Eksploracja danych metodami uczenia maszynowego

Sprawozdanie

Analiza i predykcja rozwoju epidemii COVID-19

Jakub Jaszczuk, 238556

Styczeń 2021

Spis treści

1 Wstęp

1.1 Opis projektu

Celem projektu jest analiza i predykcja rozwoju epidemii COVID-19. W tym celu zostanie wykorzystany model regresyjny. W celu modelowania rozwoju epidemii można zastosować jeden z wielu wielokrotnie sprawdzonych modeli rozwoju epidemii, takich jak SIR, SEIR oraz ich liczne rozszerzenia. Takie modele zazwyczaj posiadają niewielką liczbę parametrów i są przedstawiane przy pomocy równań różniczkowych, jednak możliwe jest też wykorzystanie modelu wieloagantowego albo automatu komórkowego. Przy takich modelach przewidywane są liczności populacji odpowiednich do zastosowanego modelu, a łączna ich liczba nie może przekroczyć liczności całej populacji. w przypadku modelu regresyjnego nie istnieje prosty sposób na ograniczenie wartości maksymalnej. Jako że w ramach pracy magisterskiej zajmuję się wieloagentowym modelem SEIR to w ramach tego projektu chciałbym się zająć modelem regresyjnym z możliwością późniejszego porównania obu metod i wskazania ich mocnych stron oraz słabości.

1.2 Źródła danych

W internecie można znaleźć wiele źródeł danych o COVID-19. Są to zarówno źródła krajowe jak i światowe, zagregowane. W ramach projektu postanowiono wykorzystać dane udostępnione przez Center for Systems Science and Engineering na uniwersytecie Johns Hopkins, które dostępne są pod adresem https://github.com/CSSEGISandData/COVID-19[1]. Wśród dostępnych danych znajdują się dane surowe z kolejnych dni oraz przygotowane pliki csv tworzące szeregi czasowe. Wybrano dane surowe ze względu na więcej możliwości ich samodzielnego przygotowania, analizowania i przetworzenia.

1.3 Zastosowana technologia

W ramach projektu należy pobrać dane z wybranego źródła, przetworzyć je oraz przygotować model. Dane znajdują się w publicznym repozytorium w serwisie Github, dlatego w celu ich pobrania został wykorzystany system kontroli wersji Git. Pozawala ona na bezproblemowe pobierane zaktualizowanego zbioru danych przy pomocy jednego polecenia. W celu przetworzenia danych został wykorzystany język programowania Python wraz z bibliote-

ką Numpy[2] oraz Pandas[3]. Pandas jest popularną biblioteką do analizy i manipulacji danymi dla języka Python. Pozwala ona an łatwe wczytywanie i zapisywanie plików csv. Ponadto stanowi potężne narzędzie do manipulacji danymi tabelarycznymi, co idealnie pasuje do analizy serii czasowych, gdzie indeksami są koleje daty oraz kraje. Numpy został wykorzystany w kilku miejscach, gdzie Pandas nie dostarcza odpowiednich funkcjonalności. Do przygotowania wykresów wykorzystano bibliotekę Matplotlib/Seaborn.

2 Eksploracja danych

Pierwszy etap przygotowania danych polegał na ich pobraniu, połączeniu w jeden plik, wybraniu istotnych informacji. Dane składają się z jednego pliku na każdy dzień, co na chwilę obecną daje ponad 300 osobnych plików. Początek zbierania informacji to 22 stycznia 2020 roku. Wartości dla kolejnych dni są zagregowane, jednak zdarzają się nieprawidłowości. Objawiają się one spadkiem liczby zachorowań. W większości wypadków są to anomalie jednodniowe, ale w niektórych dotyczą one nawet okresu dwóch tygodni. W celu uzyskania plików zawierających gotowe szeregi czasowe zastosowano następujące transformacje danych:

1. Usunięcie zbędnych kolumn. Do takich kolumn zaliczają się: współrzędne geograficzne, numer administracyjny dla USA, hrabstwo (dotyczy tylko USA), data aktualizacji, pełna nazwa lokalizacji.

2. Normalizacja nazw kolumn. W plikach zawierających dane znajdują się 4 różne formaty danych. Różnią się one liczbą kolumn oraz ich nazwami. Początkowo było 6 kolumn, a w nowszych plikach jest ich 13. Przykładową zmianą nazwy jest zastąpienie „Country/Region” przez „Country_Region”

3. Usunięcie kraju o nazwie „Recovered”. Taki kraj przedstawia liczbę wyzdrowiałych w USA.

4. Agregacja stanów/regionów do krajów. W przypadku Australii, Chin, Kanady i USA dane są przedstawione na poziomie regionu i należy je zagregować by otrzymać dane na poziomie kraju.

5. Rozdzielenie na osobne szeregi czasowe przypadków potwierdzonych, zgonów i wyzdrowiałych. W celu łatwiejszego zarządzania te informacje zostały rozdzielone na osobne pliki.

6. Połączenie powtarzających się nazw krajów. Żmudny proces przeglądania nazw krajów, czy taki kraj istnieje, czy jest to inna nazwa dla istniejącego kraju. W wielu przypadkach inna nazwa była użyta wyłącznie raz, co powodowało jednodniową przerwę w danych dla danego kraju. Do przykładów należą: „Czechia" - „Czech Republic", „Iran" - „Iran (Islamic Republic of)", „Gambia" - „Gambia, The" - „The Gambia", „Ivory Coast" - „Cote d'Ivoire".

7. Usunięcie krajów dla których dane są mało kompletne. Usunięcie następuje jeżeli ponad połowa wartości nie jest określona. Powoduje to usunięcie 52 krajów. Takie mało kompletne kraje mogą powstawać na skutek połączenia z innym krajem w późniejszych datach ze względu na dość burzliwy początek walki z COVID.

8. Wypełnienie brakujących wartości zerami. Brak wartości pojawia się z powodu braku informacji o zarażeniach w danym kraju. Braki wartości najczęściej występują na początku pandemii.

9. Ostatnim etapem jest zapisanie gotowych i przetworzonych szeregów czasowych do plików csv. Dla każdej kategorii ilości pacjentów powstaje osobny plik z danymi. Takie dane można bezproblemowo ładować w celu dalszych analiz i przygotowania modelu.

Dla danych w ten sposób przetworzonych można przeprowadzać dalsze operacje. Te operacje zostały wydzielone jako drugi etap ze względu na operowanie na szeregach czasowych, gdzie pierwszy etap polegał na przygotowywaniu danych.

