_N = x/\mu_N$, albo względem wartości średniej μ_N i odchylenia standardowego σ_N wszystkich próbek w tym fragmencie jako $x_N = (x - \mu_N)/\sigma_N$. Długość N fragmentu sygnału dla którego obliczane są parametry μ_N i σ_N ma znaczenie o tyle, że przy anomaliach, których próbki mają względnie stałą amplitudę wartość $N \cong N_{\max}$ (bliską całkowitej długości analizowanej serii próbek, np. z całego sezonu wegetacji) normalizacja wprowadza trudną do wykrycia dodatkową wariancję przekształconego sygnału. Z przeprowadzonych badań [7] wynika, że najlepsze wyniki daje normalizacja względem średniej wartości μ_N , gdzie $N \ll N_{\max}$ odpowiada dużo krótszym fragmentom sygnału, np. dziennym lub tygodniowym.

5.1.1. Heurystyczne

Metodę rozwiązania zadania optymalizacji wartości parametrów charakteryzujących anomalie poszczególnych klas opisanych na Rysunkach 16-19 przedstawia poglądowo Rysunek 23.



Rysunek 23: Prawdziwe i wykryte anomalie w analizowanych sygnale [6]

Jak widać, przy dobranych heurystycznie wartościach parametrów anomalii opisanych wzorami (1)-(5) w p.4.3.2 fragmenty próbek zaetykietowane symbolem „1” przez dekoder jako wykryte anomalie mogą się częściowo lub całkowicie nie pokrywać z fragmentami sygnału, które rzeczywiście stanowią anomalię. Zadanie optymalizacji polega zatem na znalezieniu możliwie najlepszych wartości parametrów poszczególnych anomalii tak, aby wykryte anomalie jak najlepiej odpowiadały anomaliom prawdziwym.

Do tego celu niezbędne jest wykorzystanie dwóch wartości miar: *błędu próbki* (ang. *sample error*) i *błędu sekwencji* (ang. *sequence error*) [6]. Błąd próbki wyliczany jest jako znormalizowana suma próbek z etykietami anomalii prawdziwych t i wykrytych y :

$$E_{smp} = \frac{|t \oplus y|}{(|t| + \min(|y|, |t|))} \quad (6)$$

gdzie $t \oplus y$ oznacza operację XOR wykonywaną na odpowiadających sobie bitach sekwencji t i y , a $|\cdot|$ oznacza sumę wartości bitów ciągu binarnego. Z kolei błąd sekwencji wyliczany jest jako:

Do użytku wewnętrznego zespołu wykonawców projektu

$$E_{sqn} = \frac{\sum_{j=1}^M E_y(p_j(t)) + \sum_{j=1}^K E_t(p_j(y))}{M + \min(K, M)} \quad (7)$$

gdzie M i K oznaczają odpowiednio liczbę „prawdziwych” i „wykrytych” anomalii, $E_y(p_j(t))$ jest miarą pokrycia błędu j -tej prawdziwej anomalii pasującej do $p_j(t)$ etykiet wykrytych anomalii y , a $E_t(p_j(y))$ jest miarą pokrycia j -tej wykrytej anomalii pasującej do $p_j(y)$ etykiet prawdziwych anomalii t . Błąd pokrycia wyliczany jest jako:

$$E_q(p) = \left(\frac{1 - |p \cap q|}{|p|} \right)^2 \quad (8)$$

gdzie $|p \cap q|$ to liczba próbek zaetykietowanych odpowiednio p i q . Funkcja $\min()$ została użyta we wzorach (7) i (8) w celu przeciwdziałania tendencji nadmiernego wzrostu liczby i zakresu fałszywych wykryć anomalii, jeśli zamiast niej użyto by odpowiednio $\sum_i y_i$ i K . W przykładowej sytuacji przedstawionej na Rysunku 23, obliczając wzory (6) i (7), można łatwo zauważyć, że $E_{smp} = 0,50$, a $E_{sqn} = 0,56$. Ze względu na fakt, że podczas eksperymentu oba wzory mogą dawać różne wyniki dla różnych anomalii i czujników, jako funkcję straty dla algorytmu optymalizacji należy przyjąć średni błąd $E = (E_{smp} + E_{sqn})/2$. Optymalizację parametrów detektora anomalii metodą symulowanego wyżarzania (SA) można przeprowadzić z wykorzystaniem stosunkowo niewielkiego zbioru danych pomiarowych, zarejestrowanych w kilku losowo wybranych lokalizacjach przez np. 3 czujniki. Dla wybranych czujników wykorzystywane są po co najmniej dwa szeregi czasowe dla każdego sygnału – odpowiednio pH, T i M. W ten sposób uzyskuje się zestawy danych do trenowania detektora anomalii w różnych warunkach. Celem tego działania jest precyzyjne dostrojenie parametrów detekcji dla czterech typów anomalii opisanych wcześniej, z oddzielnymi procesami optymalizacji dla każdego typu anomalii i sygnału. W rezultacie tworzonych jest np. 12 niezależnych zadań optymalizacyjnych do przeprowadzenia równolegle na chmurze obliczeniowej, z których każde ukierunkowane jest na określoną kombinację typów anomalii–sygnał. Zadania te charakteryzują się stosunkowo niewielką liczbą parametrów podlegających strojeniu, zwykle od 2 do 5 na typ anomalii, dając łącznie 15 elementowe wektory Θ parametrów dla wszystkich 4 klas anomalii. Podejście SA dobrze sprawdza się w takich warunkach, ponieważ pozwala procesowi wyszukiwania uciec od minimów lokalnych poprzez tymczasowe akceptowanie gorszych rozwiązań. To zachowanie jest z kolei regulowane za pomocą meta parametru τ sterującego zarówno prawdopodobieństwem zaakceptowania gorszych rozwiązań, jak i rozmiarem sąsiedztwa przeszukiwań. Prawdopodobieństwo zaakceptowania każdego nowo zaproponowanego rozwiązania określone jest za pomocą wzoru:

$$P_{acc} = \frac{1}{1 + \exp\left(\frac{E}{c\tau}\right)} \quad (9)$$

gdzie $\Delta E = E_{curr} - E_{prev}$ określa różnicę pomiędzy wartościami funkcji celu bieżącego i poprzednich rozwiązań, a c jest współczynnikiem skalowania, który moduluje wrażliwość na zmiany τ . Wartość $\Delta E < 0$ wskazuje na poprawę, a prawdopodobieństwo akceptacji zbliża się do 1. W miarę jak wartość parametru τ maleje z upływem czasu, prawdopodobieństwo zaakceptowania gorszych rozwiązań maleje, czyniąc proces optymalizacji zbieżnym. Szybkość redukcji wartości parametru τ dobierana jest tak, by osiągnąć minimalną wartość τ_{min} w ostatniej epoce optymalizacji. Do tego celu służy współczynnik zaniku (ang. *decay factor*), wyliczany ze wzoru:

$$w_{\tau} = \exp \frac{\log(\tau_{min}/\tau)}{n_e + 1} \quad (10)$$

gdzie n_e oznacza liczbę epok. Oprócz malejącego prawdopodobieństwa akceptacji nowych rozwiązań, ich proponowany zakres również ulega zmniejszaniu proporcjonalnie do τ , co pozwala skutecznie zawęzić zakres przeszukiwań w miarę postępu procesu optymalizacji. Nowe rozwiązania są generowane w pobliżu bieżącego wektora parametrów Θ poprzez dodawanie losowego zaburzenia Δ , odpowiednio skalowanego w celu odzwierciedlenia zarówno bieżącej wartości τ , jak i względnych wielkości parametrów z wektora Θ . Jego aktualizacja wyliczana jest wzorami:

$$\theta' = \theta + \Delta, \quad \Delta = \mathcal{N}(\mu = 0, \sigma^2 = \tau) \odot \theta_h \quad (11)$$

gdzie $\mathcal{N}$ jest 15-wymiarowym rozkładem normalnym, Θ_h jest wektorem parametrów analizowanych anomalii o wartościach dobranych heurystycznie, a $\odot$ oznacza operator mnożenia wektorów element po elemencie.

Szczegółowe wyniki uzyskane po zastosowaniu opisanej wyżej metody optymalizacji parametrów anomalii opisanych w p.4.3.2 dla danych pomiarowych dostępnych w repozytorium IEEEDataPort [11] można znaleźć w publikacji [10].

5.1.2. Neuronowo-heurystyczne (NPH)

Podejście NPH obejmuje dwie fazy. W pierwszej fazie model neuronowy jest uczony wykrywania środkowych i skrajnych próbek danej anomalii, a w drugiej fazie na podstawie uzyskanych w ten sposób prawdopodobieństw wystąpienia każdej próbki w środku i na skraju anomalii, poszczególne anomalie są lokalizowane metodą heurystyczną.

