Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Zaawansowane zagadnienia sieci neuronowych

Predykcja liczby zachorowań na COVID-19 za pomocą grafowych sieci czasowo-przestrzennych - dokumentacja wstępna

Belniak Michał,
Młyniec Paweł

Warszawa, 8 czerwca 2021

1 Opis zadania

Zadanie polegało na zaimplementowaniu i wytrenowaniu modelu czasowo-przestrzennej grafowej splotowej sieci neuronowej do predykcji liczby zachorowań na COVID-19 dla danych ze Stanów Zjednoczonych. Posłużyliśmy się danymi dostarczonymi przez The New York Times [2], które są podzielone na dane dla całego państwa, dla stanów i dla hrabstw. W chwili tworzenia projektu zbiór ten zawierał dane z 482 dni. Celem projektu była predykcja liczby zachorowań przy podziale na hrabstwa. Model zgodnie z założeniami miał dokonywać predykcji na konkretny moment czasowy na podstawie poprzedzającej owy moment sekwencji ramek czasowych. Zadanie miało także obejmować weryfikację działania modelu na danych benchmarkowych o zachorowaniach na ospę wietrzną w Węgrzech w latach 2004-2014 [3], zawierającym dane z 521 tygodni (521 ramek) oraz porównanie modelu z prostymi lokalnymi modelami autoregresyjnymi.

2 Opis rozwiązania.

Wykorzystanym modelem, zgodnie z założeniami, była grafowa splotowa sieć czasowo-przestrzenna (*Spatio-Temporal Graph Convolutional Network* [1]). Model ten dobrze pasuje do problemu zadania, jako że rozprzestrzenianie się wirusa SARS-Cov-2 jest zależne od czasu i hipotetycznie od przestrzennego rozmieszczenia ludności oraz poziomu przepływu ludności między lokalizacjami. Działanie modelu oparte jest na danych w formie ważonego, skierowanego grafu, dlatego pierwszym etapem było przygotowanie zbioru danych w formie odzwierciedlającej graf, gdzie hrabstwa sąsiadujące ze sobą lub leżące blisko siebie powinny być w grafie połączone krawędziami.

Aby stworzyć ważoną macierz sąsiedztwa wykorzystaliśmy metodę przedstawioną w [1], gdzie wagi krawędzi w grafie obliczane były na podstawie wzoru:

$$w_{ij} = \begin{cases} exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) & \text{if } i \neq j \text{ i } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) > \epsilon \\ 0 & \text{w p. p.} \end{cases}, \tag{1}$$

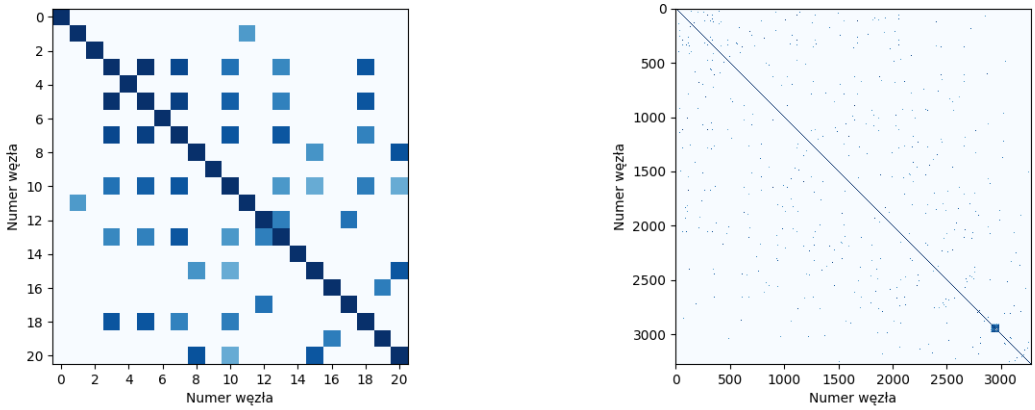
gdzie w_{ij} jest wyznaczoną wagą krawędzi (wagą sąsiedztwa) uwarunkowaną przez odległość d_{ij} między hrabstwem i oraz j. σ^2 oraz ϵ są stałymi kontrolującymi rozkład i gęstość macierzy sąsiedztwa. W naszym przypadku przyjęcie $\sigma = 100$ oraz $\epsilon = 0.5$ pozwoliło uzyskać rozsądną gęstość macierzy. Współrzędne hrabstw i odległości między nimi zostały wyznaczone z wykorzystaniem biblioteki *GeoPy*[6]. Odległość była liczona w kilometrach. W ramach zadania stworzyliśmy w ten sposób 2 zbiory danych - jeden mniejszy, zawierający 21 hrabstw z kwadratowego regionu wyznaczonego przez współrzędne (40°N, -120°W) i (35°N, -115°W), oraz pełen zbiór, zawierający wszystkie 3274 hrabstwa znajdujące się w oryginalnym zbiorze [2]. W mniejszym zbiorze znajdują się 23 dwustronne krawędzie, natomiast w pełnym jest ich 22049 (nie licząc krawędzi własnych, tj. (i, i)).