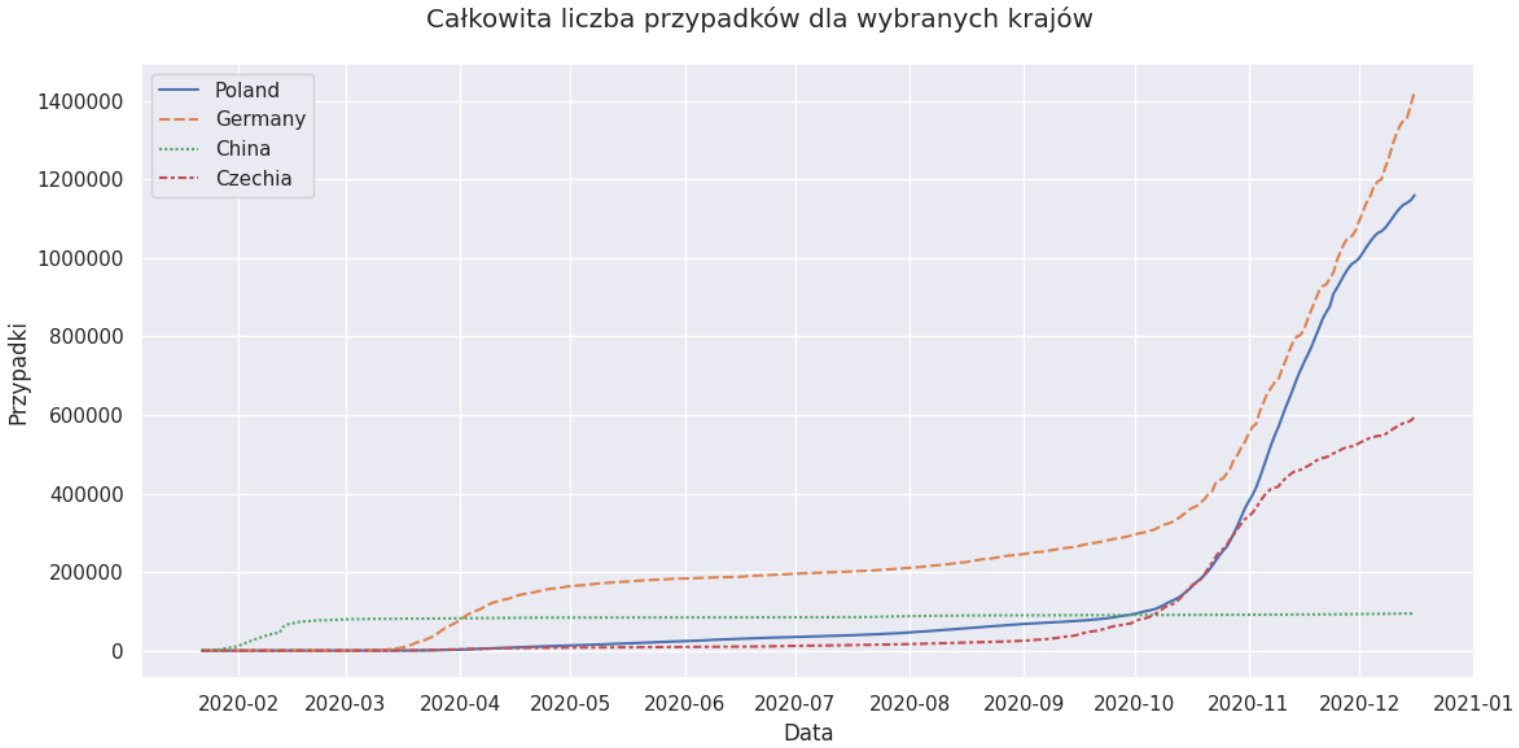
1. Sprawdzenie monotoniczności danych zagregowanych. W 67 przypadkach dla 44 krajów występują anomalie. 11 krajów ma ponad jedna anomalię.

2. Próba naprawy dla anomalii jednodniowych. 15 krajów udaje się naprawić. Naprawa polega na liniowej interpolacji pomiędzy poprzednim i następnym dniem i jest wykonywana jedynie jeżeli poprzedni dzień ma mniejszą wartość niż następny.

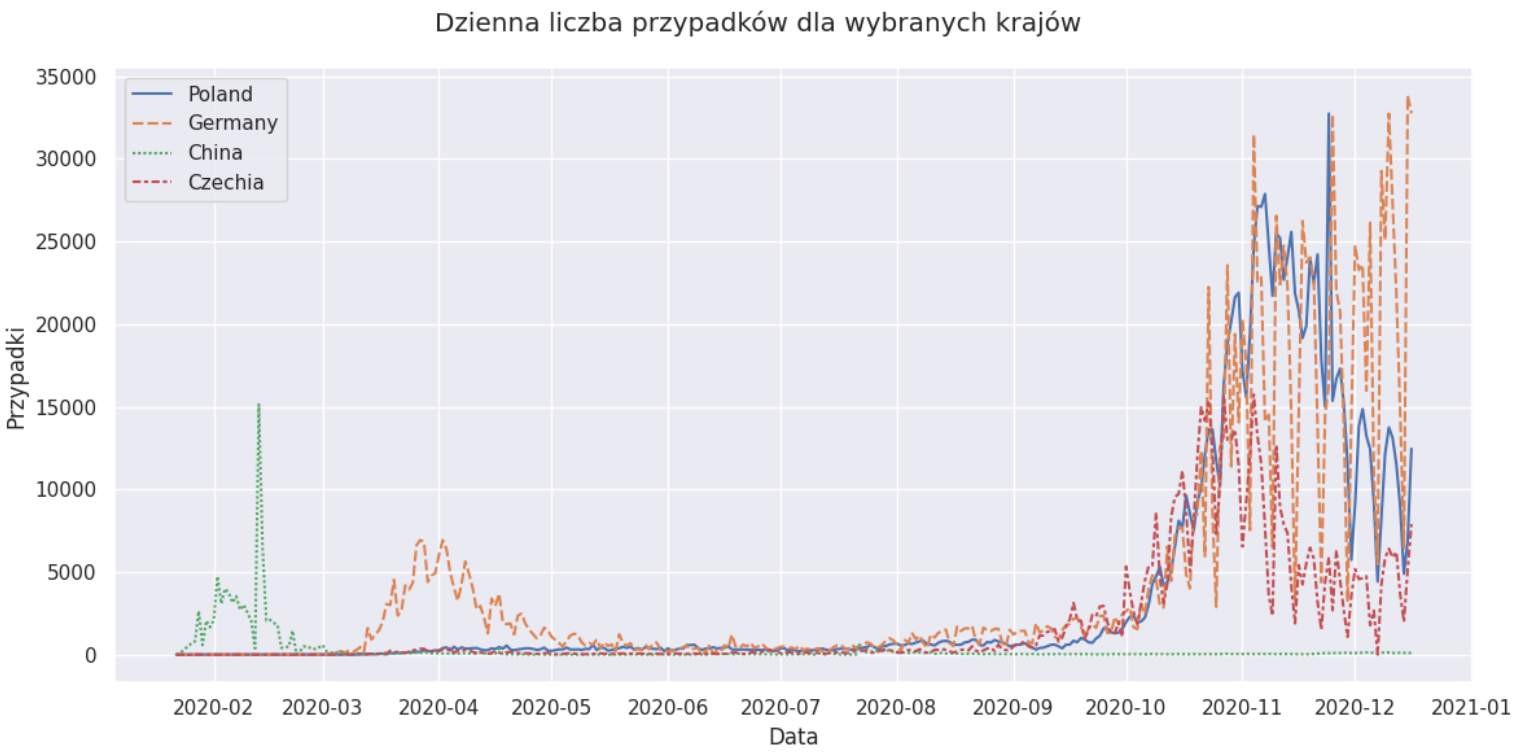
3. Usunięcie pozostałych krajów z anomaliami. W większości przypadków brak monotoniczności obejmuje dłuższy okres i nie istnieje metoda pozwalająca w prosty sposób na znalezienie prawdziwych wartości.