Uczenie modelu NPH; Model neuronowy jest trenowany w oparciu o przykłady zawierające wektory wejściowe w postaci fragmentów sygnału o stałej długości. Przykłady mogą być pozytywne, tj. zawierające anomalię lub negatywne, bez anomalii. Ucznieniu podlegają dwa typy modeli: **ANN-Center** uczony pozytywnymi przykładami z anomalią zlokalizowaną w środku wektora wejściowego oraz **ANN-Edge** uczony pozytywnymi przykładami z lewym skrajem anomalii względem wektora wejściowego. W ten sposób trenowanych jest osiem modeli, po dwa dla każdego z czterech zdefiniowanych na Rysunkach 16-19 typów anomalii i dla każdego parametru fizycznego (T, M, pH), co daje łącznie 24 modele. Każdy model zwraca prawdopodobieństwo wystąpienia środka anomalii lub odpowiednio jej lewego skraju w danym fragmencie sygnału. Prawdopodobieństwo wystąpienia prawego skraju danej anomalii jest określane poprzez zastosowanie sygnału wejściowego z odwróconą kolejnością próbek dla modelu ANN-Edge. Struktura sieci neuronowej jest taka sama dla ANN-Center i ANN-Edge, z wyjątkiem liczby wejść, która różni się nieco dla każdego z tych modeli, w zależności od typu anomalii. Przykładowo [10], może to być 4-warstwowa sieć, której 2 pierwsze warstwy splotowe zawierają odpowiednio 16 filtrów o szerokości 5 i 8 filtrów o szerokości 3; daje to na wyjściu drugiej warstwy łącznie 8 wektorów o długości równej długości wektora wejściowego. Ostatnie 2 warstwy tej sieci to warstwy w pełni połączone, zawierające odpowiednio 20 i 2 neurony. Ostatnia (czwarta) warstwa wykorzystuje funkcję aktywacji *softmax* do obliczenia prawdopodobieństwa wystąpienia anomalii, podczas gdy wszystkie wcześniejsze warstwy wykorzystują funkcję aktywacji *ReLU*. Pewnym problemem jest tutaj dobór długości fragmentu sygnału N , stanowiącego długość wektora wejściowego podawanego do sieci. Dłuższe fragmenty sygnału pozwalają na lepsze uwypuklenie anomalii na tle sygnału poprawnego, ale wiąże się to z dłuższym czasem przetwarzania danych przez sieć neuronową, który jest mniej więcej proporcjonalny do długości wektora wejściowego; ma to

kluczowe znaczenie podczas przetwarzania próbek sygnału przez mikrokontroler (MCU) czujnika inteligentnego. Dłuższe fragmenty sygnału mogą również prowadzić do nadmiernego dopasowania do danych uczących. W związku z tym niezbędny jest pewien kompromis pomiędzy miarą poprawności segmentacji a czasem przetwarzania danych. Na podstawie wyników przedstawionych w [10], długość sygnału N do wykrycia środka anomalii najlepiej wybrać tak, aby była równa trzem średnim szerokościom anomalii, a do wykrycia krawędzi – równą dwóm średnim szerokościom anomalii.

Detekcja anomalii przez NPH; wyuczone wcześniej modele neuronowe wykorzystywane są do wykrywania anomalii we fragmencie sygnału (oknie), o stałej długości podanym na wejście każdego modelu. Prawdopodobieństwo środkowej i skrajnej lewej próbki anomalii uzyskiwane jest bezpośrednio z modeli ANN-Center i ANN-Edge. Prawdopodobieństwo skrajnej prawej próbki danej anomalii uzyskuje się poprzez odwrócenie wektorów wejściowych do odpowiednich modeli ANN-Edge, wyuczonych wcześniej do wykrywania próbek lewego skraju tej anomalii. Fragment sygnału $N_*, * \in \{P, J, B, I\}$, jest uważany za potencjalną anomalię, gdy po obu swoich stronach (lewej N^L i prawej N^R) oraz w środku ma dostatecznie wysokie maksima wartości odpowiednich prawdopodobieństw. Dla uproszczenia przyjmuje się jedną wartość progową th_h dla wszystkich trzech maksimów, po czym każdorazowo rozważane jest dane maksimum. Zakłada się również, że dopuszczalne zakresy szerokości poszczególnych anomalii są równe minimalnej i maksymalnej szerokości anomalii danego typu w danych uczących. Określenie wartości lokalnego maksimum prawdopodobieństwa polega na przyjęciu progu th_r wysokości regionu maksymalnego, który może zawierać tylko jedno maksimum – analogicznie do topograficznej miary wybitności szczytu (MWD) w paśmie górskim, określającej wysokość najniższego konturu, w obrębie którego nie ma wyższego szczytu. Dla wartości maksimum wszystkich kluczowych punktów anomalii w ciągu próbek sygnału, można wyznaczyć potencjalne anomalie potencjalne z trójek maksimów występujących w odpowiedniej kolejności, przy czym szerokość potencjalnej anomalii jest ograniczona do maksymalnej szerokości anomalii w zbiorze uczącym. Zakłada się, że anomalie tego samego typu nie mogą się nakładać w jednym sygnale pomiarowym. Dlatego, aby ocenić istotność potencjalnej anomalii obliczana jest metryka:

$$grade = h_L h_C h_R f_d \quad (12)$$

gdzie h_L , h_C i h_R oznaczają wartości odpowiednich maksimów prawdopodobieństwa, natomiast

$$f_d = \frac{\min(d_{LC}, d_{CR})}{\max(d_{LC}, d_{CR})} \quad (13)$$

jest miarą „centralności” maksymalnego prawdopodobieństwa środka anomalii względem maksimów prawdopodobieństw jej lewego i prawego skraju. Jeśli potencjalne anomalie się nakładają, to te o niższej wartości metryki $grade$ są odrzucane. Po wyznaczeniu wektorów prawdopodobieństwa środka i skrajów anomalii, faza segmentacji heurystycznej jest powtarzana dla różnych wartości progu th_h , w celu wyznaczenia jego optymalnej wartości. Uzyskane w tym zakresie wyniki zostały szczegółowo opisane w [10].

5.1.3. Autokodery „binarne” (AEB)

Szacowanie prawdopodobieństwa przynależności poszczególnych próbek sygnału do fragmentu odpowiadającego anomalii może zostać zrealizowane także za pomocą autokodera. Charakterystyczną cechą sieci tego typu jest taka sama liczba wejść i wyjść, co wymaga użycia wielu jednowymiarowych warstw splotowych typu Conv1D, niepowodujących skrócenia sygnału na wyjściu każdej warstwy. W tym celu należy użyć warstw z parametrami $padding = same$ i $strides = 1$. Alternatywnie można skrócić

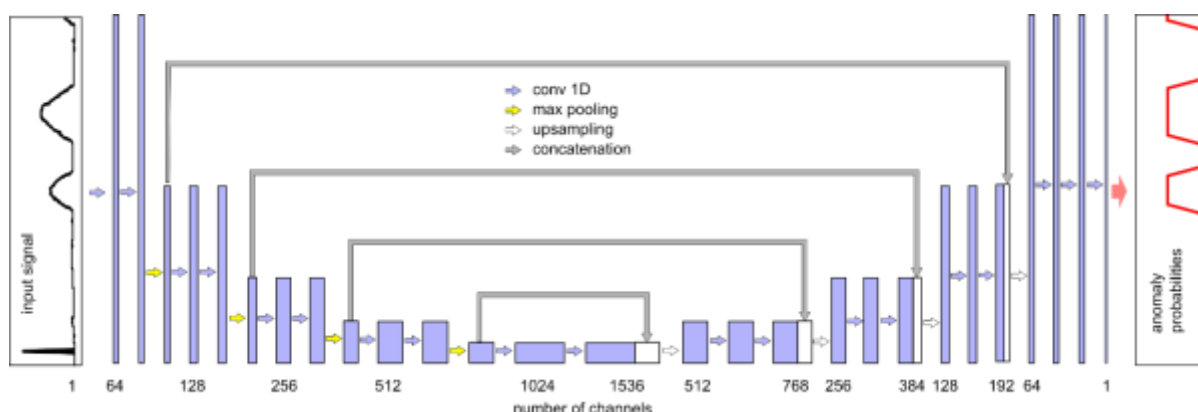
sygnał wyjściowy w początkowej strukturze sieci, np. poprzez użycie *strides* = 2, a następnie wydłużyć sygnał za pomocą transponowanych warstw splotowych Conv1DTranspose w końcowych warstwach sieci. Szczegółowy opis tych wariantów można znaleźć w [10]. Kompresja informacji w centralnej części struktury autokodera sprawia, że sieć tego typu umożliwia całościową analizę całego fragmentu sygnału wejściowego przy użyciu niewielkiej liczby warstw, co może mieć znaczenie w przypadku segmentacji anomalii na ograniczonym obliczeniowo urządzeniu końcowym [21].

Uczenie modelu AEB; Model jest uczony przy użyciu fragmentów próbek sygnału o ustalonej długości, równej np. 30-krotności średniej długości anomalii, w którym każdy losowo wybrany fragment zawiera w różnych miejscach ciągu próbek po kilka anomalii oznaczonych binarną etykietą. Taka długość fragmentu sygnału powinna gwarantować odpowiednio szeroki kontekst (tło) sygnału do wyizolowania anomalii. Długość fragmentu wejściowego zadawanego sieci autokodera wydłuża czas uczenia, ale nie wpływa na czas detekcji – pod warunkiem, że struktura sieci zawiera wyłącznie warstwy splotowe. Jednak w przypadku dłuższych fragmentów sygnału nie obserwuje się już poprawy wyników [10]. Dodatkowym problemem związanym z warstwami splotowymi w AEB jest zmniejszanie (zwykle dwukrotnie) rozdzielczości sygnału w poszczególnych warstwach poprzez próbkowanie z redukcją (ang. *subsampling*), tj. *stride* = 2, a następnie nadpróbkowanie z użyciem warstw splotowych typu Conv1DTranspose. Dla każdej anomalii występującej w danych uczących należy wygenerować np. 50 losowo wybranych fragmentów zawierających tę anomalię, tak by wyuczyć autokoder niezmiennie względem położenia anomalii w analizowanym fragmencie sygnału. Ponadto, zbiór uczący należy uzupełnić o np. 500 losowo wybranych fragmentów sygnału, zawierających głównie tło, tj. tak by ich liczba była co najmniej dziesięciokrotnie większa niż liczba fragmentów na pewno zawierającymi anomalie (ang. *ground truth*). Dla każdej pary (anomalie i parametr fizyczny) funkcję straty określa błąd średniokwadratowy (MSE), obliczony jako kwadrat różnicy między prawdopodobieństwami uzyskanymi na wyjściu sieci a etykietami oznaczającymi prawdziwe anomalie (por. Rysunek 23). Uczenie modelu prowadzi się przez np. 100 epok, optymalizując wagi sieci za pomocą np. adaptacyjnej metody momentów ADAM [22]. W każdej epoce wszystkie przykłady uczące pogrupowane są w niewielkie podzbiory danych (ang. *mini-batch*).

