

Konrad Ryczek
Michał Sacharczuk

Przestrzenne bazy danych

Temat

Dla danych dotyczących przejazdów komunikacji miejskiej dokonać analizy opóźnień i przyspieszeń, w szczególności zidentyfikować odcinki, na których najczęściej dochodzi do opóźnień lub przyspieszeń i okres kiedy te sytuacje mają miejsce. Dokonać wizualizacji wyników na mapie.

Opis rozwiązania

Aplikacja dla każdego przystanku określa opóźnienie powstałe podczas na odcinku (wyznacza różnicę między opóźnieniami na obecnym i poprzednim przystankiem)
Na podstawie tych danych wyznaczane są odcinki na których dochodzi do opóźnień.

Stos technologiczny

Backend

Został zaimplementowany z wykorzystaniem Spring Framework w języku Java. Serwer wystawia API REST-owe pozwalające na pobranie informacji o opóźnieniach na danej linii komunikacyjnej w określonym przedziale czasowym.

Aplikacja serwera składa się z modułów: kontrolera, serwisu aplikacyjnego oraz warstwy dostępu do danych. Kontroler jest odpowiedzialny za komunikację z aplikacją klienta poprzez zapytania REST-owe.

Frontend:

Aplikacja kliencka została zaimplementowana przy użyciu frameworka AngularJS dla języka JavaScript, oraz AngularJS Material jako framework komponentów interfejsu użytkownika. Do wizualizacji opóźnień został wykorzystany Google Maps oraz warstwa Heatmap, która umożliwia przedstawianie punktów na mapie, w określonej intensywności.

Model danych

Model danych zastosowany w aplikacji składa się z obiektów:

- DelayQueryResult służący do przechowywania wyników zapytania pobranych z bazy. Wyniki te poddawane są dalszej obróbce mającej na celu ich normalizację i umożliwiającą ich efektywną prezentację na warstwie prezentacji.
- DelayDTO - klasa reprezentująca obiekty służące do przesyłania wyników poszukiwania opóźnień do warstwy prezentacji.

```
public class DelaysQueryResult {  
    private double stopLatitude;  
    private double stopLongitude;  
    private double sectionDelay;  
}  
  
public class DelaysDTO {  
    private String x;  
    private String y;  
    private Double delay;  
}
```

Opis algorytmu wyznaczania przyspieszeń opóźnień

Proces wyznaczania opóźnień odbywa się w zapytaniu do bazy danych. Zapytanie wykorzystuje funkcję analityczną LAG. Pozwala ona na dostęp do innych rekordów pochodzących z tej samej tabeli w tym samym czasie unikając złączenia tabeli samej ze sobą (ang. Self join). Funkcja szereguje wartości za pomocą klauzuli order by a następnie pobiera wartość bezpośrednio poprzedzającą ją w szeregu.

```
daystop.delaysec - coalesce(lag(delaysec)  
                            OVER (  
                                PARTITION BY daystop.daycourse_lolid  
                                ORDER BY daystop.scheduleddeparture  
                            ), 0) delay_on_section,
```

Powyższy fragment kodu dla każdego przystanku pobiera przystanek go poprzedzający i odejmuje opóźnienie na poprzednim przystanku od opóźnienia na obecnym. W wyniku tej operacji otrzymujemy całkowite opóźnienie na odcinku łączącym te dwa przystanki.

Zapytanie posiada następujące parametry:

- Numer linii

- Godzina początkowa - wszystkie kursy które odbyły się przed tą godziną są odrzucane.
- Godzina końcowa - wszystkie kursy które odbyły się po tej godzinie są odrzucane.

W wyniku powyższego zapytania otrzymujemy listę kursów wraz z opóźnieniami na konkretnych odcinkach.