Należałoby wykorzystać inny zbiór danych w celu weryfikacji lub wyszukiwać możliwe opisane zmiany w postaci erraty.

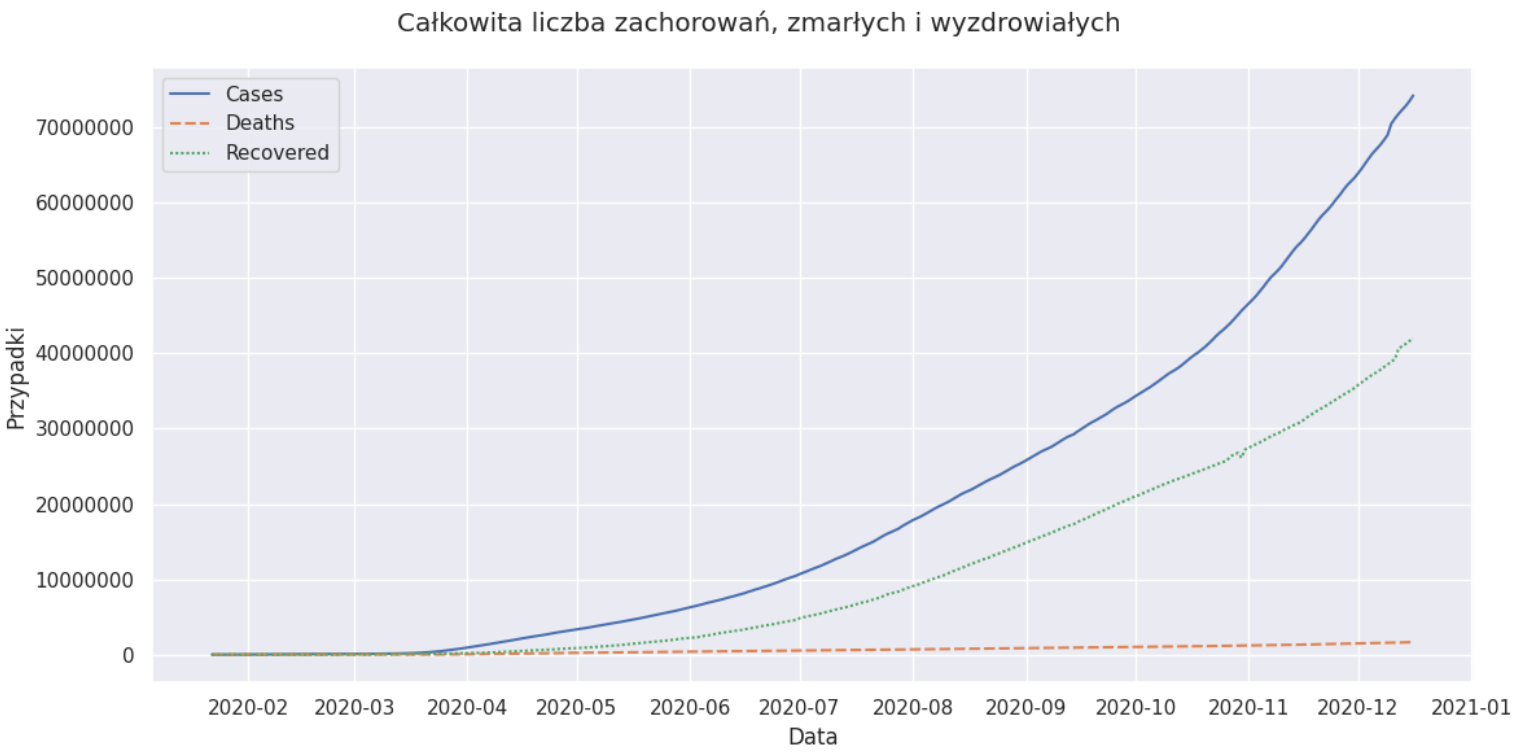
4. Deagregacja wartości. W celu otrzymania dziennej liczby nowych przypadków postanowiono dokonać deagregacji wartości. W tym celu wykorzystano funkcje diff z biblioteki pandas.



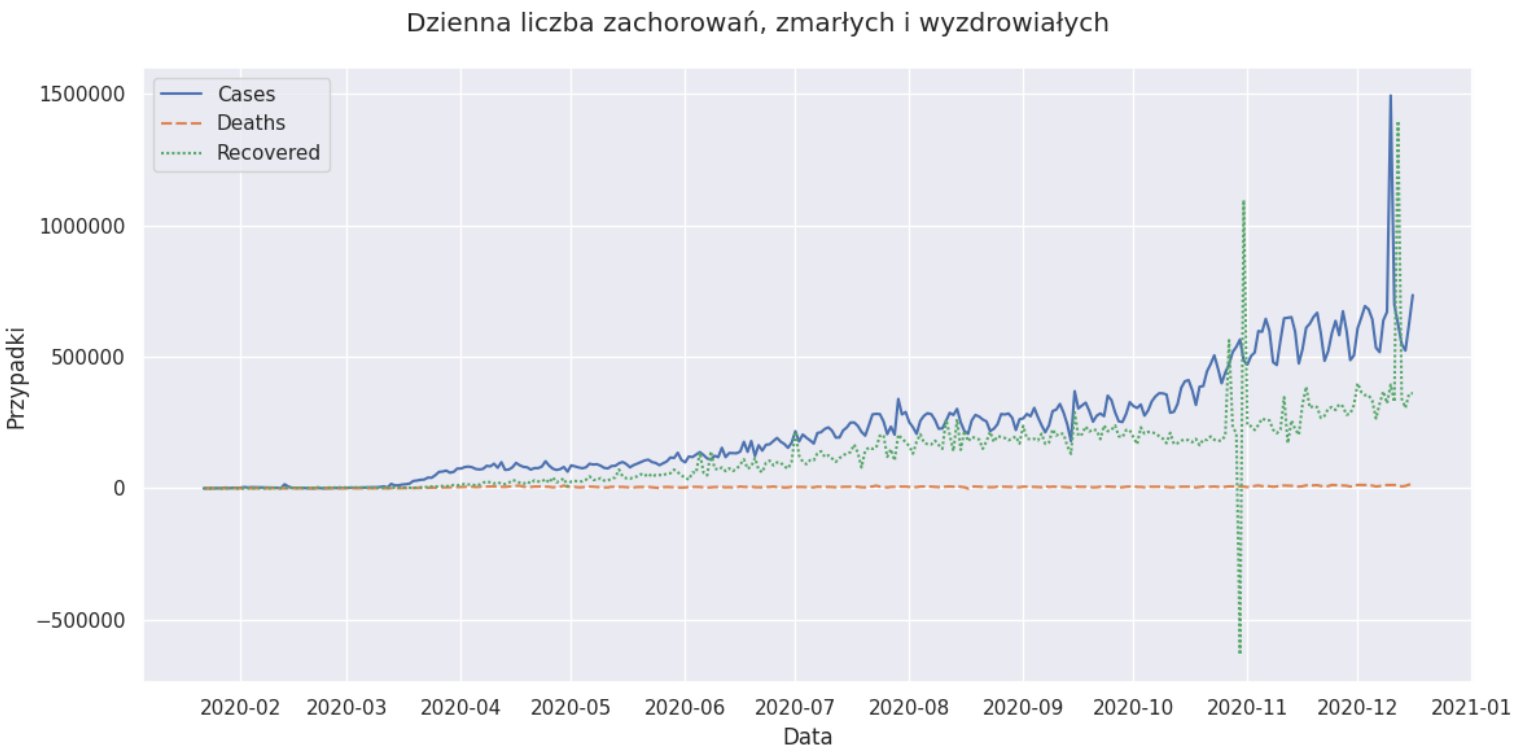
Rysunek 1: Całkowita liczba zachorowań w wybranych krajach.