Detekcja anomalii przez AEB; Segmentacja anomalii w szeregach czasowych pH, T i M polega na zadawaniu modelowi kolejnych fragmentów sygnału o długościach zależnych od rodzaju anomalii. Każdy kolejny fragment próbek powinien nakładać się na poprzedni o około 30%. Odpowiednio wyuczony model wskazuje prawdopodobieństwo wystąpienia anomalii w każdej próbce wybranego fragmentu sygnału. W obszarach nakładających się, gdy dana próbka należy do dwóch niezależnie przetworzonych fragmentów sygnału, prawdopodobieństwo anomalii oblicza się jako maksymalną wartość z dwóch uzyskanych wartości. Obszar 30% nakładania się badanych fragmentów pozwala na uwzględnienie kontekstu anomalii, jeśli jeden z fragmentów zawiera tylko jej część. Z przeprowadzonych badań wynika, że wyższe wartości nakładania się zwiększają złożoność czasową procesu wykrywania anomalii, ale nie poprawiają znacząco skuteczności detekcji [10]. Ostatecznie binarne etykiety anomalii (por. Rysunek 23) są określane poprzez progowanie wartości ich prawdopodobieństw. Standardowy próg wynosi 0,5; powyżej niego wartość próbki jest uznawana za anomálną. Wyniki można jednak próbować poprawiać, określając dla każdego modelu neuronowego optymalny próg dla wybranego kryterium. W tym systemie segmentacji anomalii wybiera się wartość progu, która optymalizuje miarę IOU (ang. *In-tersection Over Union*) obliczaną dla danych uczących [10].

5.1.4. 1D-Unet (UNB)

To podejście różni się od AEB jedynie strukturą wykorzystywanej sieci. Wszystkie opisane wcześniej aspekty uczenia modelu i detekcji anomalii są praktycznie takie same. Początkowo sieci typu U-net stosowane były do segmentacji biomedycznych obrazów dwuwymiarowych [23]. Można je również zastosować do danych jednowymiarowych, takich jak serie czasowe [24]. Podobnie jak w AEB, sieć składa się wyłącznie z warstw splotowych, w tym przypadku 23 warstw traktowanych jako fragment struktury przetwarzającej dane za pomocą parametrów wagowych, jak przedstawiono na Rysunku 24.



Rysunek 24: Schemat sieci neuronowej typu U-net wykorzystanej do segmentacji anomalii

Na tym rysunku warstwy są oznaczone niebieskimi i białymi strzałkami. Niebieskie strzałki oznaczają filtrowanie sygnału przez funkcję splotową Conv1D do sygnału wyjściowego o tej samej rozdzielczości. Szerokość wszystkich filtrów Conv1D wynosi 3. W części kodera, co kilka warstw, rozdzielczość sygnału jest zmniejszana za pomocą operacji MaxPool1D, co wskazują żółte strzałki. Zmniejszenie rozdzielczości jest kompensowane przez zwiększenie liczby filtrów, a tym samym kanałów. W części dekodera, rozdzielczość jest zwiększana za pomocą funkcji Conv1DTranspose z szerokością filtra równą 2. W przeciwieństwie do AEB, UNB ma dodatkowe połączenia pomiędzy warstwami kodera i dekodera (ang. *skip connections*), oznaczone szarymi strzałkami. Polegają one na dodaniu do części dekodera sygnałów uzyskanych na znacznie wcześniejszych etapach przetwarzania w części kodera. Podobnie jak w metodzie AEB na wyjściu, dzięki funkcji *softmax*, uzyskiwany jest wektor prawdopodobieństw wystąpienia anomalii lub braku anomalii dla każdej próbki sygnału wejściowego. Stosując wybraną wartość progową maksymalizującą miarę IOU dla danych uczących [10], wektor prawdopodobieństw przekształcanie jest do postaci binarnej („0” oznacza brak anomalii, „1” oznacza wystąpienie anomalii w danej próbce sygnału).

5.2. Ocena użyteczności

Każdy inteligentny czujnik pomiarowy w ekosystemie Rural IoT realizuje samodzielnie dzienny cykl czyszczenia danych pomiarowych (por. Rysunek 20), które po usunięciu lub korekcie wartości poszczególnych próbek zapisywane są w pamięci lokalnej czujnika. Stąd dane oczyszczone przesyłane są na pokład BSP, a po jego wylądowaniu ładowane do chmury obliczeniowej celem dalszego przetworzenia. Jednym z najważniejszych zadań tej fazy przetwarzania danych pomiarowych jest ocena ich jakości, na podstawie której podejmowana jest decyzja o ewentualnej modyfikacji parametrów modelu wykorzystywanego przez detektor anomalii. Modele te zostały opisane wcześniej w p.5.1. W razie potrzeby zmodyfikowane (zoptymalizowane) wartości parametrów wprowadzane są do kolejnych aktualizacji oprogramowania czujników (por. Rysunek 22).

Ocena jakości danych oczyszczonych odbywa się poprzez pomiar odległości od siebie szeregów czasowych próbek sygnału w zadanych przedziałach czasowych, np. tygodniowych. Spośród wielu znanych metod pomiaru odległości szeregów czasowych [25] najlepiej wybrać metody polegające na wyznaczaniu odległości pomiędzy wektorami cech. W tym celu należy przedstawić każdy szereg czasowy jako pewien wektor w wielowymiarowej przestrzeni cech [6]. Zaletą tego podejścia jest możliwość porównywania ze sobą szeregów o różnej długości, a przy odpowiednio dobranych cechach można uwypuklić różnice między szeregami idealnymi (referencyjnymi) a zawierającymi anomalie. Naczelną zasadą tego wyboru jest zdefiniowanie zestawu cech wrażliwych na anomalie występujące w rzeczywistych danych pomiarowych rejestrowanych przez czujniki. W przypadku ekosystemu Rural IoT są to cztery anomalie opisane na Rysunkach 16-19, dla których rozważono dwa zestawy cech. Pierwszy z nich zawiera cechy opisujące dłuższe fragmenty sygnału, tj. sekwencje próbek obejmujące okresy wielodobowe, np. tygodnie lub miesiące, jak przedstawia to Tabela 5.

Tabela 5: Cechy wielodobowej sekwencji próbek (tygodniowego fragmentu sygnału)

Cecha fragmentu sygnału	Sygnał
Całkowita liczba próbek w szeregu podzielona przez założoną liczbę próbek (długość próby)	T
Odchylenie standardowe	T, M, pH
Wartość maksymalna	T, M, pH
Kurtoza	T, M, pH
Odsetek wartości przekraczających wartość średnią	T
Odsetek próbek, których wartość różni się od średniej o więcej niż odchylenie standardowe	T, M, pH
Średnia, odchylenie standardowe i maksimum bezwzględnych różnic między kolejnymi wartościami	T, M, pH
Współczynnik zmienności	T, pH
Liczba zmian kierunku nachylenia sygnału w stosunku do liczby wszystkich możliwych zmian kierunku	M
Średnia wszystkich lokalnych maksimów	M
Odchylenie standardowe wszystkich lokalnych maksimów	T, M, pH
Odchylenie standardowe wszystkich lokalnych minimów	T, pH
Średnia i odchylenie standardowe odległości między kolejnymi lokalnymi maksimami	T, M
Średnia odległości między kolejnymi lokalnymi minimami	T, M
Odchylenie standardowe odległości między kolejnymi lokalnymi minimami	M
Średnia odległości między minimami lokalnymi a najbliższymi kolejnymi maksimami	M
Odchylenie standardowe odległości między minimami lokalnymi a najbliższymi kolejnymi maksimami	T
Średnia i odchylenie standardowe odległości między lokalnymi maksimami i najbliższymi kolejnymi minimami	T

Drugi zestaw cech zawiera cechy opisujące dobowe (dziennie) fragmenty sygnału, wymienione w Tabeli 6. Są to cechy jednodniowe, ale liczone na podstawie całego ciągu siedmiodniowego. Liczy się

je tak, że dla każdego dnia oblicza się pewną wartość a następnie z tych siedmiu wartości wyznacza się ostateczną jako *min*, *max* lub średnią *m*. Na przykład, wartość maksymalna w drugim wierszu Tabeli 6 jest obliczana jako średnia maksimumów dobowych.

