

Składowanie danych w systemach Big Data

Raport z projektu

Kacper Skonieczka, 313505
Grzegorz Zakrzewski, 313555

10 stycznia 2024

Spis treści

1	Cel projektu	1
2	Zbiory danych	2
2.1	Źródło - ENTSO-E Transparency Platform	2
2.2	Źródło - Global Ensemble Forecast System	3
3	Architektura rozwiązania	6
4	Pozyskiwanie, przetwarzanie i składowanie danych	7
4.1	Przepływ danych ze źródła ENTSO-E Transparency Platform	7
4.2	Przepływ danych ze źródła GEFS	11
5	Warstwa analityczna rozwiązania	13
5.1	Analiza porównawcza cen energii i produkcji energii wiatrowej	14
5.2	Analiza wpływu prędkości wiatru na produkcję i ceny energii	15
6	Pozostałe informacje o projekcie	18
7	Podsumowanie	18
8	Podział pracy	19

1 Cel projektu

Celem naszego projektu jest opracowanie zaawansowanego systemu do gromadzenia, przetwarzania, przechowywania i analizowania danych meteorologicznych oraz informacji o produkcji energii wiatrowej w krajach Europy. Wykorzystamy do tego technologie Big Data z rodziny Apache, co gwarantuje efektywność i skalowalność rozwiązania.

W ramach projektu skupiamy na wybranym kraju - na Niemczech. Nasz system jest zaprojektowany w taki sposób, aby łatwo można było rozszerzyć zakres danych o dodatkowe państwa, co zwiększa jego uniwersalność i przyszłościowy potencjał.

Kluczowym aspektem naszego projektu jest integracja danych meteorologicznych z informacjami o cenach energii na giełdzie i produkcji energii z farm wiatrowych. Dzięki temu będziemy mogli przeprowadzać głębokie analizy oraz wnioskować o wpływie warunków pogodowych i produkcji energii wiatrowej na rynek energetyczny. To podejście pozwala nam na uzyskanie cennych wniosków i wspieranie decyzji biznesowych w sektorze energetycznym.

2 Zbiory danych

2.1 Źródło - ENTSO-E Transparency Platform

Opis źródła ENTSO-E Transparency Platform, stworzona przez Europejską Sieć Operatorów Systemów Przesyłowych Energii Elektrycznej, odgrywa kluczową rolę w zwiększaniu przejrzystości rynku energii w Europie. Platforma ta oferuje szeroki zakres danych, od produkcji energii z różnorodnych źródeł po niezbilansowanie sieci energetycznych i prognozy konsumpcji, obejmujących niemal wszystkie kraje europejskie.

W ramach naszego projektu koncentrujemy się na danych dotyczących produkcji energii z farm wiatrowych. Analizujemy ich wpływ na rynek energii, uwzględniając zmienność produkcji w zależności od warunków pogodowych. Istotnym elementem analizy są także dane o cenach energii, dostarczane przez ENTSO-E, które są kluczowe dla zrozumienia dynamiki rynku i przewidywania trendów cenowych.

Dostęp do danych z platformy ENTSO-E uzyskuje się poprzez API, które wymaga autoryzacji. Dostęp do API jest możliwy po uzyskaniu tokena zabezpieczającego, który otrzymuje się po bezpłatnej rejestracji w serwisie. Autoryzacja odbywa się poprzez podanie tokenu zabezpieczającego jako parametru zapytania:

`https://web-api.tp.entsoe.eu/api?securityToken=MYTOKEN`

Dokumentacja RESTful API udostępniona przez ENTSO-E zawiera szczegółowe informacje na temat atrybutów stosowanych w zapytaniach. Określa ona, które parametry są obowiązkowe lub opcjonalne dla poszczególnych rodzajów danych. Szczegóły dotyczące atrybutów można znaleźć w tym przewodniku [5].

Do określenia żadanego przedziału czasowego stosujemy parametry `periodStart` oraz `periodEnd`, pozwalające na określenie granic czasowych w formacie `yyyyMMddHHmm`, na przykład: `202401010000`.

W ramach naszego projektu skorzystamy z następujących endpoint-ów ENTSO-E Transparency Platform:

- zapytanie pozwalające pobrać ceny energii dla giełdy energetycznej w Niemczech:

```
GET /api?documentType=A44
    &in_Domain=10Y1001A1001A82H
    &out_Domain=10Y1001A1001A82H
    &periodStart=202312010000
    &periodEnd=202401082300
```

- zapytanie pozwalające pobrać wartości wyprodukowanej energii z wiatru na morzu w MW w Niemczech:

```
GET /api?documentType=A73
    &processType=A16
    &psrType=B18
    &in_Domain=10Y1001A1001A82H
    &periodStart=202312010000
    &periodEnd=202401082300
```

Zapytania będą wykonywane raz dziennie. Odpowiedź API na zapytanie o produkcję energii wiatrowej w Niemczech zawiera szczegółowe dane o każdej godzinie produkcji w określonym przedziale czasowym. Informacje zawarte w sekcji szeregów czasowych obejmują:

- przedział czasowy produkcji, na przykład: `2023-12-01T00:00Z` do `2024-01-01T00:00Z`;
- ilość wyprodukowanej energii wiatrowej na każdy kwadrans danego dnia.

Odpowiedź API na zapytanie o ceny energii jest analogiczna. Obejmuje ona:

```

▼<Publication_MarketDocument xmlns="urn:iec62325.351:tc57wg16:451-3:publicationdocument:7:0">
  <mRID>0e1dfb878dfd470b840f6e9ae0f45b91</mRID>
  <revisionNumber>1</revisionNumber>
  <type>A44</type>
  <sender_MarketParticipant.mRID codingScheme="A01">10X1001A1001A450</sender_MarketParticipant.mRID>
  <sender_MarketParticipant.marketRole.type>A32</sender_MarketParticipant.marketRole.type>
  <receiver_MarketParticipant.mRID codingScheme="A01">10X1001A1001A450</receiver_MarketParticipant.mRID>
  <receiver_MarketParticipant.marketRole.type>A33</receiver_MarketParticipant.marketRole.type>
  <createdDateTime>2023-11-25T17:34:10Z</createdDateTime>
  ▼<period.timeInterval>
    <start>2022-12-31T23:00Z</start>
    <end>2023-01-01T23:00Z</end>
  </period.timeInterval>
  ▼<TimeSeries>
    <mRID>1</mRID>
    <businessType>A62</businessType>
    <in_Domain.mRID codingScheme="A01">10YFR-RTE-----C</in_Domain.mRID>
    <out_Domain.mRID codingScheme="A01">10YFR-RTE-----C</out_Domain.mRID>
    <currency_Unit.name>EUR</currency_Unit.name>
    <price_Measure_Unit.name>MWh</price_Measure_Unit.name>
    <curveType>A01</curveType>
    ▼<Period>
      ▼<timeInterval>
        <start>2022-12-31T23:00Z</start>
        <end>2023-01-01T23:00Z</end>
      </timeInterval>
      <resolution>PT60M</resolution>
      ▼<Point>
        <position>1</position>
        <price.amount>0.00</price.amount>
      </Point>
      ▼<Point>
        <position>2</position>
        <price.amount>-0.10</price.amount>
      </Point>
    </Period>
  </TimeSeries>
</Publication_MarketDocument>

```

Rysunek 1: Przykładowa odpowiedź XML na zapytanie o ceny energii w Niemczech.