Rysunek 2: Dzienna liczba zachorowań w wybranych krajach.



Rysunek 3: Globalna liczba przypadków w poszczególnych kategoriach.



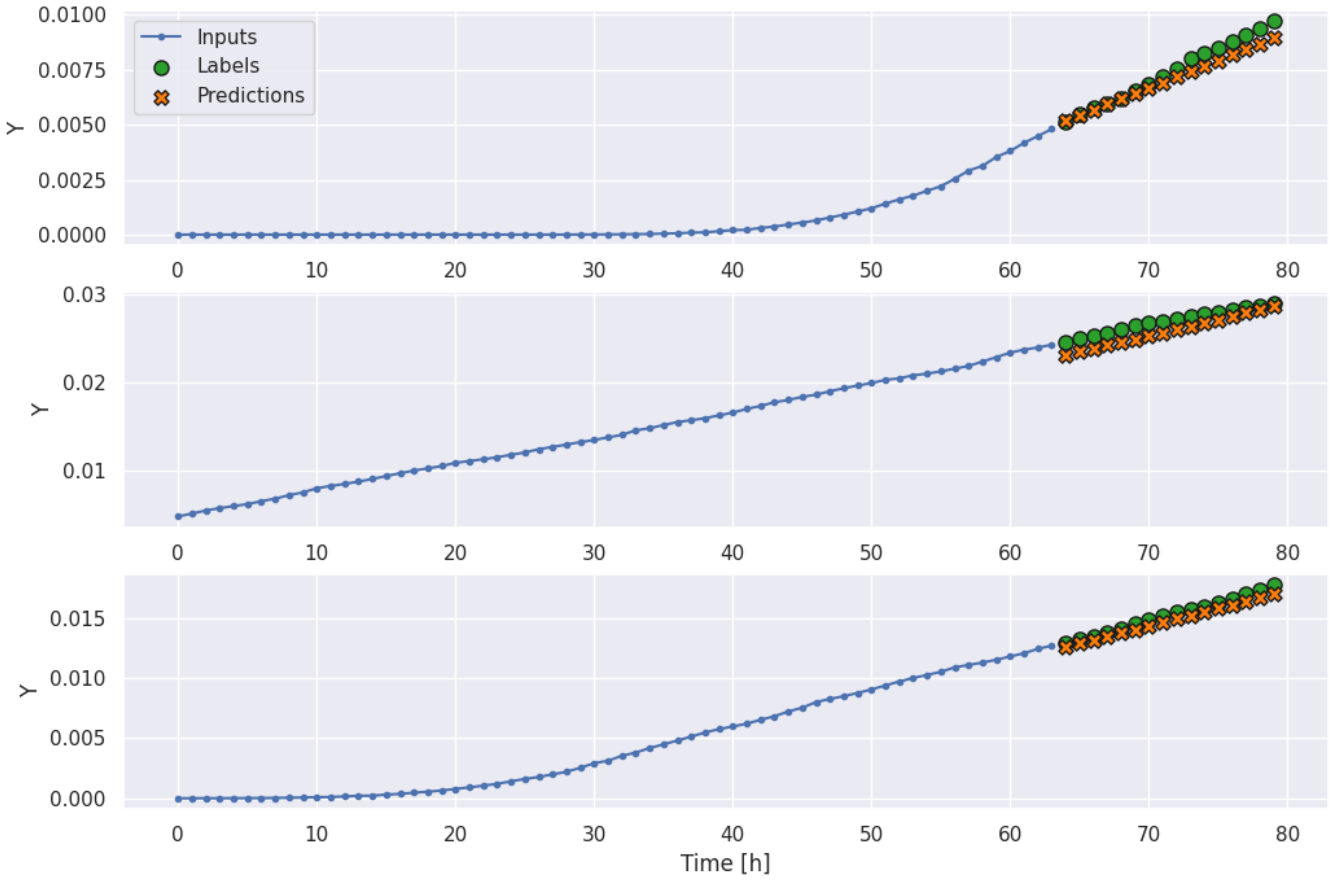
Rysunek 4: Dzienna globalna liczba przypadków w poszczególnych kategoriach.

Widoczne na rysunku 4 nagłe skoki wartości są spowodowane przez dwie różne rzeczy. Skok w liczbie wyzdrowiałych (zielona linia) jest spowodowany pominięciem jednej cyfry dla Kolumbii, co skutkuje wartością dziesięciokrotnie mniejsza dla 29 października, a sam wykres pokazuje zmiany w dziennych liczbach i z tego powodu błąd został rozpropagowany na 2 dni. Błąd w liczbie zachorowań, przypadający na 10 grudnia, spowodowany jest zmianą sposobu podawania informacji w Turcji. Postanowiono od tego dnia uwzględniać również osoby asymptomatyczne i o znikomych objawach.