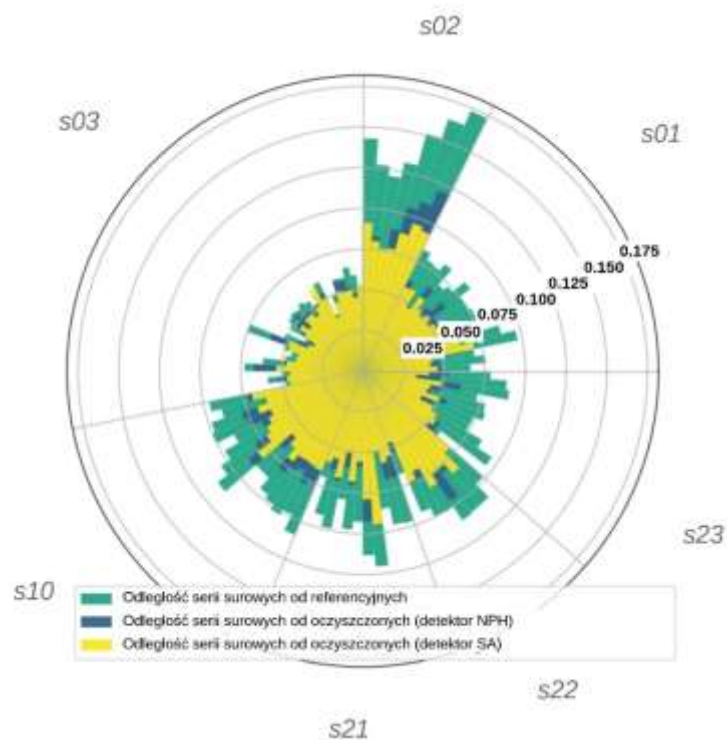
Tabela 6: Cechy dobowej sekwencji próbek (dobowego fragmentu sygnału)

Cecha fragmentu sygnału	Sygnał
Odchylenie standardowe	T[m, max], M[m, max], pH[m, max]
Wartość maksymalna	T[m], M[m], pH[m]
Pozycja pierwszego maksimum w stosunku do długości sekwencji	pH[min], T[max]
Pozycja ostatniego maksimum w stosunku do długości sekwencji	M[m, max], pH[max]
Pozycja pierwszego minimum w stosunku do długości sekwencji	T[max], M[m]
Pozycja pierwszej próbki, dla której suma bezwzględnych wartości próbek po lewej stronie przekracza 50% sumy wszystkich próbek w ciągu	pH[min, max, m], T[min,m]
Liczba zmian kierunku nachylenia sygnału w stosunku do liczby wszystkich możliwych zmian kierunku	M[min, max]
Współczynnik zmienności	T, pH[max,m]

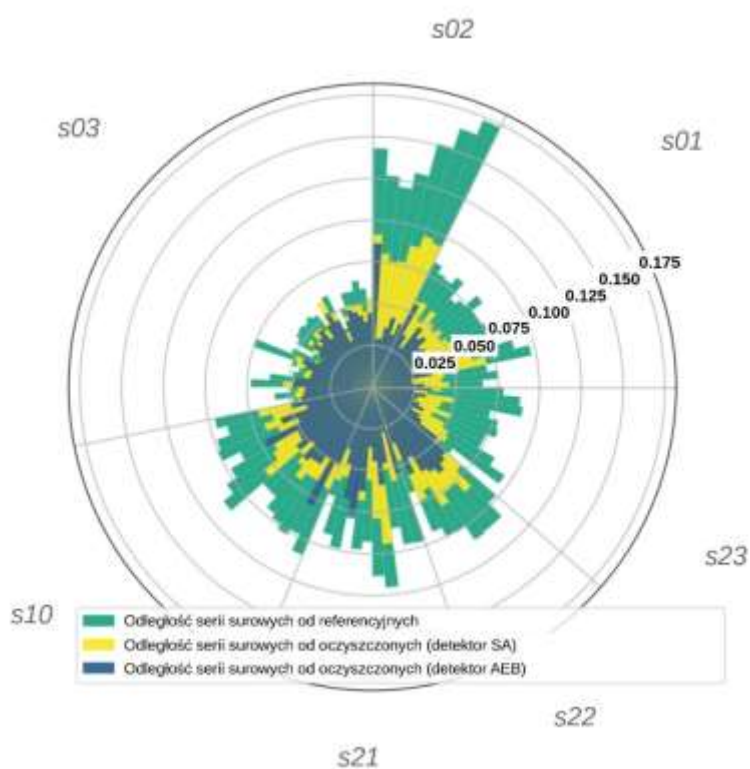
Reasumując, pojedynczy wektor cech opisuje tygodniowy fragment sygnału. Wektor ten zawiera dwa rodzaje cech obliczanych w różny sposób:

- Cechy tygodniowe przedstawione w Tabeli 5 obliczane są na podstawie całego fragmentu bez uwzględnienia podziału na doby. W ten sposób można wyznaczyć np. średnią lub maksymalną temperaturę z całego tygodnia.
- Cechy dobowe przedstawione w Tabeli 6 obliczane są dla każdej z 7 dób a następnie jako ostateczną wartości cechy przyjmuje się wartość min, max lub średnią z obliczonych 7 wartości. W ten sposób można wyznaczyć np. średnią maksymalną temperaturę dnia.

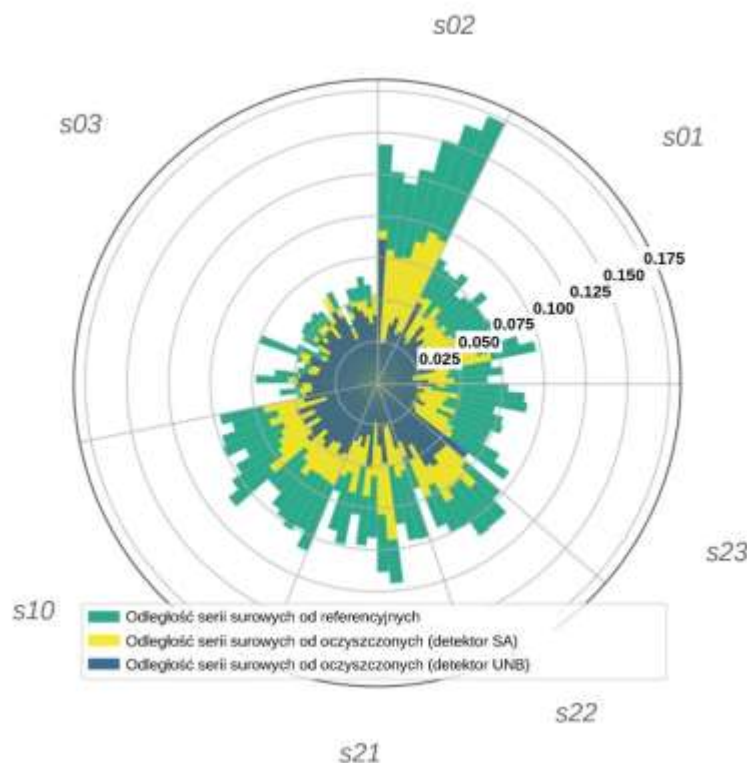
Wrażliwość parametrów wymienionych w obu powyższych tabelach na określone anomalie traktujemy jako hipotezę, która wymaga weryfikacji na podstawie odpowiednio wygenerowanych danych syntetycznych (por. Rysunek 21) lub danych rzeczywistych (por. Rysunek 22) je zawierających. Celem jest natomiast zidentyfikowanie tych cech, które wykazują istotnie różne rozkłady między referencyjnymi (idealnymi) szeregami czasowymi próbek (bez anomalii), a ich odpowiednikami ze znanymi anomaliami. Do tej oceny można zastosować różne testy statystyczne, np. nieparametryczny test Kołmogorowa-Smirnowa, ale należy pamiętać o ryzyku błędnego uznania niektórych nieistotnych cech za istotne. Ryzyko to można znacząco zredukować stosując dodatkowo procedurę statystyczną Benjaminiego–Hochberga (ang. *linear step-up*, LSU), pozwalającą kontrolować współczynnik fałszywych odkryć (FDR) [26]. Przykładowo, po zastosowaniu opisanego wyżej podejścia dla danych pomiarowych parametrów gleby zebranych w całym okresie wegetacji od kwietnia do października 2023 przez 7 czujników *s01–s03*, *s10* i *s21–s23* rozmieszczonych w promieniu ok. 35 km od chmury obliczeniowej [11], dla poszczególnych detektorów opisanych w p.5.1 otrzymano wyniki przedstawione na Rysunkach 25–26.



Rysunek 25: Ocena porównawcza jakości czyszczenia danych z wykorzystaniem detektorów SA i NPH [10]



Rysunek 26: Ocena porównawcza jakości czyszczenia danych z wykorzystaniem detektorów SA i AEB [10]



Rysunek 27: Ocena porównawcza jakości czyszczenia danych z wykorzystaniem detektorów SA i UNB [10]

Przedstawione na powyższych rysunkach wyniki zostały uzyskane przy założeniach opisanych w Dembski, J., [10]. Analizie podlegały tygodniowe fragmenty sygnałów. Na podstawie każdego tygodniowego fragmentu referencyjnego, których było 96, utworzono 10 tygodniowych sygnałów z anomaliami. Całkowita liczba sygnałów z anomaliami wynosiła zatem 960. Zaproponowano 79 parametrów obliczanych dla każdego z trzech rodzajów sygnałów, czyli łącznie $3 \cdot 79 = 237$ cech. Te 237 cech poddano procesowi selekcji, w wyniku którego wybrano 70 cech, które przedstawione są w Tabelach 5 i 6. Celem selekcji było wybranie cech, na podstawie których można odróżniać sygnały z anomaliami od sygnałów bez anomalii. Musiały to być cechy, których rozkłady w tych dwóch zbiorach jak najbardziej się różnią. Do oceny tego, czy rozkłady danej cechy w dwóch zbiorach (bez anomalii i z anomaliami) statystycznie istotnie się różnią wykorzystano, jak wspomniano wcześniej, test Kołmogorowa-Smirnowa (K-S), a do kontroli wskaźnika fałszywych odkryć (FDR) zastosowano procedurę Benjamini-Hochberga. Jako odległość między dwoma wektorami cech przyjęto wartość metryki Manhattan podzielonej przez wymiar wektora.