- przedział czasowy, na przykład: 2023-12-01T00:00Z do 2024-01-01T00:00Z;
- ceny energii elektrycznej na każdą godzinę w tym okresie.

Przykładowa odpowiedź XML na zapytanie o ceny energii w Niemczech znajduje się na Rysunku 1.

Podsumowanie

- **rodzaj danych:** ceny energii i produkcja energii z elektrowni wiatrowych w Niemczech;
- **liczba rekordów:** dwie serie po 24 (ceny) i 96 rekordy (energia wiatrowa) na każdy dzień w zakresie od 01.12.2023 r. - 08.01.2024 r.;
- **częstotliwość napływu:** raz dziennie w przypadku cen (12:45) i co godzinę w przypadku energii wiatrowej;
- **format:** dokument XML.

2.2 Źródło - Global Ensemble Forecast System

Opis źródła Global Ensemble Forecast System (GEFS) to model prognozy pogody składający się z 21 oddzielnych prognoz (*ensemble members*). National Centers for Environmental Prediction (NCEP) uruchomiły model GEFS, aby zaadresować naturę “niepewności” w obserwacjach pogody. Model GEFS próbuje określić niepewność prognozy, generując zestaw wielu prognoz, z których każda jest minimalnie inna lub zaburzona w stosunku do pierwotnych obserwacji.

Model GEFS ma zasięg globalny, nowe prognozy są publikowane cztery razy dziennie (co sześć godzin zaczynając od północy). Sama prognoza pogody obejmuje okres 16 dni od momentu publikacji, w grupach co trzy godziny przez pierwsze 10 dni i co sześć godzin w dniach od 10 do 16. Oznacza to, że jedna publikacja zawiera predykcje wartości kilkudziesięciu zmiennych (parametrów meteorologicznych) osobno dla każdej z 21 wiązek, dla każdej szerokości

i długości geograficznej z dokładnością do pół stopnia, dla każdego punktu w przyszłości oddalonego od daty publikacji o 0, 3, 6, ..., 237, 240, 246, ..., 378, 384 godziny.

Prognozy modelu GEFS są publikowane jako zestaw plików w formacie GRIB2. Format plików GRIB2 to popularny format przechowywania danych meteorologicznych. Otwarcie takiego pliku wymaga użycia dedykowanych bibliotek. Dla języka programowania Python, takie biblioteki to paczka *xarray* [7] z silnikiem *cfgrib* [3].

Prognozy modelu GEFS są przechowywane i publicznie dostępne przez Amazon S3 bucket [4]. Korzystanie z przeglądarkowego eksploratora [1] ułatwi zapoznanie się z nazewnictwem oraz hierarchią plików. Adres przykładowego pliku z prognozami modelu GEFS prezentuje się następująco:

<https://noaa-gefs-pds.s3.amazonaws.com/gefs.20231201/00/atmos/pgrb2ap5/geavg.t00z.pgrb2a.0p50.f000>

Z pewną dozą cierpliwości można rozszyfrować ten adres:

- <https://noaa-gefs-pds.s3.amazonaws.com/gefs>
bazowy adres, pod którym znajduje się Amazon S3 bucket;
- [.20231201/](#)
data w formacie YYYYMMDD;
- [/00/](#)
oznacza godzinę publikacji; w ciągu jednego dnia mają miejsce cztery publikacje: 00, 06, 12 i 18;
- [/atmos/](#)
oznacza odczyty atmosferyczne;
- [/pgrb2ap5/](#)
oznacza grupę **a**; grupa **a** zawiera kilkadziesiąt najczęściej wykorzystywanych zmiennych, między innymi zmienne określające interesującą nas prędkość wiatru; pełen spis zmiennych można znaleźć w tym miejscu [6]; istnieje również grupa **b** kilkuset “mniej popularnych” zmiennych;
- [/geavg.](#)
to numer wiązki; istnieje 21 zwykłych wiązek od **gep01** do **gep21**, wiązka kontrolna **gec00** (prognoza modelu bez zaburzeń) oraz wiązka **geavg** będąca średnią wszystkich wiązek;
- [.t00z.](#)
ponownie godzina publikacji;
- [.pgrb2ap5.](#)
ponownie grupa zmiennych;
- [.0p50.](#)
nieistotny element adresu;
- [.f000](#)
punkt w przyszłości, do którego odnoszą się predykcje, od **f000** do **f384** (16 dni) co 3 godziny lub co 6 godzin.













Na każdą wiązkę wypada 105 plików GRIB2 z prognozami. Suma 23 wiązek daje 2415 plików na jedną publikację, czyli 9660 plików dziennie. Każdy plik waży około 10 MB, co daje już znaczny wolumen 100 GB surowych danych dziennie. W ramach tego projektu z powodów sprzętowych niemożliwe jest przetwarzanie takiej ilości danych. Ograniczamy się tylko do wiązki uśrednionej **geavg** publikowanej codziennie o północy, do predykcji “na tydzień wprzód” **f168** dla wycinka współrzędnych geograficznych obejmujących wybrane państwo - Niemcy. Cały system jest zaprojektowany tak, żeby można było rozszerzyć zakres danych.

Na Rysunku 2 znajduje się przykładowy plik z prognozami modelu GEFS otwarty za pomocą biblioteki *xarray*, a w Tabeli 1 znajduje się kilka pierwszych wierszy z tabeli utworzonej na podstawie tego samego pliku. Ten przykład przedstawia tylko dwie interesujące nas zmienne, to jest składowe *u* i *v* prędkości wiatru zmierzone na wysokości 10 metrów nad powierzchnią ziemi.





xarray.Dataset

► **Dimensions:** (latitude: 361, longitude: 720)

▼ **Coordinates:**

time	()	datetime64[ns]	2023-11-25	 
step	()	timedelta64[ns]	7 days	 
heightAboveGro...	()	float64	10.0	 
latitude	(latitude)	float64	90.0 89.5 89.0 ... -89.5 -90.0	 
longitude	(longitude)	float64	0.0 0.5 1.0 ... 358.5 359.0 359.5	 
valid_time	()	datetime64[ns]	2023-12-02	 

▼ **Data variables:**

u10	(latitude, longitude)	float32	...	 
v10	(latitude, longitude)	float32	...	 

Rysunek 2: Przykładowy plik z prognozami modelu GEFS otwarty za pomocą biblioteki *xarray*.

latitude	longitude	time	step	valid_time	u10	v10
45.0	5.0	2023-12-01	7 days	2023-12-08	2.43	0.66
45.0	5.5	2023-12-01	7 days	2023-12-08	2.44	0.64
45.0	6.0	2023-12-01	7 days	2023-12-08	2.44	0.62
45.0	6.5	2023-12-01	7 days	2023-12-08	2.45	0.59
45.0	7.0	2023-12-01	7 days	2023-12-08	2.45	0.57

Tabela 1: Przykładowy plik z prognozami modelu GEFS przekształcony do postaci tabelarycznej.