Następnie, zaimplementowaliśmy wspomniany wcześniej model z wykorzystaniem biblioteki PyTorch Geometric Temporal [4]. Głównymi elementami składowymi modelu są splotowe czasowo-przestrzenne bloki (*ST-Conv block*), zawierające po 2 warstwy **GLU**, (*Gated Linear Unit* [8]) przeprowadzające wyłuskiwanie informacji o zmienności cechy w czasie, oraz warstwę splotu grafowego między nimi (*Spatial Graph-Conv*), której zadaniem jest uwzględnienie przestrzennych zależności między wierzchołkami grafu. W każdej z tych warstw określa się liczbę filtrów, które oryginalnie wynoszą 64 dla warstw GLU oraz 16 dla warstwy środkowej. Tuż przed wyjściem modelu zostały umieszczone warstwa jednowymiarowego, czasowego splotu oraz warstwa gęsta.

2.1 Narzędzia

Do zaimplementowania modelu posłużyliśmy się biblioteką PyTorch Geometric Temporal [4]. Zawiera ona różne dynamiczne oraz temporalne geometryczne głębokie sieci neuronowe w tym grafowe sieci czasowo-przestrzenne.

Kod uruchamia się po postawieniu środowiska *conda* i został ustrukturyzowany wykorzystaniem biblioteki *PyTorchLightning*. W trakcie testów korzystaliśmy ze środowiska *Google Colaboratory* oraz *Weights and Biases* [7]. Do wizualizacji rezultatów wykorzystaliśmy bibliotekę Streamlit.



Rysunek 1: Ważona macierz sąsiedztwa dla mniejszego zbioru danych (po lewej) oraz pełnego zbioru danych (po prawej). Ciemniejsze kolory oznaczają wyższą wagę. Zagęszczenie w okolicach węzła numer 2900 odpowiada hrabstwom znajdującym się w Puerto Rico.

3 Eksperymenty

Testy przeprowadzaliśmy głównie na mniejszym zbiorze ze względu na podobieństwo do zbioru porównawczego oraz niższe wymagania czasowe. Przeprowadziliśmy strojenie pewnych hiperparametrów modelu, a dokładnie liczby kanałów w blokach *ST-Conv* i w warstwie splotowej znajdującej się na wyjściu modelu oraz rozmiaru mini-pakietu oraz stopy uczenia dla optymalizatora. Poszukiwanie najlepszej konfiguracji liczby kanałów odbyło się metodą *grid-search* z 2-krotnym powtarzaniem, a rozmiar mini-pakietu i stopa uczenia zostały wyznaczone empirycznie ze względu na wyraźną różnicę w jakości działania modelu dla innych niż finalnie wybranych wartości. Ostateczne hiperparametry przedstawione są w tabeli 1. Liczba epok w procesie strojenia wynosiła 20. Wyniki strojenia wskazały, że oryginalne wyniki parametrów dają najlepsze rezultaty.

Stopa uczenia (*lr*)	10^{-4}
Rozmiar mini-pakietu	8
Liczba filtrów w warstwach GLU	64
Liczba filtrów w warstwach splotu grafowego	16
Liczba filtrów w warstwie wyjściowej	16