6. Planowanie misji

Planowanie i efektywna realizacja misji pomiarowych realizowanych z wykorzystaniem BSP wymaga uwzględnienia trzech zasadniczych czynników: jego charakterystyk operacyjnych (lotnych) określających możliwy do pokrycia zakres obszaru objętego pomiarami, na którym umieszczone zostały czujniki pomiarowe, przepisów prawa lotniczego dotyczących przestrzeni powietrznej nad tym obszarem oraz uwarunkowań środowiskowych mających wpływ na jakość transmisji danych z poziomu gruntu do wędrownego bramki przenoszonej przez BSP. W przypadku ekosystemu pomiarowego typu Rural IoT zakłada się wykonywanie lotów nad obszarami lądowymi zarówno w trybie VLOS (ang. Visual Line of Sight), tj. przy założeniu utrzymywaniu bezpośredniego i samodzielnego kontaktu wzrokowego pilota (operatora) z BSP przez cały czas lotu – od startu do lądowania, jak i BVLOS (ang. Beyond Visual

Line of Sight), gdy pilot nie utrzymuje stałego kontaktu wzrokowego z BSP i steruje nim za pośrednictwem aplikacji przekazującej dane z kamery, mapę z położeniem BSP oraz informacje o jego parametrach lotu (wysokość, prędkość, kurs).

6.1. Charakterystyki operacyjne BSP

Misje pomiarowe w ekosystemie Rural IoT można realizować z wykorzystaniem szerokiego spektrum bezzałogowych platform latających – od małych wielowirnikowców (multikopterów) przeznaczonych głównie do użytku prywatnego po profesjonalne BSP eksploatowane przez firmy świadczące odpłatnie usługi lotnicze, jak np.:

- *DJI Inspire* [34]; seria popularnych dronów przeznaczonych głównie do filmowania i wykonywania zdjęć z powietrza, o masie startowej do 4,5 kg. Nowsze modele, takie jak Inspire 3, oferują m.i. precyzyjne pozycjonowanie RTK (dokładność do 1 cm), wydłużony czas lotu do 28 minut i moduł transmisji O3 Pro z zasięgiem do 8 km. Za ich pomocą możliwe jest wykonywanie misji pomiarowych nad obszarami lądowymi w trybie VLOS. Czas lotu wynosi typowo 15-18 min z ładunkiem użytecznym ok. 2,5–3 kg. Zaleca się ograniczanie użytecznego czasu misji do 10–12 min, pozostawiając margines energii min. 10% na powrót i nieprzewidziane manewry. Z tym zapasem zasilania możliwe jest jednorazowe pokrycie obszaru o powierzchni rzędu 0,5–1 km².
- *Farada G1* [35]; profesjonalny BSP dalekiego zasięgu, w postaci hybrydowego układu płatowca ze śmigłem ciągnącym oraz czterema zespołami napędowymi w układzie wielowirnikowca, które umożliwiają pionowy start i lądowanie. Platforma ta, o masie startowej ok. 14 kg przeznaczona jest do zadań związanych z mapowaniem terenu w geodezji, kartografii, geologii, planowania przestrzennego, wykonywania inspekcji infrastruktury naziemnej, jak również w misjach związanych z transportem powietrznym małych ładunków. Za jego pomocą możliwe jest realizowanie misji w trybie BVLOS, z wykorzystaniem Farada Cloud Control – opracowanego przez producenta protokołu transmisji sterująco-telemetrycznej oraz wysyłki danych po LTE. Czas lotu z ładunkiem użytecznym 3 kg, wynoszący typowo 1,5 h z prędkość przelotową 85-90 km/h zapewnią pokrycie obszaru o powierzchni rzędu 20–30 km².

Z badań przeprowadzonych w projekcie Rural IoT wynika, że dla zadań małoobszarowych i punktowych, gdy realizowane są tylko misje w trybie VLOS, wystarczy w zupełności mały wielowirnikowiec. Rozwiązanie takie charakteryzuje się niskimi kosztami - w szczególności, gdy wybrany BSP będzie eksploatowany tylko na własny użytek, np. jednego lub kilku rolników w danej miejscowości. Z kolei dla zadań wielkoobszarowych, gdzie loty będą wykonywane w trybie BVLOS (duże gospodarstwa rolne lub kompleksy leśne) niezbędne będzie wykorzystanie profesjonalnych BSP, eksploatowanych przez firmy świadczące usługi lotnicze i dysponujące odpowiednim zapleczem. Każdorazowo wybór platformy latającej powinien wynikać z wymaganej powierzchni do pokrycia pomiarem, dostępności łączności naziemnej (LTE), uprawnień operatora do wymaganej kategorii lotów VLOS lub BVLOS, oraz obowiązujących ograniczeń dla rodzaju i kategorii przestrzeni powietrznej, w której miałyby się odbywać loty.

Po dokonaniu wyboru platformy w oparciu o wyżej wymienione zagadnienia przy planowaniu konkretnych misji pomiarowych należy dokonać oceny warunków pogodowych. Muszą one bezwzględnie spełniać minima podane przed producenta wykorzystywanej platformy; z reguły instrukcja użytkowania BSP podaje:

- maksymalną dopuszczalną prędkość wiatru (typowo do 6–8 m/s dla mniejszych wielowirnikowców i do 10 m/s dla większych BSP VTOL o masie startowej i rozmiarach jak Farada G1);
- brak opadów atmosferycznych;
- dopuszczalny zakres temperatury użytkowania (typowo od -10°C do 40°C); w przypadku bardzo niskich temperatur (poniżej -10°C) zaleca się zachowanie szczególnej ostrożności i unikanie startu w ekstremalnych warunkach, ponieważ może to prowadzić do nieoczekiwanej awarii sprzętu;
- w przypadku lotów w strefach o podwyższonym poziomie zakłóceń GNSS (por. p.6.3) może być konieczne przełączenie trybu lotu BSP na ręczny w celu zapobieżenia utracie kontroli przez operatora, a w skrajnych przypadkach nawet ucieczce i upadkowi BSP.

Z kolei organizacja lotów powinna obejmować następujące zagadnienia:

- wydzielone i odpowiednio oznakowane stanowisko operatora/dowodzenia z wizualnym kontaktem z BSP (VLOS) lub z redundancją łączności (BVLOS/LTE);
- rotacyjnie ładowane zestawy akumulatorów (min. trzy dla małego wielowirnikowca lub z zapasem na dwie pełne misje dla dużego VTOL);
- listy przedstartowe i procedury awaryjne na wypadek utraty sygnału GNSS, utraty łączności z BSP i awaryjnego lądowania (należy przewidzieć miejsca zapasowe);
- szczegółowy plan kampanii, obejmujący realistycznie 3–5 lotów dziennie dla małego wielowirnikowca lub 1–2 długie misje z przerwą serwisową dla dużego VTOL;
- przedstartowa kontrola montażu w kadłubie BSP bramki pokładowej z uwzględnieniem położenia środka ciężkości, właściwej separacji anteny bramki od anten GNSS oraz możliwości szybkiej wymiany pokładowego źródła zasilania. Ponadto bezpośrednio przed startem wymagane jest wykonanie testu aktualnego zasięgu sygnału LoRaWAN z czujników.

6.2. Uwarunkowania prawne

Od 31 grudnia 2020 roku na terenie wszystkich krajów Unii Europejskiej oraz w Lichtensteinie, Norwegii i Szwajcarii obowiązują wspólne przepisy dotyczące bezzałogowych systemów powietrznych oraz operatorów bezzałogowych systemów powietrznych z państw trzecich [36]. Loty pomiarowe w systemie Rural IoT mogą być prowadzone w dwóch kategoriach opisanych w tych przepisach:

- **VLOS** (ang. Visual Line of Sight) odnosi się do operacji BSP w zasięgu wzroku operatora i zaliczana jest do kategorii otwartej o niskim ryzyku. W jej ramach można wykonywać loty do 120 m nad ziemią z BSP o masie poniżej 25 kg. Dozwolone działania obejmują m.in. fotografowanie i filmowanie oraz zbieranie danych. Jednak w tym przypadku (loty pomiarowe w systemie Rural IoT) wymagana jest rejestracja w systemie jako operator systemu bezzałogowego statku powietrznego (SBSP), po zaliczeniu szkolenia on-line [37] oraz wykupienie polisy OC.
- **BVLOS** (ang. Beyond Visual Line of Sight) w kategorii „szczególnej” to operacje BSP wykonywane poza zasięgiem wzroku pilota, które wiążą się ze podwyższonym ryzykiem i wykraczają poza ramy kategorii „otwartej”. Ze względu na konieczność posiadania przez operatora odpowiednich uprawnień oraz zezwolenia na operacje do Urzędu Lotnictwa Cywilnego (ULC), a także realizacji procedur zgłaszania każdorazowo lotu do Polskiej Agencji

Żegluga Powietrznej (PAŻP), oceny ryzyka SORA i akceptacji ULC loty pomiarowe w tej kategorii powinny być prowadzone przez firmę świadczącą usługi lotnicze z wykorzystaniem BSP.

W przestrzeni powietrznej, w której mają być wykonywane operacje BSP wyróżnia się strefy geograficzne (geo-strefy) w których obowiązują różne ograniczenia dotyczące lotów; ich pełna klasyfikacja wraz organami zarządzającymi, od których należy każdorazowo uzyskać odpowiednie zgody została opublikowana na stronie Polskiej Agencji Żegluga Powietrznej (PAŻP) [38]. Ich szczegółowy opis wykracza poza zakres tego raportu. PAŻP zaleca wykorzystanie w tym zakresie darmowej aplikacji DroneTower do pobrania na urządzenia mobilne [39]. Na przykład, oznaczenia MCTR, DRA-R i ATZ na mapie wskazują odpowiednio: obecność obszaru kontrolowanego przez lotnisko wojskowe (Military Control Zone), dla którego wymagana jest zgoda Szefostwa Służby Ruchu Lotniczego tego lotniska, kategorię przestrzeni powietrznej, w której wykonywanie lotów jest ograniczone (Restricted Geographical Zone) i wymaga uzyskania zgody organu nadzorującego, obszar w pobliżu lotniska aeroklubowego (Aerodrome Traffic Zone), dla którego wytyczne ustalane są przez aeroklub.