Podsumowanie

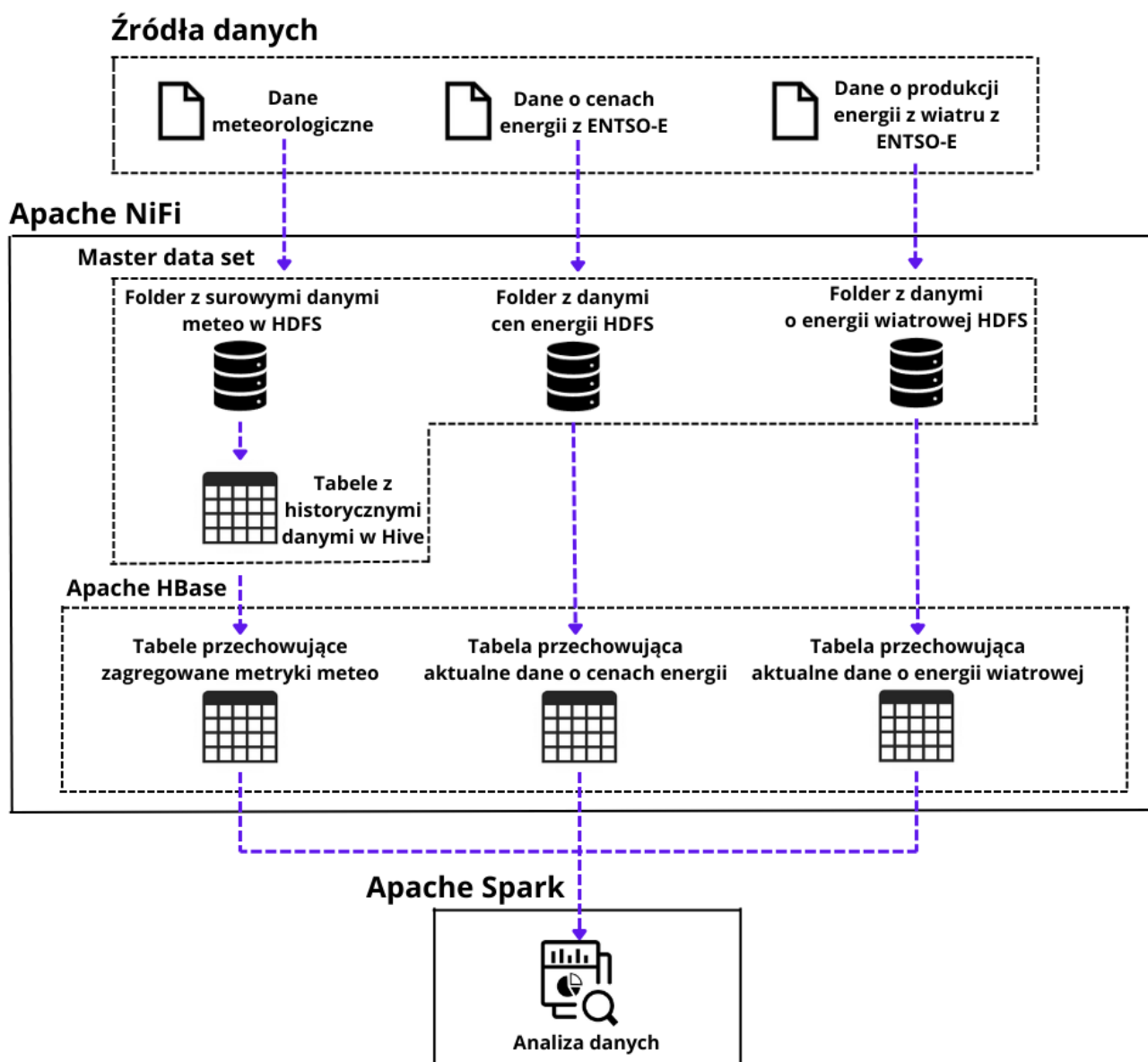
- **rodzaj danych:** dane meteorologiczne obejmujące tygodniowe prognozy składowych *u* i *v* prędkości wiatru, dla wycinka współrzędnych geograficznych (szerokości od 45 do 55, długości od 5 do 15 co 0.5 stopnia) obejmującego obszar Niemiec;
- **liczba rekordów:** 441 rekordy na każdy dzień w zakresie od 01.12.2023 r. - 08.01.2024 r.;
- **częstotliwość napływu:** raz dziennie;
- **format:** plik GRIB2.

3 Architektura rozwiązania

Projekt został zrealizowany w oparciu o zespół rozwiązań technologicznych z rodziny Apache obejmujący:

- Apache NiFi - automatyzacja przepływu danych
- Apache HDFS - składowanie danych surowych
- Apache Hive - składowanie przetworzonych danych meteorologicznych
- Apache HBase - składowanie szeregów czasowych i zagregowanych danych meteorologicznych
- Apache Spark - wsadowa analiza danych

Diagram architektury rozwiązania przedstawiony jest na Rysunku 3.



Rysunek 3: Diagram architektury rozwiązania.

4 Pozyskiwanie, przetwarzanie i składowanie danych

W tej sekcji opisany jest sposób pozyskiwania, przetwarzania i składowania danych źródłowych. Przepływ danych jest nadzorowany przez Apache Nifi. Wybór tego narzędzia był podyktowany tym, że spodziewany wolumen danych do przetworzenia pomiędzy węzłami jest relatywnie niewielki i nie zakładano akumulowania pomiędzy węzłami istotnych zasobów informacji.

Opracowane przepływy danych obejmują wszystkie etapy klasycznego podejścia ETL (*Extract, Transform, Load*). Etap transformacji odnosi się głównie do konwersji formatu danych bez modyfikacji ich pierwotnej treści.

4.1 Przepływ danych ze źródła ENTSO-E Transparency Platform

Projekt skupia się na danych od początku grudnia 2023 roku. Liczba rekordów odpowiada liczbie godzin od tego momentu aż do obecnego dnia, generując w sumie około 1000 rekordów w postaci szeregu czasowego. Dane zawierają informacje o cenie energii elektrycznej (EUR/MWh) oraz o ilości energii produkowanej z elektrowni wiatrach (MW) na każde godzinę. Informacje o cenach na kolejne 24 godziny są udostępniane raz dziennie, natomiast dane o energii wiatrowej są aktualizowane co godzinę. Obie serie danych są pobierane z zapasem jednej godziny w tył, co pozwala na uzupełnienie ewentualnych braków i zabezpieczenie przed opóźnieniami w publikacji.

Przepływ danych jest analogiczny dla obu typów zapytań - zarówno dla cen energii, jak i dla wartości produkcji energii z wiatru. Opisujemy proces na przykładzie danych dotyczących wiatru. W grupie procesów realizujących przepływ danych wyróżniamy następujące etapy:

- wygenerowanie Flow File (abstrakcji właściwej dla projektu Apache NiFi) i pobranie pliku z EntsoeApi
- przetwarzanie pobranego pliku
- wyciąganie danych z pobranego pliku
- konwersja danych
- zapis danych

Etapy są oznaczone kolorami odpowiadającymi kolorom procesorów Apache NiFi, które odpowiadają za realizację tych etapów. Przepływy danych w Apache Nifi przedstawione są na Rysunkach 4 oraz 5.