Tablica 1: Ostateczne hiperparametry modelu oraz procesu uczenia

Przebadano cztery różne horyzonty czasowe przewidywań modelu. Trzy z nich zostały zaczerpnięte z [3] i obejmują 10, 20 oraz 40 tygodni. Czwartym horyzontem jest 1 dzień dla danych dotyczących zachorowań na COVID-19 oraz 1 tydzień dla danych z [3], co wynikało z postaci danych. Dla danych *COVID-19* długość sekwencji wejściowej wynosiła 12 (12 dni), natomiast dla danych *Chickenpox Cases* 8 (8 tygodni). Długości sekwencji odpowiadają tym użytym w oryginalnych artykułach. Każdy zbiór podzielono na zbiory testowy i treningowy w proporcji 1:4. Liczba epok w eksperymentach wynosiła 50. Metryką jakości modelu była wartość pierwiastka ze średniego błędu kwadratowego (**RMSE**). Wykorzystanym optymalizatorem był algorytm Adam. Wyniki ewaluacji modeli na zbiorze testowym przedstawia tabela 2. Jako że wyniki dla zbioru *Chickenpox* rozczarowują, zdecydowaliśmy się na przeprowadzenie testów z oryginalnym rozmiarem mini-pakietu (50) i oryginalną stopą uczenia (10^{-2}) (wyniki prezentuje wiersz 'Chickenpox Or.') oraz z oryginalnym rozmiarem mini-pakietu (50) i stopą uczenia równa 10^{-4} (wyniki prezentuje wiersz 'Chickenpox Mod-4'). Ponadto, wyniki porównano z lokalnym modelem autoregresyjnym **ARIMA** (*Autoregressive integrated moving average*).

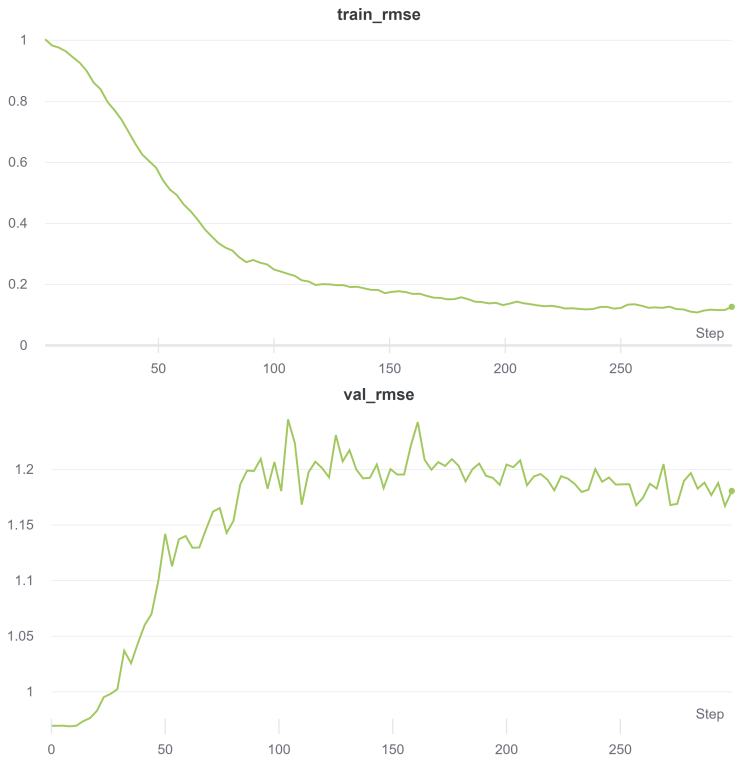
	1 dzień/tydzień	10 tygodni	20 tygodni	40 tygodni
COVID-19	0.2524	0.4025	2.446	1.72
Chickenpox	1.156	1.18	1.173	1.147
Chickenpox Or.	1.02	1.015	1.008	0.9827
Chickenpox Mod-4	1.076	1.073	1.034	1.008
ARIMA	0.346	1.749	2.047	1.535

Tablica 2: Wyniki eksperymentów w postaci uzyskanych wartości pierwiastka ze średniego błędu kwadratowego.

4 Omówienie wyników.

Stosunkowo dobre wyniki udało się uzyskać na zbiorze *COVID-19* dla 1-dniowego oraz 10-tygodniowego horyzontu czasowego. Dla większych horyzontów wyniki są nawet kilkukrotnie gorsze, przy czym najmniejszą jakość model osiąga dla horyzontu 20 tygodni, a nie tak jak można by się spodziewać dla 40 tygodni. Najprawdopodobniej jest to związane z faktem, że im dłuższy horyzont czasowy, tym mniej danych pozostaje w zbiorze. Rolę może też grać specyfika danych. Ponadto, dla horyzontu 1 dnia i 10 tygodni wartość metryki RMSE jest niższa niż w przypadku modelu ARIMA, więc wytrenowany model jest skuteczniejszy niż prostsze modele autoregresyjne w krótkim horyzoncie czasowym.