Ze względu na występujące w pobliżu obszaru objętego pomiarami geo-strefy, planując loty w praktyce należy każdorazowo:

1. sprawdzić aktywność stref w aplikacji DroneTower [39],
2. jeśli aplikacja wskazuje, że strefa jest czynna – uzyskać zgodę organu wskazanego w opisie strefy i stosować narzucone warunki (czas, wysokość, korytarz),
3. prowadzić za pośrednictwem tej aplikacji *check-in* przed startem i *check-out* po lądowaniu, rejestrując przebieg operacji.

Zgodnie przepisami prawa lotniczego zabronione jest wykonywanie lotów nad zgromadzeniami osób, naruszanie czynnych stref, latanie bez wymaganych zgód w strefach DRA-R, a także startu przy niespełnieniu minimalnych wymogów pogodowych i bez przygotowanych procedur awaryjnych. Należy pamiętać, że operator BSP odpowiada cywilnie za wyrządzone szkody. Z tego względu zaleca się podczas lotu utrzymywanie stałego bufora bezpieczeństwa ponad 30 m od osób (w poziomie), obiektów i maszyn polowych, systematyczne dokumentowanie lotów (np. prowadzenie dziennika lotów i dziennika okresowych przeglądów technicznych BSP) oraz posiadanie ważnej polisy OC.

6.3. Wskazówki i dobre praktyki

Poza technicznymi i formalno-prawnymi wymaganiami realizacji misji pomiarowych w ekosystemie Rural IoT z wykorzystaniem BSP warto także wymienić kilka dodatkowych uwag praktycznych, zebranych w trakcie różnych misji pomiarowych realizowanych w projekcie:

1. Poza strefami ograniczeń najczęściej możliwe są operacje VLOS do 120 m AGL. Zaleca się uprzedni kontakt z najbliższym lądowiskiem/zarządcą terenu (jeśli istnieje) w celu wyznaczenie lądowiska zapasowego oraz sprawdzenie lokalnych uwarunkowań i przeszkód, jak np. linie energetyczne, uczęszczane drogi czy tory kolejowe.
2. W północnej Polsce od pewnego czasu odnotowuje się okresy silnych lub bardzo silnych zakłóceń sygnału GNSS. Konsekwencją jest niestabilność trybów automatycznych (utrata Position Hold, błędy misji waypoint). Planowanie lotów powinno zakładać możliwość przejścia do trybu ręcznego, obniżenie pułapu i skrócenie lotu. Warto zatem każdorazowo:
 - wykonywać test nawigacji przed misją w celu sprawdzenia jakości sygnału GNSS, przeprowadzać kalibrację kompasu oraz przetestować łączność bramki z czujnikami,

- w przypadku wykorzystywania automatycznego trybu lotu BSP stosować „krótkie segmenty” automatyki (krótkie odcinki waypoint) z możliwością natychmiastowego przejścia sterowania,
- zdefiniować zapasowe punkty lądowania i powrotu oraz przyjąć większy niż zwykle margines energii na powrót BSP do miejsca startu,

Ponadto warto systematycznie trenować pilotaż BSP w trybie ręcznym w celu utrzymywania biegłości w locie bez wykorzystania GPS, tj. tylko ze stabilizacją żyroskopową i orientacją magnetyczną.

3. Na użytek własny zalecane jest przygotowanie przez operatora minimalnej listy kontrolnej określającej:
 - **Cel i zakres misji:** lista czujników, mapa obszaru, wymagane pokrycie, dostępne okno czasowe;
 - **Reżim prawny:** VLOS czy BVLOS, identyfikacja stref, uzyskane zgody, polisa OC, rejestracja operatora w Krajowym Systemie Informacji Dronowej.
 - **Platforma latająca i jej ładunek:** masa, montaż bramki, test połączenia LoRaWAN, bilans energii zasilania.
 - **Łączność:** sterowanie, telemetria, test LTE (jeśli wykorzystywane), łącza alternatywne (zapasowe).
 - **Pogoda i GNSS:** wiatr (kierunek, prędkość), opady (prognoza), temperatura, weryfikacja stabilności sygnału GNSS.
 - **Procedury:** checklisty (start, lot, lądowanie), dostępność funkcji RTH (automatycznego powrotu BSP do miejsca startu), lądowiska zapasowe, dziennik lotu i archiwizacja telemetrii.

6.4. Uwarunkowania środowiskowe

Zapewnienie jak najlepszych warunków propagacji z równoczesnym optymalnym wykorzystaniem zapasu energii zasilania naziemnego czujnika pomiarowego wymaga skonfigurowania czterech podstawowych parametrów protokołu LoRaWAN zaimplementowanego w czujniku; są to:

- **Rozmiar pakietu przesyłanych danych PL (ang. PayLoad);** zbyt duży wolumen danych do jednorazowego przesłania do bramki wędrównej w dostępnym okienku czasowym Δt oznacza konieczność wydłużenia czasu lotu BSP w zasięgu czujnika, co wpływa z kolei na jego zasięg. Ponadto dłuższe czasy nadawania powodują szybsze rozładowanie akumulatora zasilającego czujnik. Stąd podczas programowania czujnika pomiarowego zachodzi konieczność analizy dynamiki (zmienności) każdego mierzonego sygnału w celu wyznaczenia granicznych wartości okresu próbkowania – tak by możliwie w największym stopniu zredukować liczbę wysyłanych próbek. Wyniki takiej analizy dla sygnałów temperatury (T), wilgotności (M), kwasowości (pH) i nasłonecznienia (PV) gleby przedstawiono wcześniej w Tabeli 4.
- **Współczynnik rozpraszania SF (ang. Spreading Factor);** modulacja częstotliwości nośnej poprzez jej liniową zmianę w czasie (ang. chirp signal) pozwala kodować zero-jedynkowo bity przesyłanej informacji jako wzrost (bit '1') lub spadek (bit '0') częstotliwości w zadanym przedziale czasu. Im ten przedział jest krótszy tym transmisja jest szybsza a SF niższy, ale za cenę niższej odporności transmisji na zakłócenia i mniejszego zasięgu. Liczbę zmian (ang. chirps) przypadających na pojedynczy symbol przesyłanej informacji oblicza się jako 2^{SF} , zatem każdy wzrost wartości SF zmniejsza o połowę szybkość transmisji, a tym samym podwaja czas transmisji i zużycie energii. Wartość SF można ustawiać programowo w zakresie od 6 do 12 [32].

- **Moc nadawania TP (ang. Transmission Power);** moc nadawania określana jest w dBm, jednostce określającej w skali logarytmicznej moc sygnału względem 1 mW. Dodatnie wartości TP oznaczają moc sygnału powyżej 1 mW, a ujemne odpowiednio poniżej tego poziomu. Silniejszy sygnał oznacza większy zasięg, ale równocześnie większy pobór energii zasilania. Dla większości urządzeń wykorzystujących technologię LoRaWAN wartość TP ustawiana jest programowo od 2 dBm do 20 dBm.
- **Czas trwania transmisji ToA (ang. Time on Air);** rozważając dobór wartości SF należy pamiętać, że przesyłanie tej samej ilości danych przy wyższym współczynniku SF wymaga dłuższego czasu transmisji, czyli wartości ToA. Im ta wartość jest wyższa, tym dłużej BSP musi przebywać w zasięgu czujnika i tym wyższe jest zużycie energii zasilania przez czujnik. W typowych zastosowaniach wartość parametru ToA dla pojedynczego pakietu danych może się wahać od dziesiątek milisekund do kilku sekund, w zależności od przyjętych wartości PL i SF.
- **Wskaźnik mocy odbieranego sygnału RSSI (ang. Received Signal Strength Indicator);** moc odbieranego sygnału radiowego jest również wyrażana w dBm. Im ta wartość jest wyższa tym lepsza jest jakość połączenia bramki na pokładzie BSP z czujnikiem na ziemi. Typowy zakres zmienności parametru RSSI może wahać się od 0 dBm (odbior sygnału idealny) do -100 dBm (słaby odbiór sygnału). Konkretna wartość RSSI mierzona na pokładzie BSP w trakcie wykonywania misji pomiarowej silnie zależy od jego aktualnego położenia (odległość od czujnika), warunków terenowych (zabudowania, wzniesienia, pokrywa leśna) oraz obecności w pobliżu czujników pomiarowych źródeł promieniowania elektromagnetycznego, takich jak urządzeń radiokomunikacyjne, radiolokacyjne czy radionawigacyjne.

Dla ustalenia konfiguracji wartości powyższych parametrów transmisji danych z czujników naziemnych na pokład BSP dla zakładanego scenariusza pomiarowego i konkretnego środowiska operacyjnego – pożądanej z punktu widzenia zużycia energii i jakości transmisji – niezbędne jest przeprowadzenie badań eksperymentalnych z wykorzystaniem zaawansowanych symulatorów [33] lub w drodze pomiarów w docelowym środowisku planowanego ekosystemu Rural IoT. Warto zauważyć, że tylko dla samych urządzeń wykorzystujących LoRaWAN, liczba wszystkich możliwych kombinacji do zbadania przekracza 6700 [32]. Badania takie dla rozwiązań opisanych w Rozdziale 2 tego raportu zostały przeprowadzone zarówno z wykorzystaniem symulatora jak i pomiarów poligonowych dla trzech klas scenariuszy pomiarowych [9]:

- **rolnictwa precyzyjnego**, gdy loty BSP wykonywane są nad polami uprawnymi, a rejestrowane parametry gleby obejmują temperaturę (T), wilgotność (M), kwasowość (pH) i nasłonecznienie (PV);
- **ochrony lasów** przed pożarami, gdy loty BSP wykonywane są nad kompleksami leśnymi, a rejestrowane parametry gleby obejmują temperaturę (T), wilgotność (M) i nasłonecznienie (PV);
- **ochrony powodziowej**, gdy loty BSP wykonywane są wzdłuż umocnień przeciwpowodziowych rzek, a rejestrowane parametry gleby obejmują wilgotność (M) i nasłonecznienie (PV).