Ekstrakcja danych odbywa się za pomocą procesora **InvokeHTTP** oznaczonego **GetWindXmlByHttp**. Należało ustanowić bezpieczne, szyfrowane połączenie między NiFi a serwerem API Entsoe. W tym celu pobraliśmy publiczny certyfikat SSL z serwera 'entsoe.eu'. Aby pobierać dane od 2023-12-01 do obecnego dnia ustaliliśmy parametr **RemoteURL** na

```
https://web-api.tp.entsoe.eu/api?  
securityToken=7032e795-c8ae-4a50-aac8-a377b64b1c9e  
&documentType=A69  
&processType=A01  
&psrType=B19  
&in_Domain=10Y1001A1001A82H  
&periodStart=202312010000&periodEnd=${now():format('yyyyMMdHH')}00
```

W procesorze **SetPublicationDate** każdemu pobranemu plikowi XML przypisywany jest atrybut **publication_date** reprezentujący aktualną datę i godzinę. Dzięki temu wszystkie segmenty pliku XML, które powstały w wyniku jego podziału, zachowują jednolitą datę publikacji.

W procesorze **SplitByPeriods** dokonujemy podziału pliku XML na sekcje odpowiadające poszczególnym okresom (*Periods*), co pozwala na dalszą segmentację danych według dni. Za pomocą procesora **RouteOnResolution-Content** selekcjonujemy dane o interesującej nas rozdzielczości: PT15M dla danych wiatrowych (odpowiadających

produkcji energii wiatrowej co 15 minut) oraz PT60M dla danych cenowych (odpowiadających cenie energii na każdą godzinę).

Procesor **GetStartAndEndDate** wykorzystuje wyciągane z XML danych zakresy dat (**start_date** i **end_date**) dla każdego punktu czasowego (*Point*) w obrębie okresów (*Periods*). Następnie, dzięki procesorom **SplitByPoints** i **RouteOnPoints**, pliki są dalej dzielone i filtrowane zgodnie z indywidualnymi punktami czasowymi (*Points*).

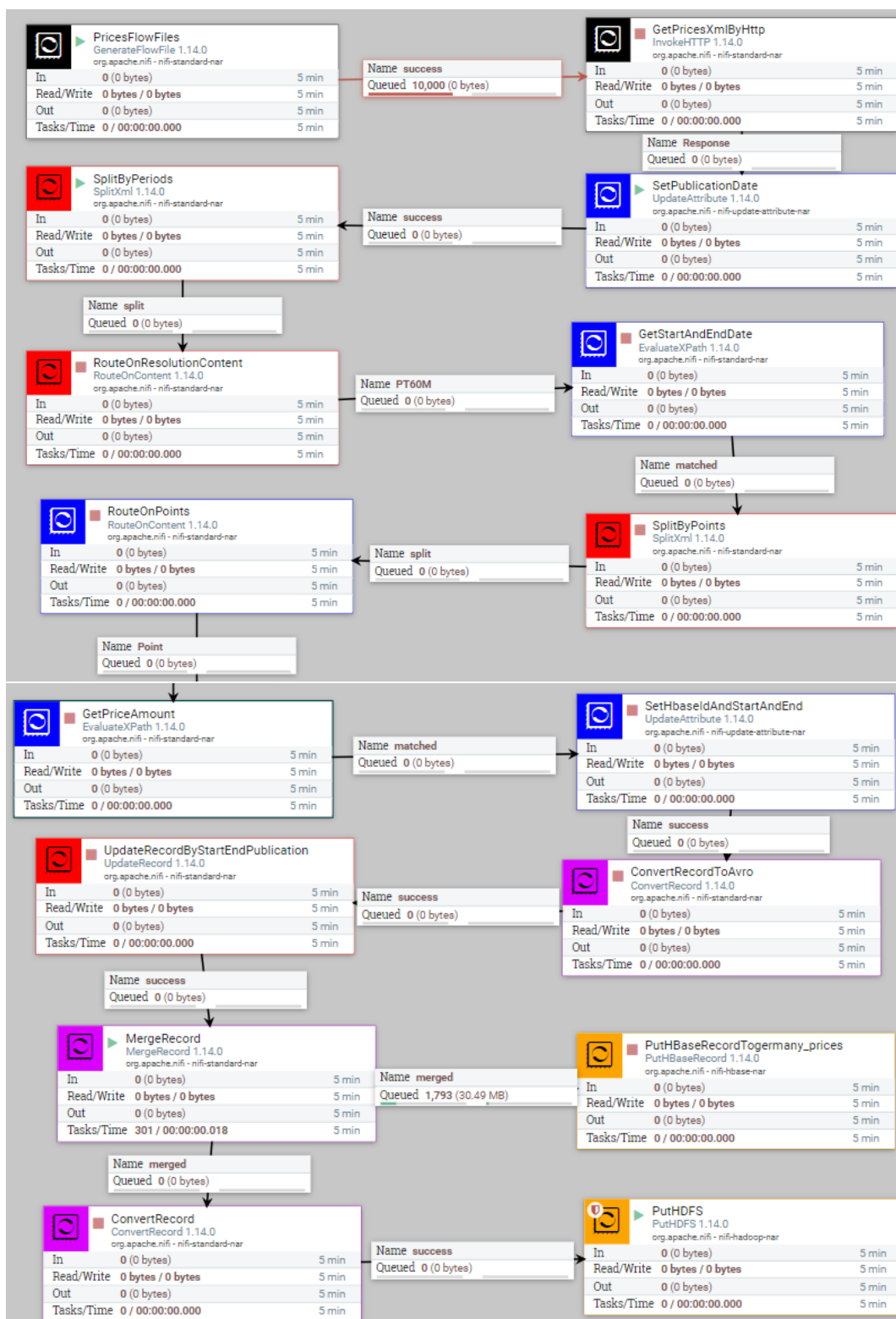
W procesorze **GetPositionQuantity** (w przypadku danych o cenach energii **GetPriceAmount**) wydobywamy informacje o konkretnej godzinie (*Position*), której dotyczą dane wartości. W procesorze **SetHbaseIdAndStartAndEnd** przydzielamy unikalny identyfikator ID dla każdego rekordu w postaci **position_end_date**.

Dane są konwertowane do formatu Avro w procesorze **ConvertRecordToAvro**, po czym wzbogacane o atrybuty **end_date**, **publication_date**, **row_id** oraz **start_date** w **UpdateRecordByStartEndPublication**. W kolejnym etapie, w procesorze **MergeRecord**, dokonujemy scalenia mniejszych segmentów danych w większe jednostki.

W ostatnim etapie, zapisujemy dane równolegle w dwóch miejscach. Za pomocą procesora **PutHbaseRecord** zapisujemy dane w bazie HBase do tabeli **windOffShore** (w przypadku danych o cenach energii **germany_prices**), gdzie każdy nowy rekord jest aktualizowany na podstawie wartości **row_id**, zachowując tylko najnowsze dane dla danego ID. Jednocześnie za pomocą procesora **PutHDFS** konwertujemy dane do formatu **parquet** i zapisujemy je w systemie plików HDFS pod ścieżką **/user/project/windOffshore/wind_{publication_date}_{UUID()}.parquet**. Takie podejście pozwala na przechowywanie wersji danych dla każdej pory pobrania w HDFS, natomiast w HBase utrzymywane są jedynie najnowsze i najbardziej kompletne dane.