Niestety, w przypadku zbioru *Chickenpox Cases* nie udało się osiągnąć takich wyników jak w oryginalnym artykule. Przy różnych konfiguracjach procesu uczenia wartość metryki **RMSE** pozostawała na relatywnie wysokim poziomie. Próbowano także zmieniać liczbę kanałów w modelu, lecz nie przynosiło to poprawy. Obiecujące rezultaty na zbiorze treningowym uzyskano przy stopie uczenia równej 10^{-3}, gdzie wartość funkcji straty osiągała wartości rzędu 0.2 po 100 epokach i 0.1 po 200 epokach, lecz na zbiorze walidacyjnym wartość pozostawała większa niż 1 (rysunek 2). Problemem może okazać się fakt wykorzystania nieco innej struktury sieci przez autorów artykułu [3], która jednak nie została dokładnie opisana. Założyliśmy, że jest ona taka sama jak w [1] lecz z mniejszym rozmiarem filtra w warstwach GLU, aby sieć generowała odpowiedni rozmiar wektora wyjściowego dla oryginalne długości sekwencji wejściowej równej 8. Możliwe też, że wyniki poprawiłyby się, gdyby do ważenia grafu wykorzystać taką samą metodę jak w przypadku zbioru *COVID-19*. W naszych eksperymentach wykorzystywaliśmy oryginalny zbiór, w którym każda krawędź ma taką samą wagę równą 1.



Rysunek 2: Wykres wartości funkcji straty dla kolejnych epok procesu treningowego dla danych treningowych (góra) i walidacyjnych (dół) ze zbioru *Chickenpox Cases* przy ustawieniu rozmiaru mini-pakietu na 50 i wartości stopy uczenia 10^{-3}.

Model autoregresyjny okazał się zwracać dobre wyniki dla najkrótszego horyzontu czasowego, czyli 1 dnia, choć i tak są one nieco gorsze niż dla użytego modelu sieci. Natomiast przy dłuższym horyzoncie wartość wynikowej funkcji straty jest dużo większa; szczególnie słabe wyniki są prezentowane dla horyzontu 20 tygodni, choć są one lepsze niż w przypadku naszego modelu. Wnioskować można, że model ARIMA może być skuteczny przy przewidywaniach krótkoterminowych, a nie długoterminowych.