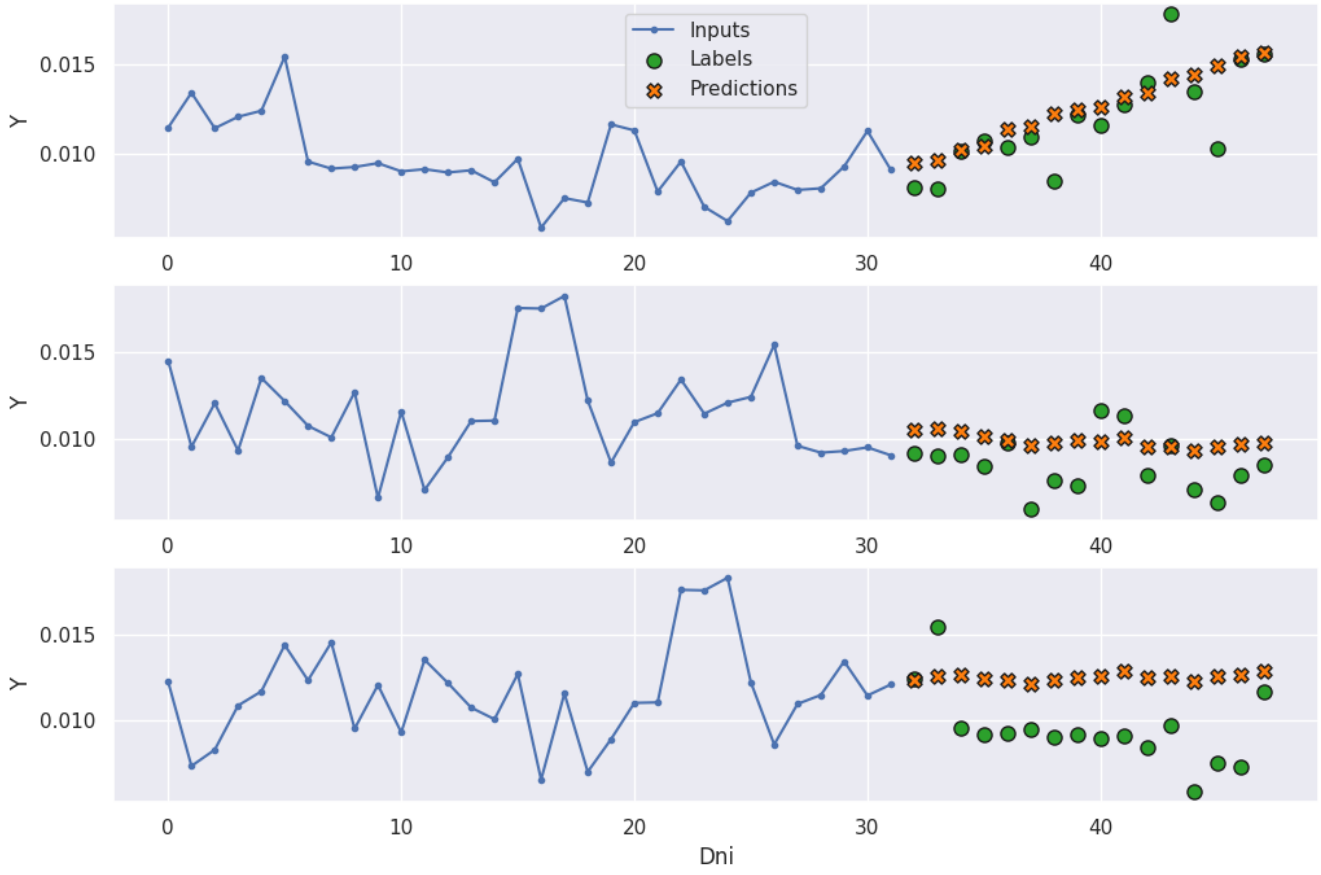
3 Uczenie maszynowe

W celu wyuczenia modelu postanowiono wykorzystać bibliotekę TensorFlow (https://www.tensorflow.org/). Pozwala ona na zastosowanie różnorod-

nych modeli, w tym sieci neuronowych i rekurencyjnych sieci neuronowych. Wybranym rodzajem sieci neuronowej jest siec typu LSTM (Long short-term memory). Dodatkowo istnieje możliwość wykorzystania biblioteki TensorFlow Probability, która umożliwia analizę szeregów czasowych oraz zastosowanie modeli autoregresyjnych. Jednak ze względu na potwierdzoną lepszą skuteczność modeli w postaci sieci neuronowych zdecydowano się na wykorzystanie jedynie tego rodzaju modeli[5]. Sieć LSTM może się okazać lepsza nawet o 85% od modelu ARIMA[6]. Na rysunkach 5 i 18 przedstawiono przykładowe wykresy przedstawiające dane rzeczywiste oraz przewidywania przykładowego modelu. Model nie był jeszcze specjalnie dobierany, a powstał jedynie w celu sprawdzenia działania biblioteki oraz sposobu korzystania z niej. W przypadku danych zakumulowanych przewidywania są w miarę bliskie wartością prawdziwym, a w wypadku danych dziennych przewidywania zazwyczaj zachowują trend.



Rysunek 5: Porównanie przewidywań modelu LSTM z danymi rzeczywistymi dla Polski.



Rysunek 6: Porównanie przewidywań modelu LSTM z danymi rzeczywistymi dla Polski.

4 Wyniki

4.1 Opis zastosowanej metody

W celu otrzymania wyników, czyli predykcji liczby zachorowań na COVID-19 zastosowano sztuczną sieć neuronowa typu LSTM. Sieć taka składa się

wektora wejściowego cech, dla rozważanego problemu jest to liczba przypadków w poszczególnych dniach, warstw komórek LSTM i warstwy wyjściowej, która przedstawia liczbę przewidywanych przypadków dla poszczególnych dni. Wielkość warstwy wejściowej zależy od ilości ostatnich dni branych pod uwagę, a wielkość warstwy wyjściowej zależy od liczby predykowanych dni. Komórka LSTM składa się z kilku wejść oraz stanu wewnętrznego, dzięki któremu może zapamiętywać wartości z przeszłości. Wykorzystano bibliotekę Tensorflow z interfejsem Keras.