Rysunek 4: Przepływ danych 15 minutowych z Entsoe Api dotyczących produkcji z farm wiatrowych



Rysunek 5: Przepływ danych 60 minutowych z Entsoe Api dotyczących produkcji cen energii w Niemczech

4.2 Przepływ danych ze źródła GEFS

Schemat przepływu Apache Nifi dotyczącego danych meteorologicznych pochodzących z modelu GEFS znajduje się na Rysunku 6. Obejmuje on trzy główne kroki: pobranie pliku GRIB2 z prognoząmi i zapisanie go w systemie HDFS, otwarcie i wstawienie wybranego zakresu danych do tabeli Apache Hive oraz wreszcie, agregację danych do tabeli HBase.

Szczegółowy opis przepływu danych:

1. Download GRIB2 File, Set File Name, Put GRIB2 File to HDFS

Procesor *InvokeHTTP* z ustawioną częstotliwością uruchamiania raz dziennie odpytuje odpowiedni adres i pobiera wybrany plik GRIB2 z prognoząmi. Wspomniany adres internetowy ma postać:

```
https://noaa-gefs-pds.s3.amazonaws.com/  
gefs.${now():format("yyyyMMdd")}/00/atmos/pgrb2ap5/geavg.t00z.pgrb2a.0p50.f168
```

Plik GRIB2 z prognoząmi otrzymuje nazwę adekwatną do danego dnia i jest on umieszczony w systemie plików HDFS w folderze `/user/project/gefs/raw`.

2. Convert GRIB2 to CSV

Apache NiFi nie posiada dedykowanego procesora do plików GRIB2. Dlatego musi nastąpić konwersja plików GRIB2 do bardziej odpowiedniego formatu. Dzieje się to za pośrednictwem skryptu napisanego w języku Python, który korzysta z dedykowanych bibliotek `xarray` i `cfgrib`. W tym kroku surowy plik GRIB2 jest pobierany z HDFS, a później następuje jego konwersja do postaci tabelarycznej do formatu CSV przy jednoczesnym wyborze zmiennych meteorologicznych i obszaru geograficznego. Utworzony plik CSV trafia z powrotem do HDFS.

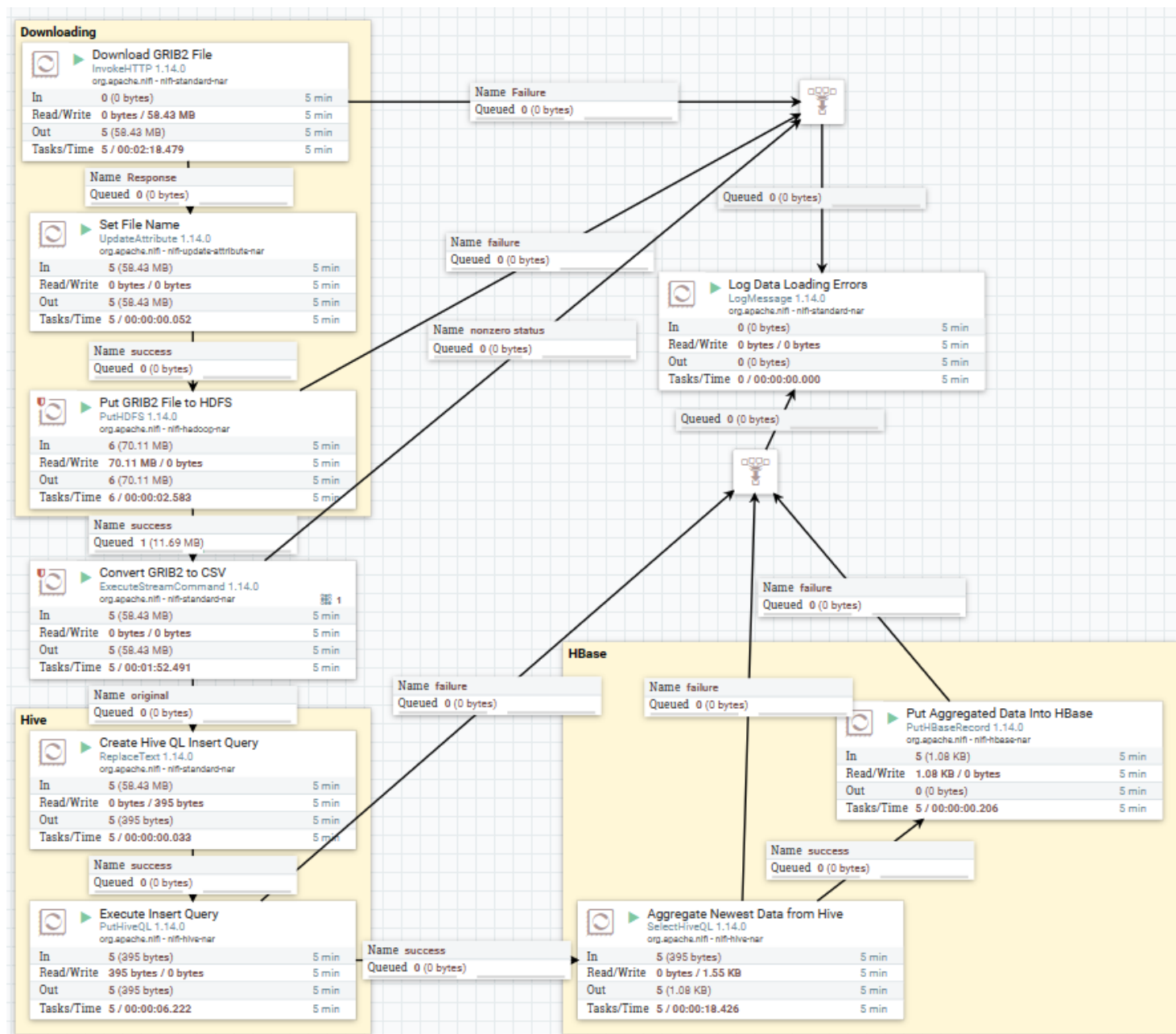
3. Create Hive QL Insert Query, Execute Insert Query

Utworzony w poprzednim kroku plik CSV jest wstawiany do tabeli `gefs` przygotowanej w Apache Hive. Dane meteorologiczne mają bardzo uporządkowaną postać, liczba wierszy i kolumn jest zawsze stała, więc tabela Apache Hive jest idealnym miejscem na ich przechowywanie. Wstawianie do tabeli odbywa się z użyciem dwóch procesorów - jeden z nich generuje polecenie wstawiające, a drugi procesor wykonuje to polecenie.

4. Aggregate Newest Data From Hive, Put Aggregated Data Into HBase

Na podstawie każdej codziennej procji danych wstawianej do Apache Hive obliczane są pewne statystyki. Obejmują one średnie wartości składowych `u` i `v` prędkości wiatru dla całego wybranego obszaru, a także dla północnych Niemiec (powyżej 50 równoleżnika) i dla południowych Niemiec (poniżej 50 równoleżnika). Powstałe w wyniku agregacji sześć zmiennych jest wstawianie do tabeli `gefs_agg` znajdującej się w Apache HBase. Pierwszy z powiązanych z tym etapem procesorów odpytuje tabelę Apache Hive tworząc wspomniane agregacje, a drugi procesor wstawia obliczone dane do tabeli Apache HBase.