Wspomniane badania przeprowadzono przy założeniu maksymalnego rozmiaru przesyłanego pakietu $PL_{max} = 120$ B, minimalnej dopuszczalnej wartości $RSSI_{min} > -100$ dBm oraz maksymalnej dopuszczalnej wartości $ToA_{max} < 1$ s, w odległościach czujnika od przelatującego BSP w przedziałach do 100 m, od 100 m do 200 m oraz od 300 m do 2000 m. Ta ostatnia badana odległość okazała się graniczna, ze względu na osiągnięcie minimalnej dopuszczalnej wartości $RSSI = 100$ dBm. Przy tych założeniach przeprowadzone pomiary wykazały, że optymalny dla wymienionych wyżej warunków terenowych przedział zmienności współczynnika rozpraszania wyniósł od $SF7 = 7$ dBm do $SF8 = 8$ dBm, a mocy

nadawania od TP8 = 8 dBm do TP14 = 14 dBm. Dla tych przedziałów zmienności mierzony czas trwania transmisji zmieniał się od ToA = 222 ms do ToA = 537 ms, a zużycie energii zasilania przez czujnik od 0,13 mWh do 0,16 mWh. Przy pojemności 300 mAh akumulatora zasilającego czujnik napięciem 3,3 V łączny zapas dostępnej energii zasilania wynosi E=990 mWh, zatem dobowe zużycie energii na transmisję pakietu danych $PL_{max} = 120$ B (por. p.4.3.4) jest pomijalnie małe (0,02%) względem całego dostępnego zapasu energii.

6.5. Klasy scenariuszy

Wyznaczone eksperymentalnie realistyczne wartości parametrów SF, TP, ToA i RSSI protokołu LoRaWAN opisanych wcześniej zostały zestawione z trzema klasami scenariuszy pomiarowych możliwych do realizacji w ekosystemie Rural IoT w Tabeli 7. Dodatkowo, dla każdego scenariusza przyjęto trzy warianty odległości czujnika od bramki na pokładzie przelatującego BSP.

Tabela 7: Rekomendowane konfiguracje parametrów ekosystemu Rural IoT dla różnych klas scenariuszy [9]

Odległość (m)	PL (B)	SF	TP (dBm)	ToA (ms)	RSSI min (dBm)	Klasy scenariuszy
0-100	120	7	10	342	-80	Rolnictwo precyzyjne
	106	7	8	505	-80	Ochrona lasów
	74	7	8	222	-80	Ochrona powodziowa
100-200	120	7	10	342	-90	Rolnictwo precyzyjne
	106	8	12	537	-90	Ochrona lasów
	74	7	10	222	-90	Ochrona powodziowa
300-2000	120	7	14	342	-100	Rolnictwo precyzyjne
	106	8	14	537	-100	Ochrona lasów
	74	7	14	222	-100	Ochrona powodziowa

Jak wynika z powyższej tabeli, ze względu na minimalne tłumienie sygnału, do lotów w niewielkiej odległości od czujników (do 100 m) wystarczą konfiguracje o stosunkowo niskiej mocy nadawania TP8 = 8...10 dBm i umiarkowanej wartości współczynnika rozpraszania SF7 = 7. Zapewniają one stosunkowo wysokie wartości RSSI > -80 dBm i najkrótsze możliwe czasy nadawania ToA = 222...342 ms dla efektywnego przesyłania pakietów o wymaganym (por. Tabela 4) rozmiarze PL = 74...120 B przy niewielkim zużyciu energii; to czyni je wysoce niezawodnymi i energooszczędnymi nawet w lasach. Przy większych odległościach przelatującego BSP od czujników, rzędu 100-200 m potrzebna jest nieco wyższa moc nadawania TP12 = 12 dBm i wyższy współczynnik rozpraszania SF8 = 8. Konfiguracja ta nadaje się szczególnie do stosowania w lasach, zachowując odporność na zakłócenia ze strony liści i gałęzi, a jednocześnie oferując RSSI > -90 dBm. Ze względu na minimalne zakłócenia i otwarty teren w lotach nad polami uprawnymi i wzdłuż rzek, do zapewnienia niezawodnej komunikacji i energooszczędności wystarczy moc nadawania TP10 = 10 dBm i współczynnik rozpraszania SF7 = 7. Wreszcie dla skrajnych odległości przelatującego BSP od czujników (do 2000 m) trzeba się liczyć z większym wydatkiem energetycznym dla utrzymania wymaganej jakości transmisji. Z Tabeli 7 wynika, że dla uzyskania RSSI > -100 dBm niezbędny jest poziom mocy nadawania co najmniej TP14 = 14 dBm, przy podobnych jak poprzednio współczynnikach rozpraszania, tj. SF8=8 dla lasów i SF7=7 dla pól uprawnych i rzek.

Przyjmując zakres zmienności rozmiaru pakietu danych PL = 74...120 B, całkowita ilość danych zbieranych przez BSP podczas pojedynczej misji mieści się w zakresie od 2 MB (ochrona powodziowa, PL=74) do 3,4 MB (rolnictwo precyzyjne, PL=120); jest to stosunkowo niewielka ilość danych do

obsłużenia przez komputer pokładowy BSP. Ponadto, w zależności od charakteru monitorowanych zjawisk glebowych, rozmieszczenie czujników na ziemi może być różne. Na przykład, monitorowanie przesiąkania wałów przeciwpowodziowych będzie wymagało większego zagęszczenia czujników na szczególnie wrażliwych odcinkach rzek, np. $d < 100$ m. Dla pól uprawnych wystarcza $d = 100...200$ m, zaś dla monitorowania ryzyka pożarowego lasów odległość ta może być $d > 300$ m.

Literatura

- [1] Thomas, G. W. (1996): Chapter 16 - Soil pH and soil acidity; Sparks, D. L., et. al (eds.), Methods of soil analysis; SSSA Book Series, Madison, Wisconsin, Soil Science Society of America, 475–490; <https://doi.org/10.2136/sssabookser5.3.c16>
- [2] Wang, X., Wang, C. (2020): Time series data cleaning: a survey, IEEE Access, vol. 8, 1866–1881; <https://doi.org/10.1109/ACCESS.2019.2962152>
- [3] Arduino Home: <https://www.arduino.cc/>
- [4] Siebeneicher, H. (2022): 1-Wire Protocol, <https://docs.arduino.cc/learn/communication/one-wire>
- [5] Rocha, A. M. D., Oliveira, M. A. D., José F. M., P., Cavaleiro, G. G. H. ABP vs. OTAA activation of LoRa devices: an Experimental Study in a Rural Context. (2023) Proc. Int. Conf. on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, pp. 630–634, <https://doi.org/10.1109/ICNC57223.2023.10074553>.
- [6] Dembski, J., Kołakowska, A., Wiszniewski, B. (2025). Automatic Cleaning of Time Series Data in Rural Internet of Things Ecosystems That Use Nomadic Gateways. Sensors, 25(1), 189. <https://doi.org/10.3390/s25010189>
- [7] Kołakowska, A., Wiszniewski, B., Dembski, J. (2025). Hierarchical Alignment of Multiple Time Series with Missing Timestamps. Proc. 23rd ACM Conf. on Embedded Networked Sensor Systems. ACM, New York, NY, USA, 616–617. <https://doi.org/10.1145/3715014.3724035>
- [8] Krawczyk, H., Wiszniewski, B. (2025). Collaborative Learning as a Service – a Blueprint for a Cloud Based Rural IoTs Deployment Facility. Proc. 15th Int. Conf. on Processing and Applied Mathematics, PPAM 2024, Ostrava, Czech Republic, September 8–11, 2024, LNCS 15579, Part I. Springer-Verlag, Berlin, Heidelberg, 328–342. https://doi.org/10.1007/978-3-031-85697-6_22
- [9] Vaghalandari, N.G., Kołakowska, A., Grundig, Ch., Dongchen Li, Wiszniewski, B., Daniel, K., Obermaisser, R. (2025). Performance Evaluation of Nomadic Data Collection with a UAV-LoRa System. 21st International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Lucca, Italy, 2025, pp. 1–8, <https://doi.org/10.1109/DCOSS-IoT65416.2025.00143>.
- [10] Dembski, J., Wiszniewski, B., Kołakowska, A. (2025). Anomaly Detection and Segmentation in Measurement Signals on Edge Devices Using Artificial Neural Networks. Sensors, 25(17), 5526. <https://doi.org/10.3390/s25175526>
- [11] Dembski, J., Kołakowska, A., Wiszniewski, B. Rural IoT soil data, IEEE Dataport, July 14, 2024, doi:10.21227/0j1h-ew11
- [12] Hernandez, R.J. (2022) Building IoT Visualizations using Grafana. Packt Publishing Ltd
- [13] Abdelmadjid, L., Boucheham, B. (2021). A Comparison Study of Dynamic Time Warping's Variants for Time Series Classification. Int. J. of Informatics and Applied Mathematics, 4(1), 56–71.
- [14] Breiman, L. (2001) Random Forests. Machine Learning 45, 5–32. <https://doi.org/10.1023/A:1010933404324>
- [15] CDSE - Copernicus Data Space Ecosystem, <https://dataspace.copernicus.eu/>
- [16] OData - Open Data Protocol, <https://www.odata.org/>
- [17] STAC SpatioTemporal Asset Catalogs, <https://stacspec.org/en>
- [18] OpenSearch, <https://opensearch.org/>
- [19] SentinelHub - API Reference, <https://docs.sentinel-hub.com/api/latest/reference/>