4.2 Wybór modelu

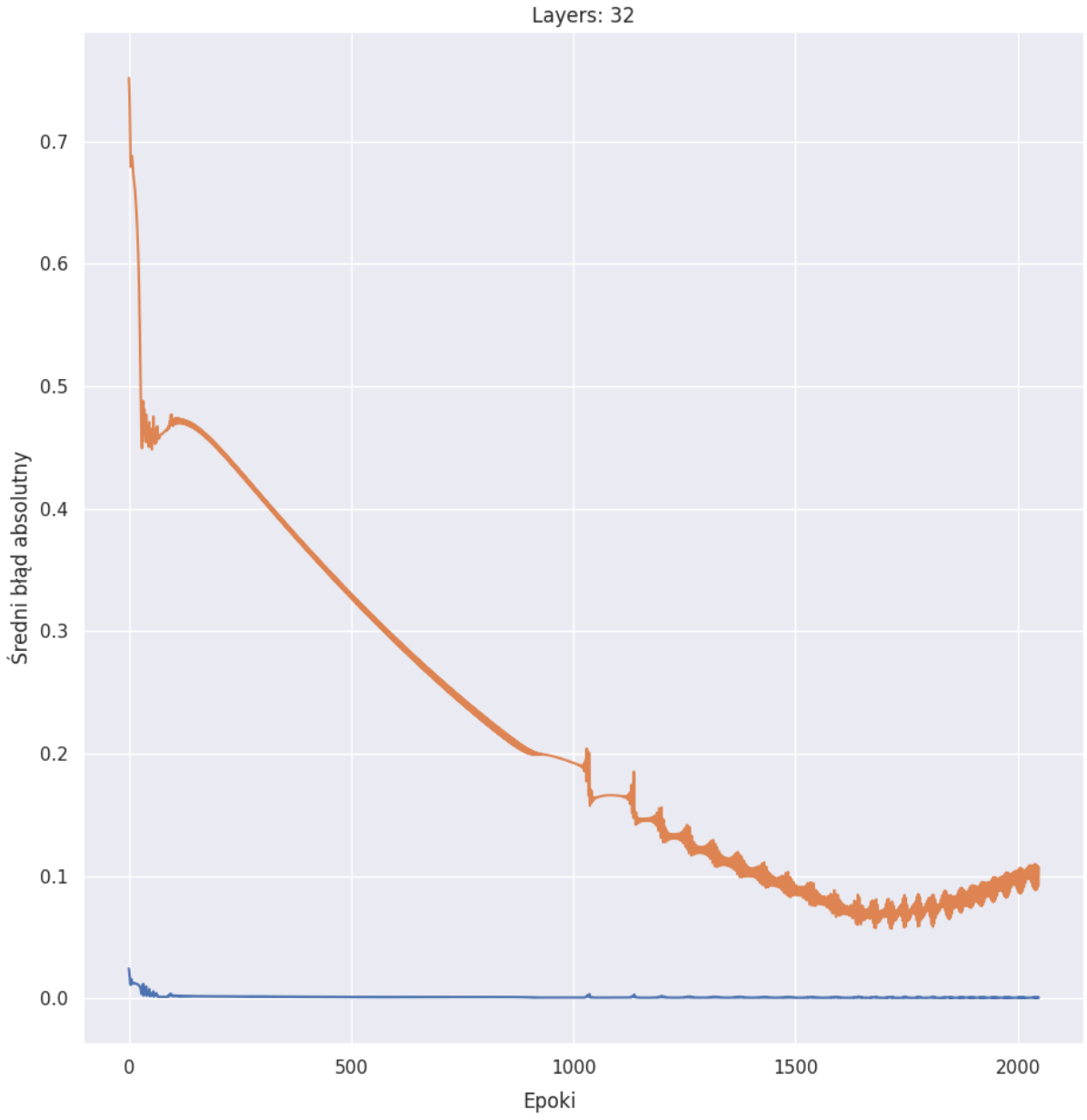
W celu wybrania modelu porównano kilka rozwiązań sieci neuronowej. Konkretną architekturę sieci należało ustalić w sposób eksperymentalny. Liczbę dni wejściowych ustalono na 60, a liczbę dni predykcji na 14. Większa ilość dni wejściowych mogłaby zwiększyć szum w wynikach i czas uczenia, a mniejsza ilość dni miałaby negatywny pływ na dokładność modelu, dlatego 60 dni uznano za kompromis. Przewidywanie na większą ilość dni uznano za bezcelowe ze względu na malejącą dokładność przewidywania. Jako warstwę wyjściową sieci neuronowej wybrano warstwę gęstą z funkcja aktywacji swish[4]. Wypróbowano też inne funkcje jednak dawały gorsze rezultaty, a w przypadku funkcji relu nie dochodziło do uczenia. W przypadku warstw LSTM zaproponowano następujące ich wielkości:

- 32
- 60
- 90
- 60 i 120 (2 warstwy)

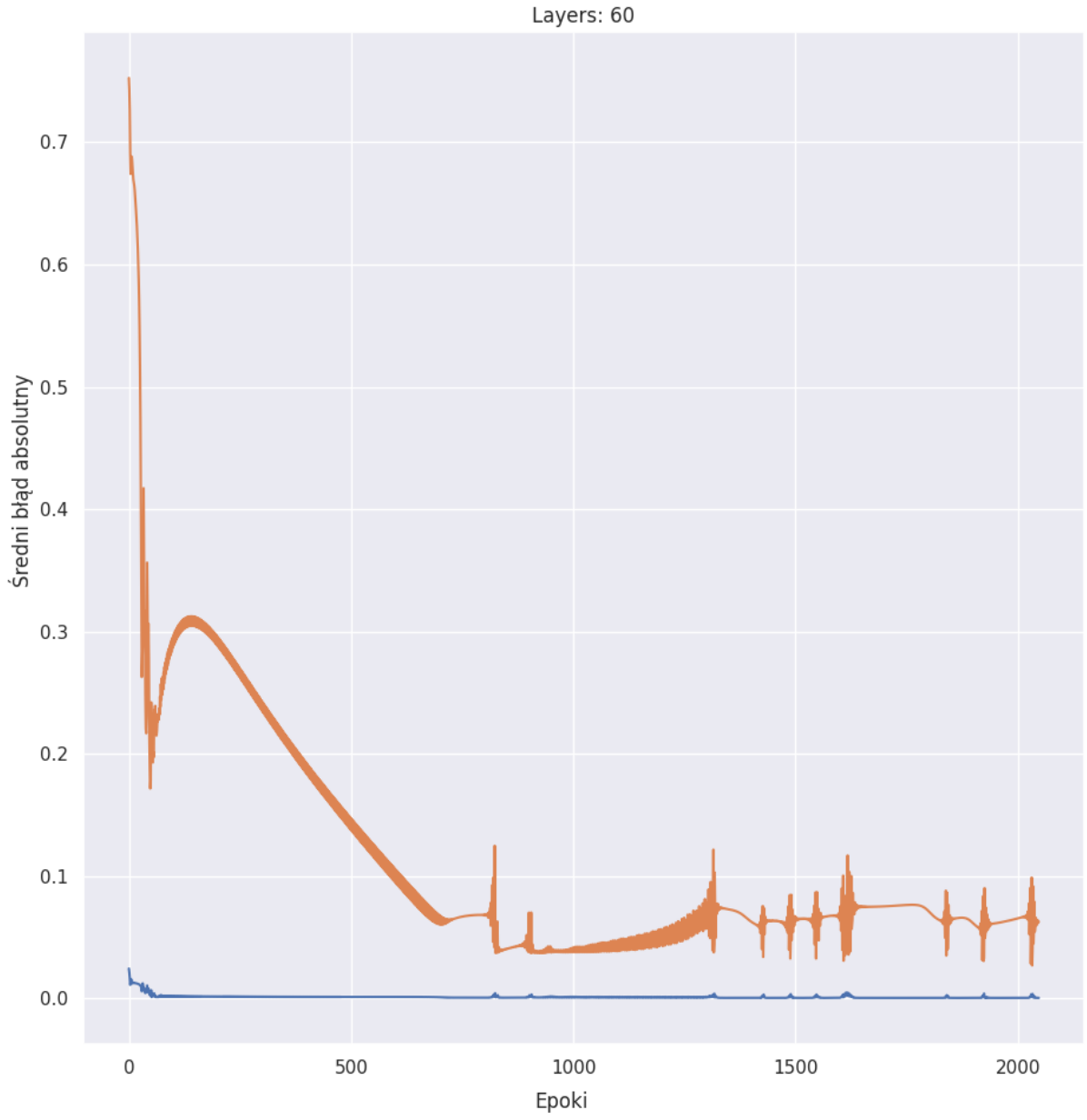
Przygotowano łącznie 3 modele, po jednym dla każdej rozważanej postaci problemu:

- Dane zagregowane
- Dane dzienne
- Dane dzienne wygładzone 7-dniową średnią ruchomą

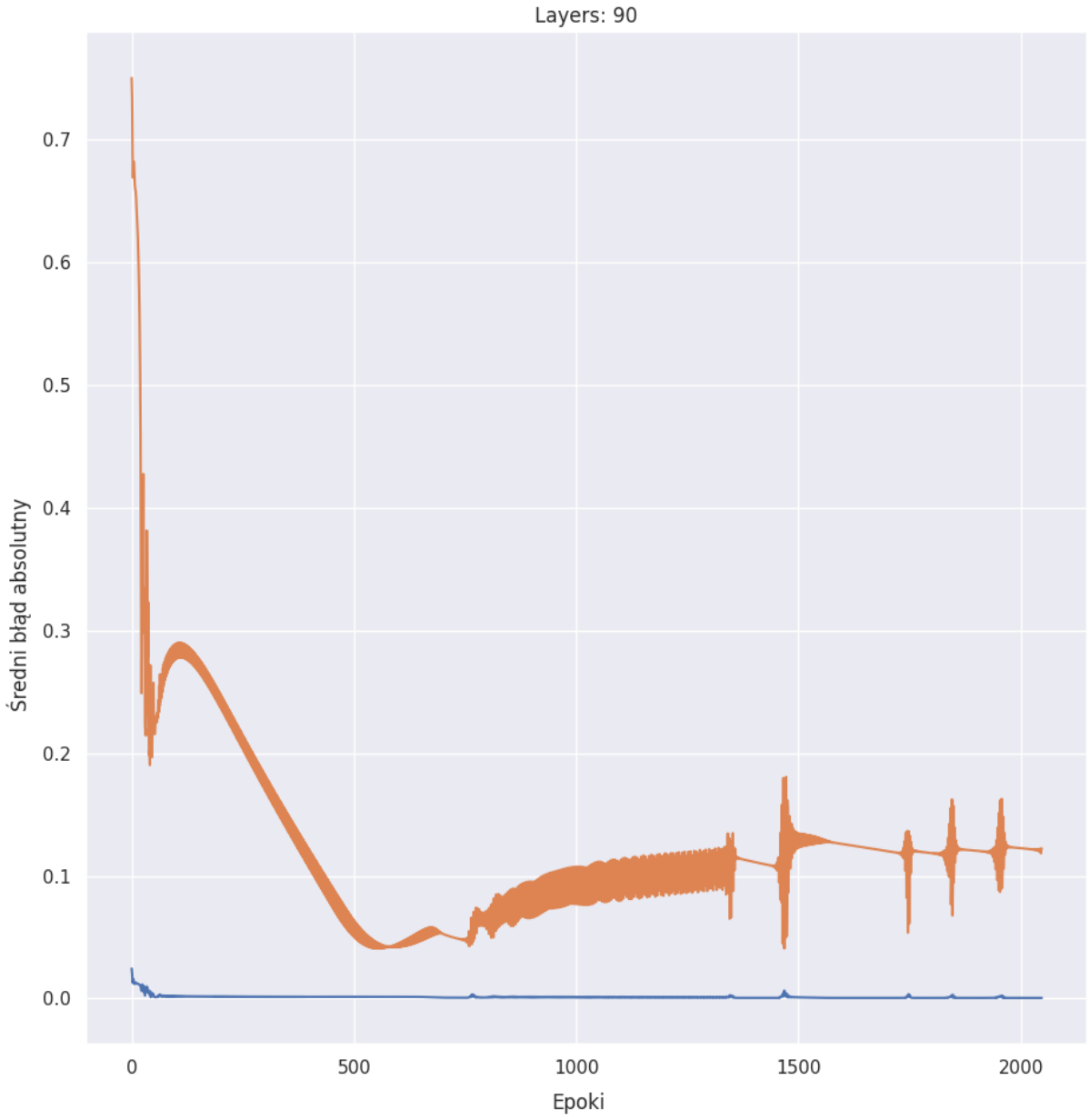
Nauka modeli przebiegała z zachowaniem podziału zbioru danych na odpowiednie 3 części. Na zbiór uczący składały się dane od 22.01.2020 do 16.09.2020, na zbiór walidacyjny dane od 17.09.2020 do 10.12.2020, a pozostałe dane tworzą zbiór testowy. Wyniki testów przedstawiono na następujących rysunkach. Kolor niebieski przedstawia wartość błędu na danych uczących, a kolor pomarańczowy na danych walidacyjnych.