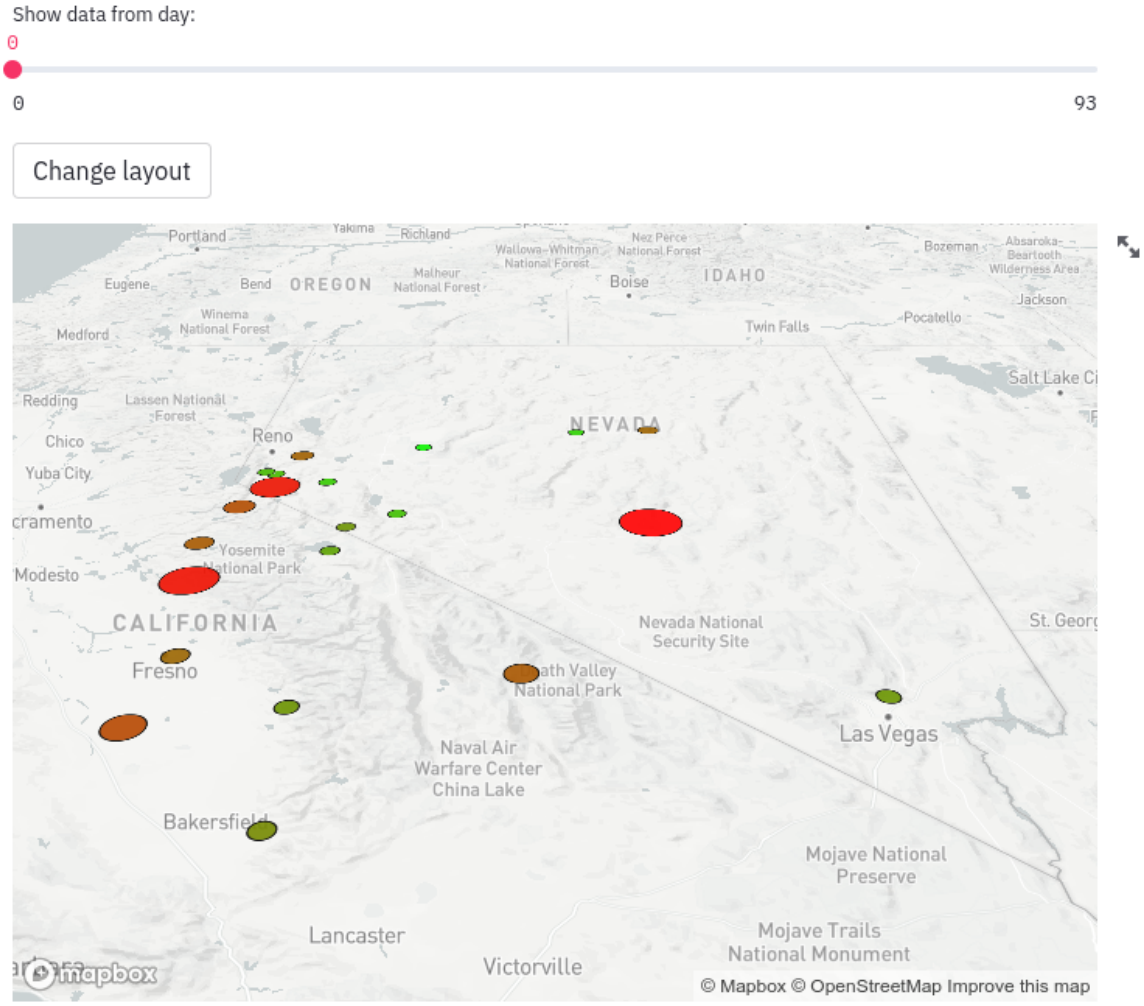
5 Wizualizacja wyników

Dla danych geoprzestrzennych została stworzona wizualizacja predykcji na mapie. W tym celu wykorzystano biblioteki *streamlit* oraz *pydeck*. Pozwalają one na otworzenie mapy w przeglądarce z naniesionymi na nią znacznikami

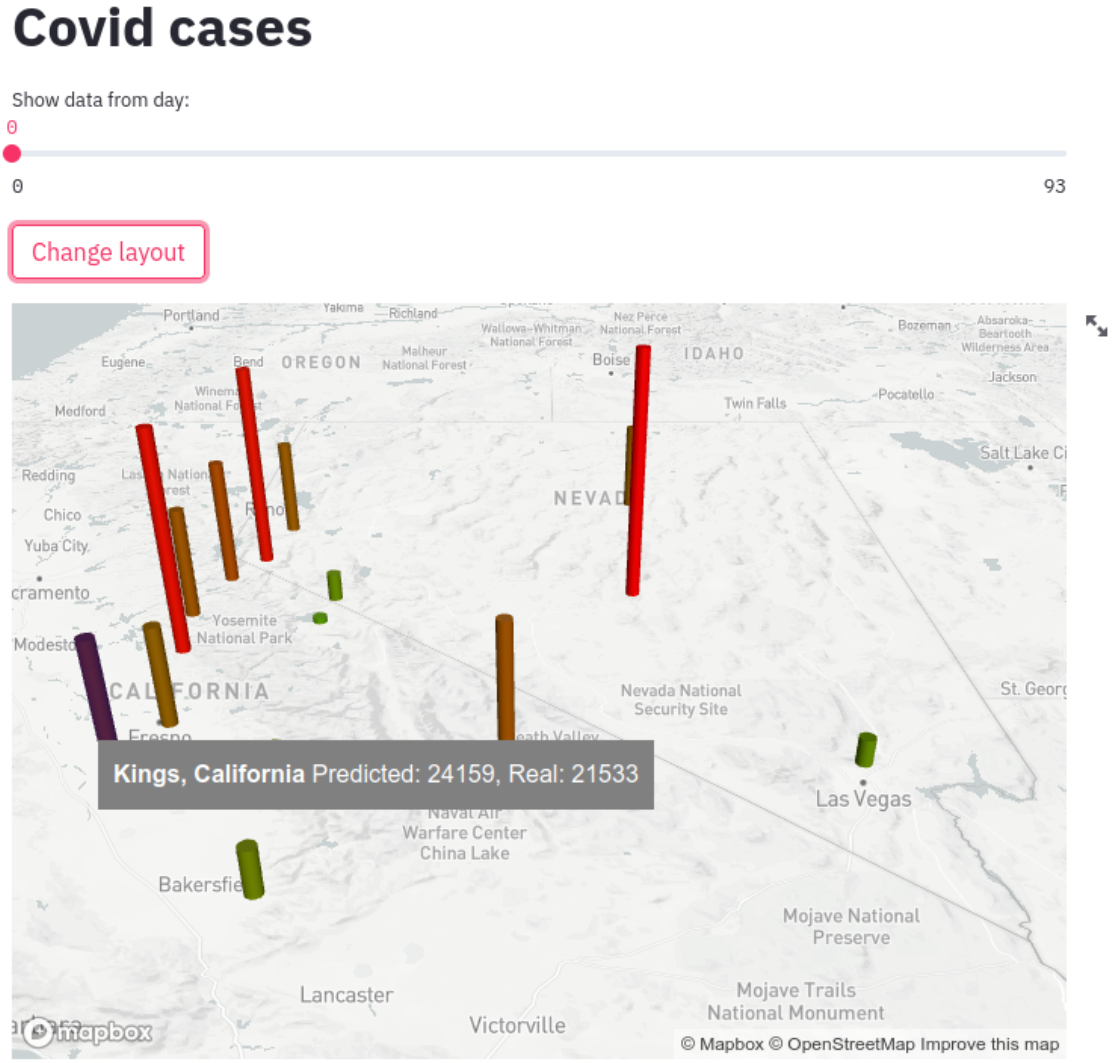
W ramach projektu skupiono się na wizualizacji liczby zachorowań na COVID-19 dla danych z Stanów zjednoczonych. Ponieważ dla predykcji danych z małą liczbą punktów lepszym znacznikiem są okręgi różnej wielkości na mapie, a dla gęstych danych lepszy wykres słupkowy stworzono więc możliwość przełączenia się pomiędzy widokami w celu wyboru lepszego sposoby prezentacji danych.