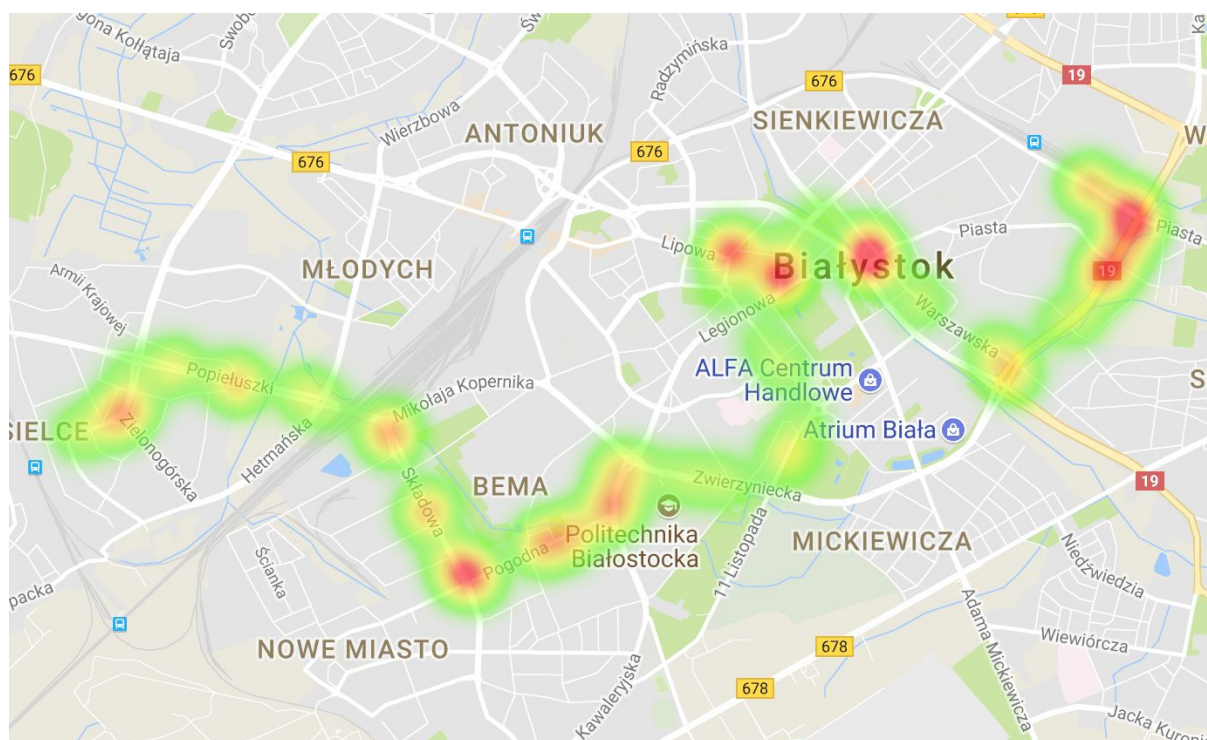
Dane te są grupowane po przystanku i przy użyciu funkcji AVG liczone jest średnie opóźnienie na każdym odcinku dla danej linii.

Wyniki testów

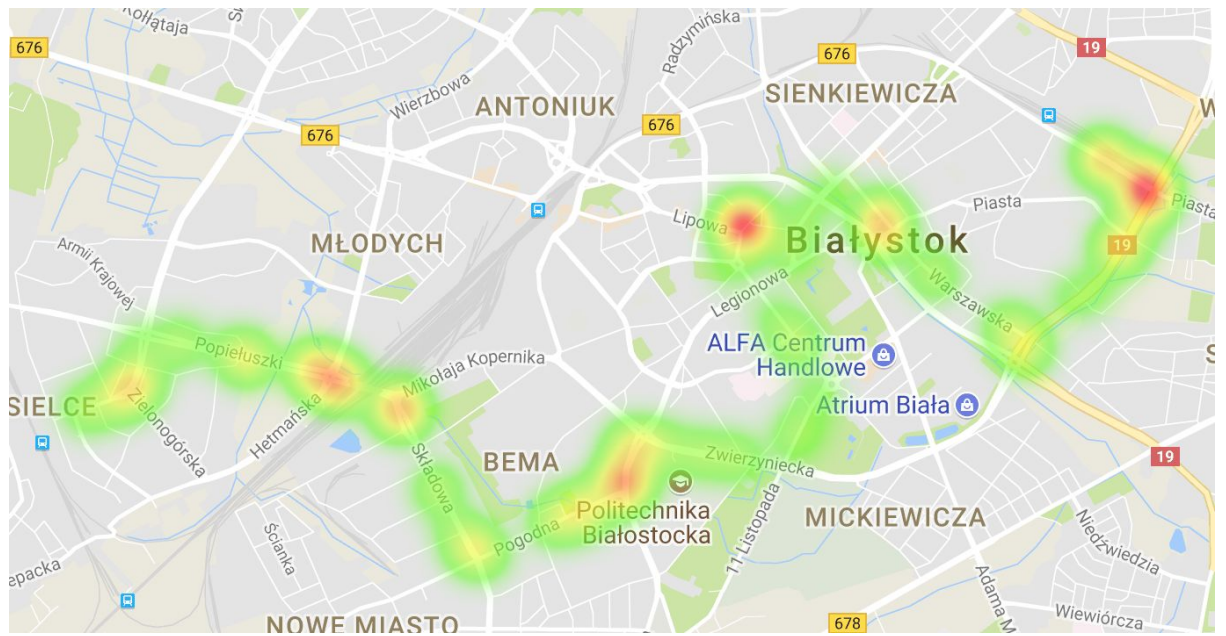
Z naszych obserwacji wynika że największe opóźnienia w czasach przejazdów komunikacją miejską następują w godzinach szczytu. Związane jest to z przemieszczaniem się ludzi do miejsca pracy.