Ponadto, na schemacie przepływu danych znajduje się jeszcze procesor `Log Data Loading Errors`. Jego zadaniem jest zbieranie wszystkich błędów, które mogą pojawić się w ramach przetwarzania danych.



Rysunek 6: Przepływ danych meteorologicznych ze źródła GEFS

5 Warstwa analityczna rozwiązania

W niniejszym projekcie do analizy danych są wykorzystane narzędzia Apache Spark oraz Jupyter Notebook. Narzędzie Apache Spark jest wykorzystane do przeprowadzenia obliczeń, agregacji danych oraz zebrania statystyk. Natomiast plik w formacie **.ipynb* pełni rolę interfejsu użytkownika aplikacji analitycznej. Ta warstwa będzie odpowiadać za finalne obliczenia oraz wizualizację danych. Z poziomu tak przygotowanej warstwy analitycznej możliwe jest wykonanie szeregu analiz, sięgając przy tym zarówno po dane umieszczone w systemie plików HDFS, jak i te zarządzane przez oprogramowania Apache Hive i Apache HBase.

Przykładowe dane dostępne z poziomu warstwy analitycznej

```
prices_da.show(n=10)
```

position	price_amount	start_date	row_id	end_date	publication_date
1	106.91	2023-11-30T23:00Z	1_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
3	99.1	2023-11-30T23:00Z	3_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
5	96.07	2023-11-30T23:00Z	5_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
19	199.16	2023-11-30T23:00Z	19_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
20	168.56	2023-11-30T23:00Z	20_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
7	120.03	2023-11-30T23:00Z	7_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
10	222.63	2023-11-30T23:00Z	10_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
13	197.09	2023-11-30T23:00Z	13_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
16	184.4	2023-11-30T23:00Z	16_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22
21	142.37	2023-11-30T23:00Z	21_2023-11-30T23:00Z	2023-12-01T23:00Z	2024-01-08 22

only showing top 10 rows

Rysunek 7: Dane o cenach energii w Niemczech załadowane za pomocą Apache Spark z HDFS

```
wind_data.show(n=10)
```

position	quantity	start_date	row_id	end_date	publication_date
42	4418.0	2023-12-11T23:00Z	42_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
1	17153.0	2023-12-11T23:00Z	1_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
43	4327.0	2023-12-11T23:00Z	43_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
2	16647.0	2023-12-11T23:00Z	2_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
44	4231.0	2023-12-11T23:00Z	44_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
3	16141.0	2023-12-11T23:00Z	3_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
4	15627.0	2023-12-11T23:00Z	4_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
45	4145.0	2023-12-11T23:00Z	45_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
5	15067.0	2023-12-11T23:00Z	5_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21
46	4107.0	2023-12-11T23:00Z	46_2023-12-12T23:00Z	2023-12-12T23:00Z	2024-01-08 21

only showing top 10 rows

Rysunek 8: Dane o produkcji energii wiatrowej w Niemczech załadowane za pomocą Apache Spark z HDFS

```
spark.sql("select * from gefs where latitude is not null;").show(n=10)
```

latitude	longitude	time	valid_time	u10	v10
45.0	5.0	2023-11-24 00:00:00	2023-12-01 00:00:00	0.25	1.71
45.0	5.5	2023-11-24 00:00:00	2023-12-01 00:00:00	-1.06	1.77
45.0	6.0	2023-11-24 00:00:00	2023-12-01 00:00:00	-1.0	2.34
45.0	6.5	2023-11-24 00:00:00	2023-12-01 00:00:00	0.37	2.29
45.0	7.0	2023-11-24 00:00:00	2023-12-01 00:00:00	0.53	0.37
45.0	7.5	2023-11-24 00:00:00	2023-12-01 00:00:00	0.26	-0.64
45.0	8.0	2023-11-24 00:00:00	2023-12-01 00:00:00	0.41	-0.7
45.0	8.5	2023-11-24 00:00:00	2023-12-01 00:00:00	0.26	-1.45
45.0	9.0	2023-11-24 00:00:00	2023-12-01 00:00:00	-0.61	-1.2
45.0	9.5	2023-11-24 00:00:00	2023-12-01 00:00:00	-1.2	-0.68

only showing top 10 rows

Rysunek 9: Dane meteorologiczne załadowane za pomocą Apache Spark z tabeli Apache Hive

```
connection = happybase.Connection("localhost")

table = connection.table('gefs_agg')
table.row(b"20231201")

{b'agg:u10_avg_north': b'-1.4940259740259734',
 b'agg:u10_avg_south': b'-0.22376190476190475',
 b'agg:u10_avg_total': b'-0.8891383219954643',
 b'agg:v10_avg_north': b'1.9530735930735934',
 b'agg:v10_avg_south': b'1.195142857142857',
 b'agg:v10_avg_total': b'1.5921541950113374'}
```

Rysunek 10: Przykładowy rekord załadowany z tabeli Apache HBase za pomocą biblioteki happybase

5.1 Analiza porównawcza cen energii i produkcji energii wiatrowej

Analiza trendów na rynku energii z wykorzystaniem danych historycznych pozwala na wyciąganie wniosków mogących wpłynąć na przyszłe decyzje inwestycyjne i regulacyjne.

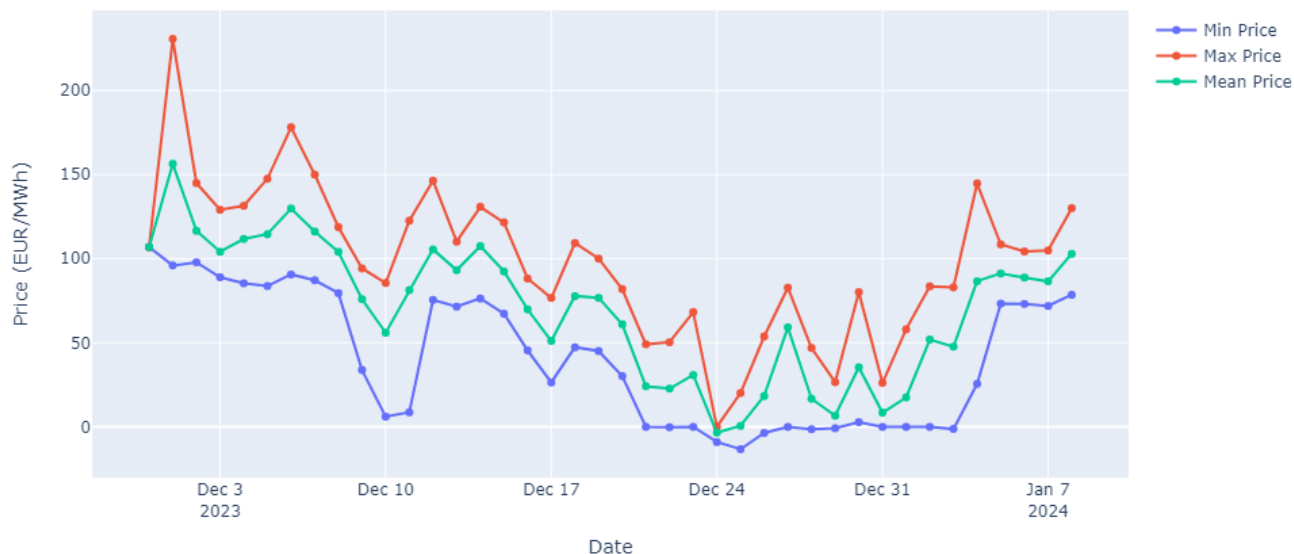
Analiza cen energii

Analizując wykres cen energii (Rysunek 11) można zauważyć, że rynek energii charakteryzuje się wysoką zmiennością cen. Wyraźne szczyty cenowe mogą odpowiadać okresom zwiększonego zapotrzebowania na energię, podczas gdy niskie ceny mogą być wynikiem niskiego zapotrzebowania lub nadprodukcji. Zmienność ta może być również wynikiem fluktuacji cen surowców energetycznych.