Jako, że predykcja wykonywana jest dla kolejnych kroków czasowych stworzono także suwak do zmiany widoku dla kolejnych dni.

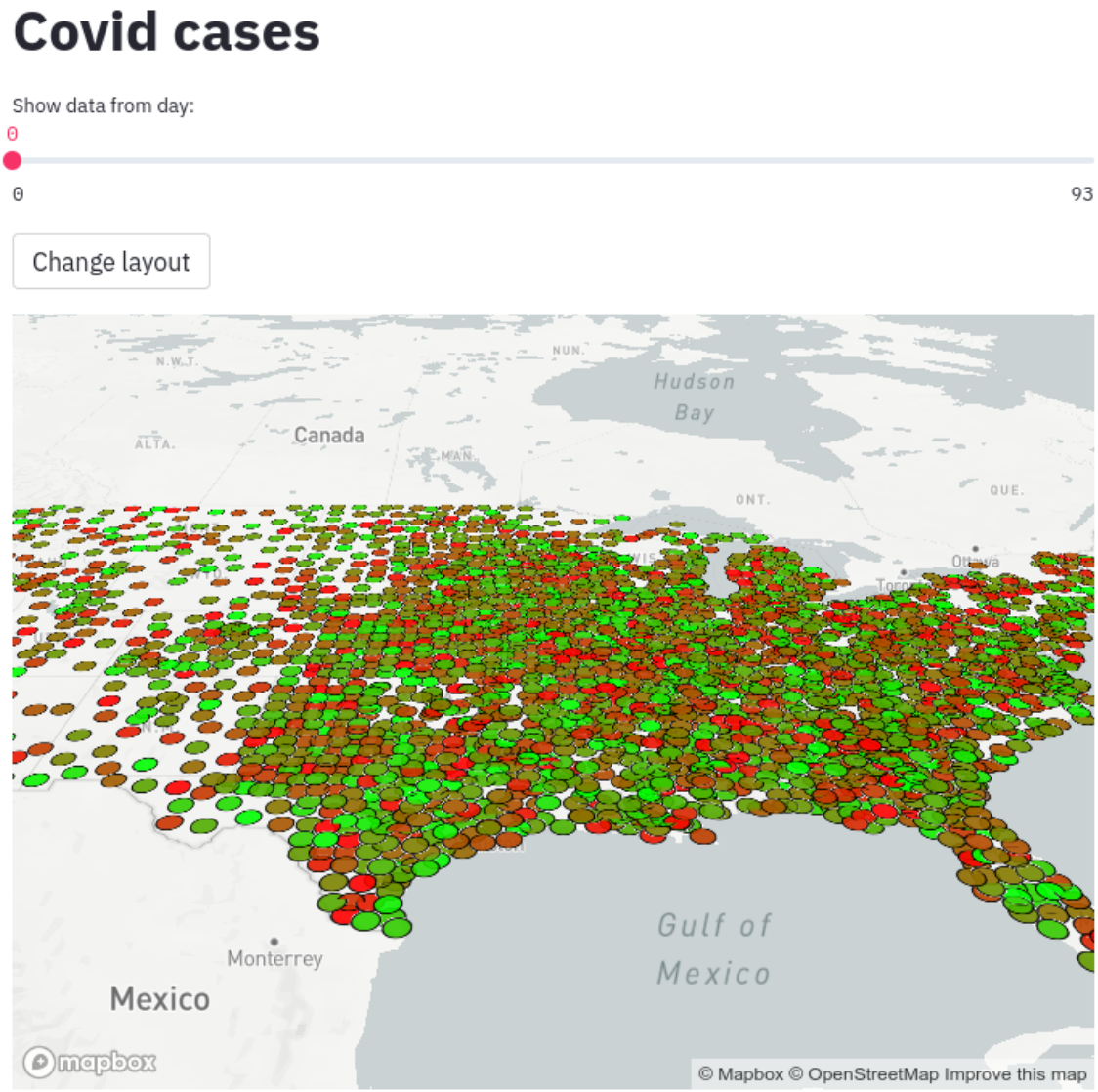
Po najechaniu na interesujący nas znacznik wyświetla się nazwa w tym wypadku hrabstwa, wartość przedyktowana oraz wartość rzeczywista. Zrzuty ekranu dla danych okrojonych oraz pełnych przedstawiono poniżej.