- [20] Wahab, O. A., Mourad, A., Otrok, H., Taleb, T. (2021). Federated Machine Learning: Survey, Multi-Level Classification, Desirable Criteria and Future Directions in Communication and Networking Systems, IEEE Communications Surveys & Tutorials, 23(2), 1342-1397, doi: 10.1109/COMST.2021.3058573.
- [21] Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning, 2018, arXiv:stat.ML/1603.07285.
- [22] Kingma, D.P., Ba, J. Adam: A Method for Stochastic Optimization. Proc. 3rd Int. Conf. on Learning Rep-resentations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015; Bengio, Y.; LeCun, Y., Eds., 2015.
- [23] Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015, 941, arXiv:cs.CV/1505.04597.
- [24] Perslev, M.; Jensen, M.H.; Darkner, S.; Jennum, P.J.; Igel, C. U-Time: a fully convolutional net-work for time series segmentation applied to sleep staging. In Proceedings of the Proceedings of the 33rd International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 2019
- [25] Górecki, T.; Piasecki, P. A Comprehensive Comparison of Distance Measures for Time Series Classification. In Stochastic Models, Statistics and Their Applications; Springer International Publishing: Cham, Switzerland, 2019; pp. 409–428.
- [26] Benjamini, Y.; Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. J. R. Stat. Soc. Ser. B (Methodol.) 1995, 57, 289–300.
- [27] OpenStack: Documentation for 2025.1 (April 2025), <https://docs.openstack.org/2025.1/>
- [28] Hermans, K.: Mastering InfluxDB Database: A Comprehensive Guide to Learn InfluxDB Database. Cybellium Ltd (2023)
- [29] Hernandez, R.J.: Building IoT Visualizations using Grafana. Packt Publishing Ltd (2022)
- [30] Figurski, M., Nykiel, G., Proficz, J. (2021). WRF-METEOPG: numerical weather forecast data for Poland - Days 253-259, Year 2021 (1–) [Dataset]. Gdańsk University of Technology. <https://doi.org/10.34808/ftb2-1r16>
- [31] Kane, S.; Matthias, K.: Docker. Niezawodne kontenery produkcyjne. Praktyczne zastosowania. Helion, 2024
- [32] Bor, M., Roedig, U.: LoRa Transmission Parameter Selection. (2017). Proc 13th Int. Conf. on Distributed Computing in Sensor Systems (DCOSS), Ottawa, ON, Canada, 2017, pp. 27-34, doi: 10.1109/DCOSS.2017.10.
- [33] Idris, S., Karunathilake, T., Förster, A. Survey and comparative study of LoRa-enabled simulators for Internet of Things and wireless sensor networks. (2022) Sensors, vol. 22, no. 15, p. 5546.
- [34] DJI Camera Drones (strona producenta) <https://www.dji.com/pl>
- [35] Farada G1 (strona producenta) <https://faradagroup.com/farada-g1/>
- [36] Rozporządzenie delegowane komisji (UE) 2019/945 z dnia 12 marca 2019 r. w sprawie bezałogowych systemów powietrznych oraz operatorów bezałogowych systemów powietrznych z państw trzecich, <https://eur-lex.europa.eu/legal-content/PL/TXT/PDF/?uri=CELEX:32019R0945>
- [37] Krajowy System Informacji Dronowej, <https://drony.gov.pl/>
- [38] Strefy geograficzne, Polska Agencja Żeglugi Powietrznej, <https://www.pansa.pl/strefy-geograficzne/>
- [39] DroneTower, <https://drony.gov.pl/drone-tower>

Słownik pojęć i skrótów

ABP	Metoda aktywacji urządzenia końcowego w sieci bezprzewodowej, polegająca na ręcznym, stałym przypisaniu kluczy bezpieczeństwa i adresu urządzenia (DevAddr) podczas jego wytworzenia (ang. Activation By Personalization).
ADAM	Algorytm stosowany w uczeniu maszynowym, który nadaje się szczególnie do rozwiązywania problemów optymalizacji parametrów modelu i danych dużej skali (ang. Adaptive Moment Estimation).
AGL	Wysokość (lotu) nad poziomem terenu (ang. Above Ground Level).
ATZ	Obszar w pobliżu lotniska aeroklubowego, dla którego wytyczne ustalane są przez aeroklub (ang. Aerodrome Traffic Zone).
BLOS	Loty BSP wykonywane poza zasięgiem wzroku pilota (ang. Beyond Visual Line of Sight).
BSP	Bezzałogowy statek powietrzny.
CDSE	Publiczna platforma zapewniająca darmowy dostęp do danych satelitarnej obserwacji Ziemi pochodzących z programu Copernicus (ang. Copernicus Data Space Ecosystem).
DTW, DDTW	Algorytmy wykorzystywane do mierzenia podobieństwa między dwoma sekwencjami sygnałów czasowych o różnej długości (ang. Dynamic Time Warping, Derivative Dynamic Time Warping).
DRA-R	Jedna z kategorii przestrzeni powietrznej, w której wykonywanie lotów jest ograniczone (ang. Restricted Geographical Zone) i wymaga uzyskania zgody organu nadzorującego.
FDR	Odsetek wyników fałszywie pozytywnych w stosunku do wszystkich wyników uznanych za istotne (ang. False Discovery Rate).
GFS	Globalny model prognozy pogody, którego dane są dostępne publicznie (ang. Global Forecast System).
GNSS	Globalny system nawigacji satelitarnej, obejmujący wszystkie dostępne systemy satelitarne (ang. Global Navigation Satellite System).
GPS	Amerykański system nawigacji satelitarnej (ang. Global Positioning System).
HRWRF	Numeryczny system prognozowania pogody wykorzystujący siatkę o wysokiej rozdzielczości (ang. High Resolution Weather Research and Forecasting).
IaaS	Warstwa usług chmurowych umożliwiających zdalny dostęp do serwerów, pamięci masowej i sieci (ang. Infrastructure as a Service).
IoU	Metryka służąca do oceny algorytmów uczenia maszynowego (ang. Intersection over Union).
LTE	Standard transmisji danych wykorzystywany w sieciach komórkowych 4G (ang. Long Term Evolution).
MAPE	Miara służąca do oceny dokładności predykcji (ang. Mean Absolute Percentage Error).
MCTR	Obszar kontrolowany przez lotnisko wojskowe (ang. Military Control Zone) wymagający zgody na loty BSP wydawane przez Szefostwo Służby Ruchu Lotniczego lotniska.
MCU	Scalony układ mikroelektroniczny zaprojektowany do sterowania operacjami w systemach wbudowanych (ang. Micro-Controller Unit).

MSE	Błąd średniokwadratowy (ang. Mean Square Error).
MWD	Topograficzna miara wybitności szczytu (minimalna deniwelacja względna).
NetCDF4	Format danych (meteorologicznych lub oceanograficznych), w którym dane i ich opisy są zawarte w tym samym zbiorze (Network Common Data Form).
nginx	Serwer WWW (HTTP) oraz serwer proxy dla HTTP i IMAP/POP3 rozwijany i wspierany przez firmę, Nginx, Inc.
OTAA	Metoda aktywacji urządzenia końcowego w sieci bezprzewodowej, polegająca na dynamicznym przydzielaniu unikalnego adresu urządzenia (DevAddr) oraz negocjowaniu kluczy sesji (NwkSKey i AppSKey) pomiędzy urządzeniem a serwerem sieciowym (ang. Over-The-Air Activation).
PaaS	Warstwa usług chmurowych dostarczająca gotowe środowisko do tworzenia, testowania i wdrażania aplikacji, bez konieczności zarządzania infrastrukturą, taką jak serwery czy systemy operacyjne (ang. Platform as a Service).
PAŻP	Polska Agencja Żeglugi Powietrznej.
RSSI	Wskaźnik siły odbieranego sygnału radiowego (ang. Received Signal Strength Indicator).
RTH	Funkcja automatycznego powrotu BSP do miejsca startu (ang. Return to Home).
SaaS	Warstwa usług chmurowych dostarczająca oprogramowania przez Internet, gdzie aplikacje działają w chmurze i są dostępne na zasadzie subskrypcji (ang. Software as a Service).
SNAP	Oprogramowanie wytworzone na zlecenie Europejskiej Agencji Kosmicznej, przeznaczone do przetwarzania danych pochodzących z misji Sentinel-1, -2 i -3 programu Copernicus (ang. Sentinel Application Platform).
SORA	Ujednolicona, 10-etapowa ocena ryzyka operacji BSP w celu zapewnienia bezpieczeństwa i uzyskania zezwolenia od Urzędu Lotnictwa Cywilnego (ang. Specific Operations Risk Assessment).
SPI	Magistrala szeregową stosowana w systemach wykorzystujących MCU, pozwalająca na stosowanie synchronicznej, dwukierunkowej transmisji danych (ang. Serial Peripheral Interfejs).
STAC	Standard opisu i wyszukiwania zbiorów danych przestrzennych (ang. geospatial data) w formacie JSON (ang. Spatio-Temporal Asset Catalog).
ULC	Urząd Lotnictwa Cywilnego.
VLOS	Loty wykonywane w zasięgu wzroku pilota, utrzymującego przez cały czas bezpośredni kontakt wzrokowy z BSP (ang. Visual Line of Sight).
VTOL	Technologia pozwalająca statkom powietrznym na pionowy start i lądowanie bez potrzeby używania pasów startowych (ang. Vertical Take-Off and Landing).