Porównanie cen energii do produkcji energii wiatrowej

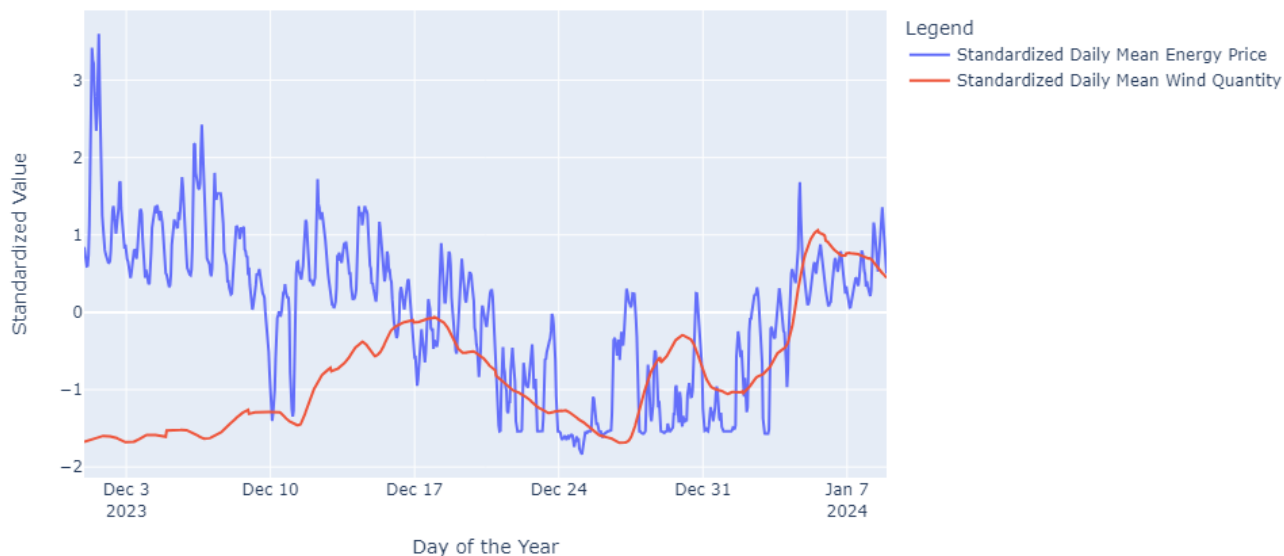
Porównując standaryzowane średnie dzienne wartości ceny energii i produkcji energii wiatrowej (Rysunek 12) można zauważyć momenty, w których wzrost produkcji energii wiatrowej koreluje z obniżką cen energii. Sugeruje to, że energetyka wiatrowa, będąca źródłem energii odnawialnej, może przyczyniać się do stabilizacji rynku energii i obniżania kosztów dla konsumentów. Jednakże nie zawsze obserwujemy bezpośrednią korelację, co wskazuje na wpływ innych czynników takich jak polityka energetyczna, stan infrastruktury energetycznej czy zmiany w cenach innych źródeł energii.

Min and Max Energy Price Per Day Over Time



Rysunek 11: Minimalne, maksymalne i średnie dzienne ceny energii na przestrzeni czasu.

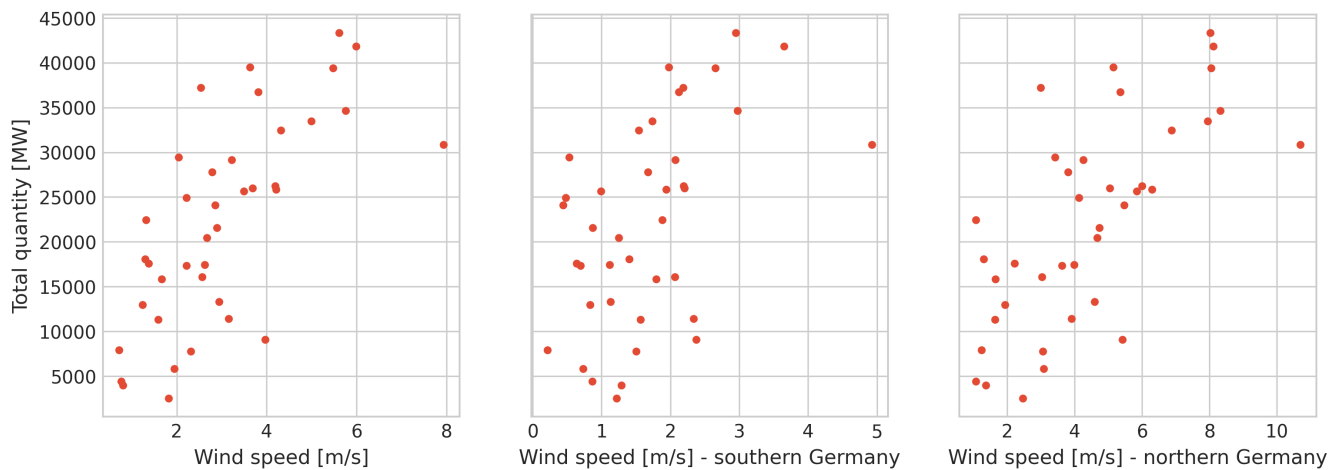
Standardized Daily Mean of Energy Price(EUR/MWh) and Wind Energy(MW)



Rysunek 12: Standaryzowane średnie dzienne wartości ceny energii i produkcji energii wiatrowej.

5.2 Analiza wpływu prędkości wiatru na produkcję i ceny energii

Na Rysunku 13 przedstawione są wykresy punktowe obrazujące zależności między sumą produkcji energii z wiatru na każdy dzień a średnią dzienną prędkością wiatru policzoną w całym badanym obszarze, a także na południu i na północy. Liczba obserwacji nie jest zbyt duża, jednak bez problemu można dostrzec pewne liniowe zależności.