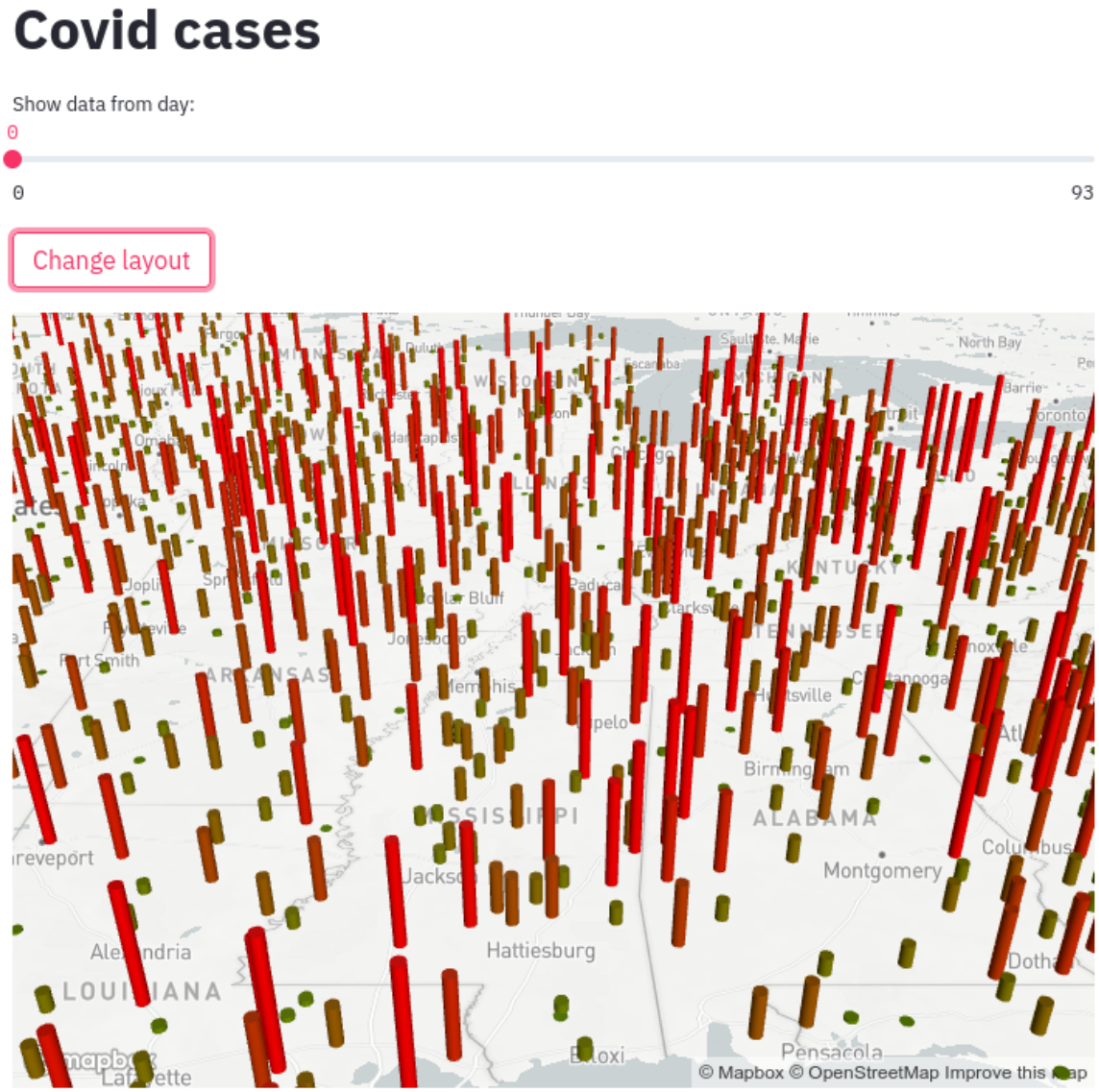
Rysunek 3: Predykcja dla hrabstw w Stanach Zjednoczonych dla danych okrojonych, punkty