Poniższe rysunki prezentują zaobserwowane opóźnienia dla linii 8w następujących godzinach:

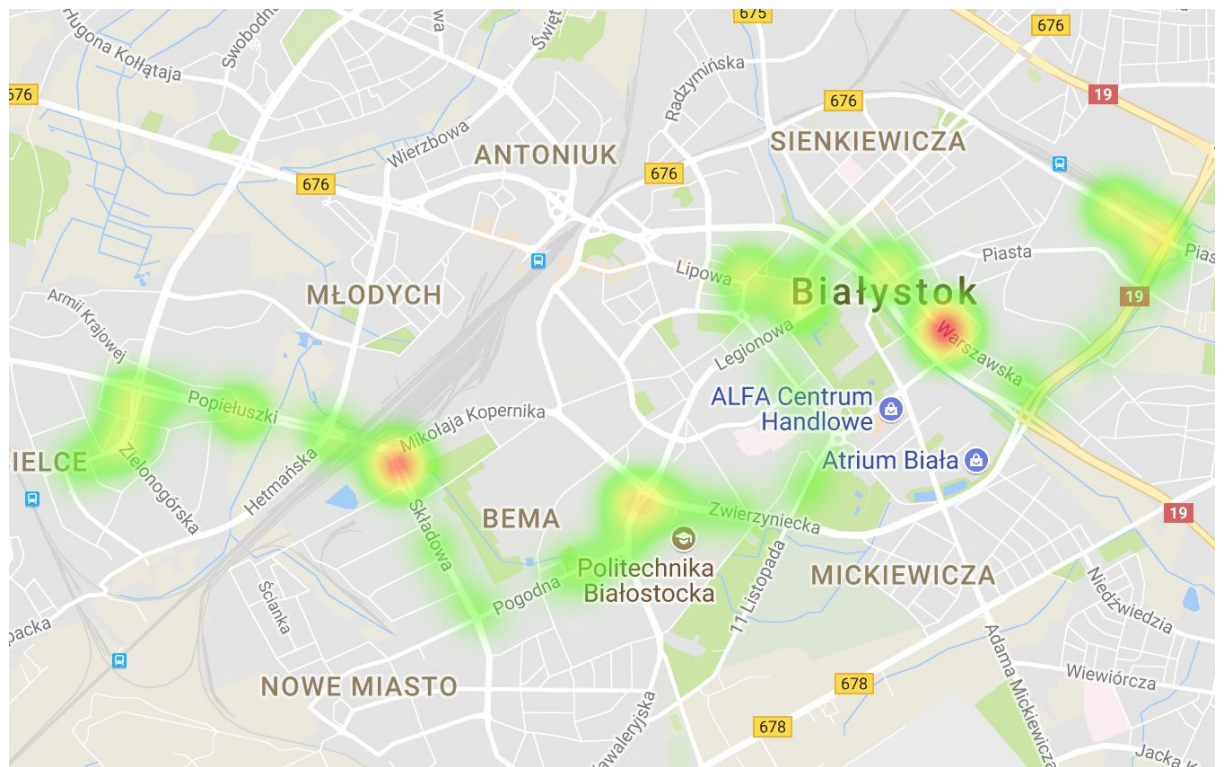
a) 16:00-18:00



b) 18:00-20:00



c) 6:00-8:00



d) 8:00-10:00