Rysunek 13: Zależność produkcji energii wiatrowej od prędkości wiatru

	Wind speed	Wind speed - north	Wind speed - south	Mean price	Total quantity
Wind speed	1.00	0.97	0.81	-0.71	0.71
Wind speed - north	0.97	1.00	0.67	-0.70	0.70
Wind speed - south	0.81	0.67	1.00	-0.55	0.52
Mean price	-0.71	-0.70	-0.55	1.00	-0.88
Total quantity	0.71	0.70	0.52	-0.88	1.00

Rysunek 14: Macierz korelacji wszystkich zmiennych

Bardzo mocno potwierdza te obserwacje macierz korelacji pokazana na Rysunku 14. Średnie prędkości wiatru: całkowita, na północy i na południu są bardzo skorelowane między sobą, w szczególności prędkość całkowita i na północy. Prędkości wiatru są mocno dodatnio skorelowane z produkcją energii w elektrowniach wiatrowych, co jest zgodne z intuicją. Z kolei wszystkie wymienione zmienne są ujemnie skorelowane z ceną energii. Wynika to najprawdopodobniej z faktu, że energii wyprodukowana za pomocą odnawialne źródeł nie jest droga. Im więcej jest takiej energii, tym ogólna cena energii jest niższa.

Wnioski

Wnioski z przeprowadzonej analizy wskazują na złożoność rynku energii, na którym ceny są determinowane przez wiele współzależnych czynników. Dynamika produkcji energii wiatrowej wraz z innymi źródłami odnawialnymi może mieć stabilizujący wpływ na ceny energii. Rozwój technologii i zmiany w polityce energetycznej mogą dodatkowo wzmacniać ten trend, co w przyszłości może skutkować większą przewidywalnością i niższymi kosztami energii dla odbiorców końcowych.

Widoki wsadowe

Z wykorzystaniem zebranych danych, poza wcześniej przeprowadzonymi analizami, został przygotowany i zapisany jeden dodatkowy widok wsadowy. Agreguje on wszystkie dane do częstotliwości dziennej. Zawiera średnią cenę, sumę produkcji energii wiatrowej oraz średnią prędkości wiatru. Przykładowy wynik z widoku wsadowego przedstawiony jest na Rysunku 15.


```
spark.sql("""
    SELECT * FROM daily_data LIMIT 10
""").show()
```

[Stage 178:===== > (193 + 1) / 200]

date	mean_price	total_quantity	avg_wind_speed
2023-12-01 00:00:00	156.32583332061768	2539.8958333333335	2.1865103331468863
2023-12-02 00:00:00	116.53333282470703	4008.9791666666665	1.9503458452958466
2023-12-03 00:00:00	104.11666679382324	7771.7708333333333	3.0556860460029345
2023-12-04 00:00:00	111.63249969482422	17367.78125	2.9495622301300233
2023-12-05 00:00:00	114.56416670481364	17575.541666666668	1.9472222612239896
2023-12-06 00:00:00	129.7166665395101	7897.40625	1.2818300350788112
2023-12-07 00:00:00	116.09749984741211	5838.1145833333333	2.466828797988474
2023-12-08 00:00:00	104.00916703542073	13336.5	3.379572182383643
2023-12-09 00:00:00	75.85999981562297	21575.40625	3.3137457383100806
2023-12-10 00:00:00	55.90374974409739	29432.239583333332	2.643277738247381

Rysunek 15: Widok wsadowy agregujący dane to częstotliwości dziennej.

6 Pozostałe informacje o projekcie

Kod źródłowy, skrypty pozyskujące i transformujące dane, dedykowane pliki konfiguracyjne oraz wszystkie inne pliki wchodzące w skład projektu są umieszczone w repozytorium za pośrednictwem serwisu GitHub [2]. Opisując zawartość poszczególnych katalogów:

- `environment` - skrypt odpowiadający za instalację języka Python i paczek potrzebnych do otwarcia plików GRIB2;
- `hadoop` - polecenia tworzące strukturę katalogów w HDFS;
- `hbase` - polecenia tworzące tabele Apache HBase;
- `hive` - polecenia tworzące tabele Apache Hive;
- `nifi_helper_scripts` - skrypt w języku Python konwertujący pliki GRIB2 do formatu CSV i odpowiadający plik `.sh`;
- `nifi_templates` - szablony przetwarzania w Apache Nifi;
- `spark` - kod aplikacji Apache Spark, to jest warstwa analityczna w formacie pliku `.ipynb`;

Rozwiązanie zostało zaimplementowane w lokalnym środowisku, z wykorzystaniem obrazu maszyny wirtualnej udostępnionej studentom na zajęciach.

Testy funkcjonalne weryfikujące działanie projektu są umieszczone w osobnym pliku.

7 Podsumowanie

Udało się zrealizować główny cel projektu, to jest przygotowanie systemu do gromadzenia, przetwarzania, przechowywania i analizowania danych meteorologicznych, cen energii oraz informacji o produkcji energii wiatrowej. Znaleźliśmy i zdobyliśmy różnorodne dane powiązane z giełdą energetyczną. Przygotowaliśmy zaawansowany system Big Data. Wykorzystane technologie z rodziny Apache gwarantują efektywność i skalowalność rozwiązania.

Ponadto, przeprowadziliśmy ciekawą analizę i wyciągnęliśmy z niej wartościowe wnioski. Przedstawiliśmy, jak kształtują się ceny energii w Niemczech i jak są one powiązane z wysokością produkcji energii w elektrowniach wiatrowych. Wykonując obliczenia i wykresy potwierdziliśmy również intuicyjne założenie, że produkcja energii wiatrowej jest wprost proporcjonalna do prędkości wiatru, a odwrotnie proporcjonalna do cen energii.

Na koniec warto napisać, w jaki sposób można by poprawić lub rozwinąć przygotowany projekt. Na pewno warto byłoby rozszerzyć system na dane z całej Europy, obejmując również dłuższą historię. Pozwoliłoby to zaadresować bardziej zaawansowane problemy biznesowe.

8 Podział pracy

Podział pracy w zespole przedstawiony jest poniżej.

Członek zespołu	Zakres zadań
Kacper Skonieczka	Idea projektu Opis celu projektu i uzasadnienie biznesowe Opis i uzasadnienie architektury systemu Pozyskanie, obsługa przepływu i składowanie danych z ENTSO-E Podsekcje raportu związane z danymi pochodzącymi z ENTSO-E Przeprowadzenie testów funkcjonalnych
Grzegorz Zakrzewski	Idea projektu Diagram architektury systemu Opis i uzasadnienie architektury systemu Pozyskanie, obsługa przepływu i składowanie danych z modelu GEFS Agregacja danych pochodzących z modelu GEFS Podsekcje raportu związane z danymi pochodzącymi z modelu GEFS Przeprowadzenie testów funkcjonalnych

Tabela 2: Podział pracy w zespole

Literatura

- [1] AWS explorer - GEFS. <https://noaa-gefs-pds.s3.amazonaws.com/index.html>.
- [2] big-data-project github. <https://github.com/zakrzewow/big-data-project>.
- [3] cfgrib. <https://github.com/ecmwf/cfgrib>.
- [4] Dane GEFS w *Registry of Open Data on AWS*. <https://registry.opendata.aws/noaa-gefs/>.
- [5] ENTSO-E RESTful API implementation guide. https://transparency.entsoe.eu/content/static_content/Static_content/web_api/Guide.html.
- [6] Opis zmiennych GEFS. <https://www.nco.ncep.noaa.gov/pmb/products/gens/gec00.t12z.pgrb2af06.shtml>.
- [7] xarray. <https://docs.xarray.dev/en/stable/>.