Rysunek 4: Predykcja dla hrabstw w Stanach Zjednoczonych dla danych okrojonych, wykres słupkowy



Rysunek 5: Predykcja dla hrabstw w Stanach Zjednoczonych dla wszystkich danych, punkty



Rysunek 6: Predykcja dla hrabstw w Stanach Zjednoczonych dla wszystkich danych, wykres słupkowy

6 Podsumowanie

W ramach projektu zrealizowaliśmy podstawowe początkowe założenia:

- zastosowaliśmy czasowo-przestrzenną grafową sieć splotową do predykcji liczby zachorowań dla hrabstw w Stanach Zjednoczonych,
- wykonaliśmy strojenie modelu oraz trening i ewaluację dla różnych horyzontów czasowych,
- porównaliśmy wyniki ze zbiorem *Chickenpox Cases* oraz modelem autoregresyjnym ARIMA,
- dokonaliśmy wizualizacji wyników na mapie za pomocą biblioteki *Streamlit*.

Pokazaliśmy, że czasowo-przestrzenna grafowa sieć splotowa może być wykorzystana do predykcji zachorowań na COVID-19, szczególnie w krótkim horyzoncie czasowym, lecz nie udało się osiągnąć podobnych

wyników na zbiorze zachorowań na ospę wietrzną. Otrzymanie dobrych wyników na tym zbiorze wymagałoby dokładniejszej analizy tego zbioru i dokonania eksperymentów ze zmodyfikowanym modelem. Udało się również pokazać, że wspomniany model sieci neuronowej może dorównywać lub nawet przewyższać jakością proste modele autoregresyjne.

7 Bibliografia

[1] *Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting*, Bing Yu and Haoteng Yin and Zhanxing Zhu, 2017

[2] *Coronavirus (Covid-19) Data in the United States*, The New York Times, (Dostęp zdalny 14.04.2021) `https://github.com/nytimes/covid-19-data`

[3] *Chickenpox Cases in Hungary: a Benchmark Dataset for Spatiotemporal Signal Processing with Graph Neural Networks* Benedek Rozemberczki, Paul Scherer, Oliver Kiss, Rik Sarkar, Tamas Ferenci, 2102.08100, 2021 (Dostęp zdalny 06.06.2021) `https://arxiv.org/pdf/2102.08100.pdf`

[4] *pytorch_ geometric_ temporal* PyTorch, (Dostęp zdalny 14.04.2021) `https://github.com/benedekrozemberczki/pytorch_geometric_temporal`

[5] *Streamlit* (Dostęp zdalny 14.04.2021) `https://docs.streamlit.io/en/stable/api.html#streamlit.map`

[6] *GeoPy* GeoPy Contributors, 2018 (Dostęp zdalny 05.06.2021) `https://geopy.readthedocs.io/en/stable/`

[7] *Experiment Tracking with Weights and Biases*, Biewald L, 2020, Software available from wandb.com, `https://www.wandb.com/`

[8] *Language Modeling with Gated Convolutional Networks*, Yann N. Dauphin, Angela Fan, Michael Auli, David Grangier, *Proceedings of the 34th International Conference on Machine Learning*, PMLR 70:933-941, 2017 (Dostęp zdalny 05.06.2021) `https://arxiv.org/abs/1612.08083`