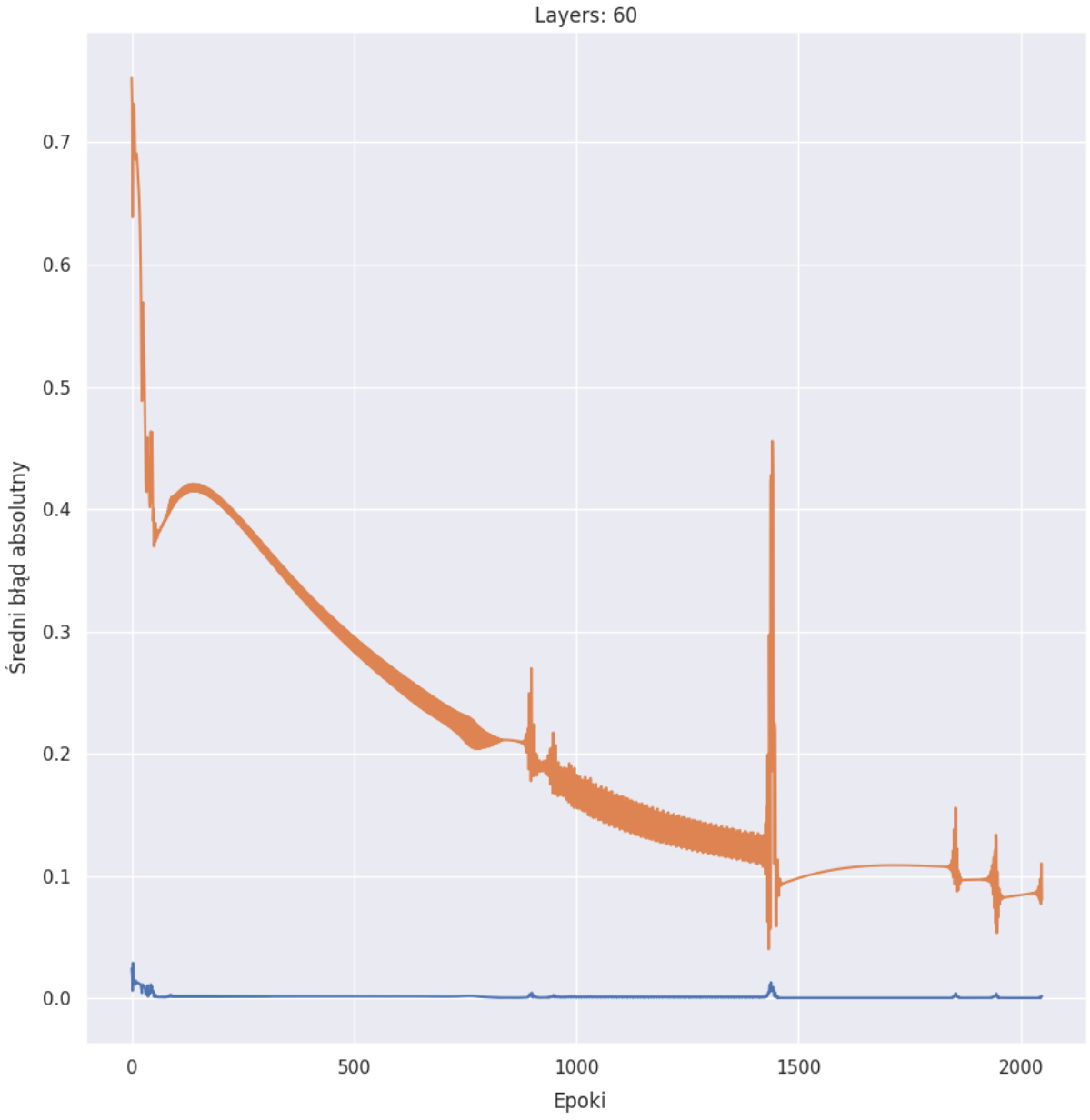
Rysunek 7: Błąd uczenia dla danych zagregowanych i wielkości warstwy równej 32.



Rysunek 8: Błąd uczenia dla danych zagregowanych i wielkości warstwy równej 60.

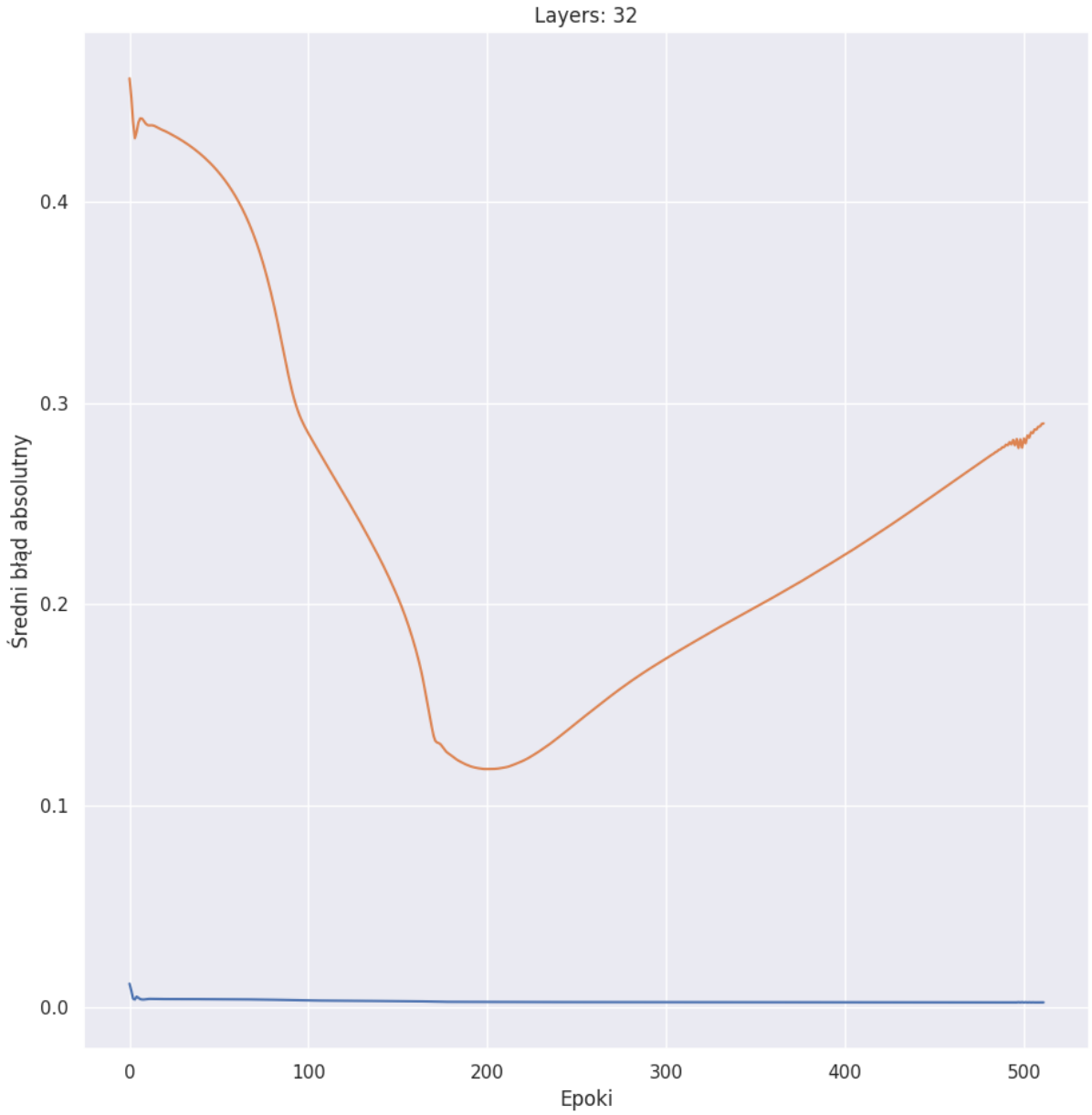


Rysunek 9: Błąd uczenia dla danych zagregowanych i wielkości warstwy równej 90.

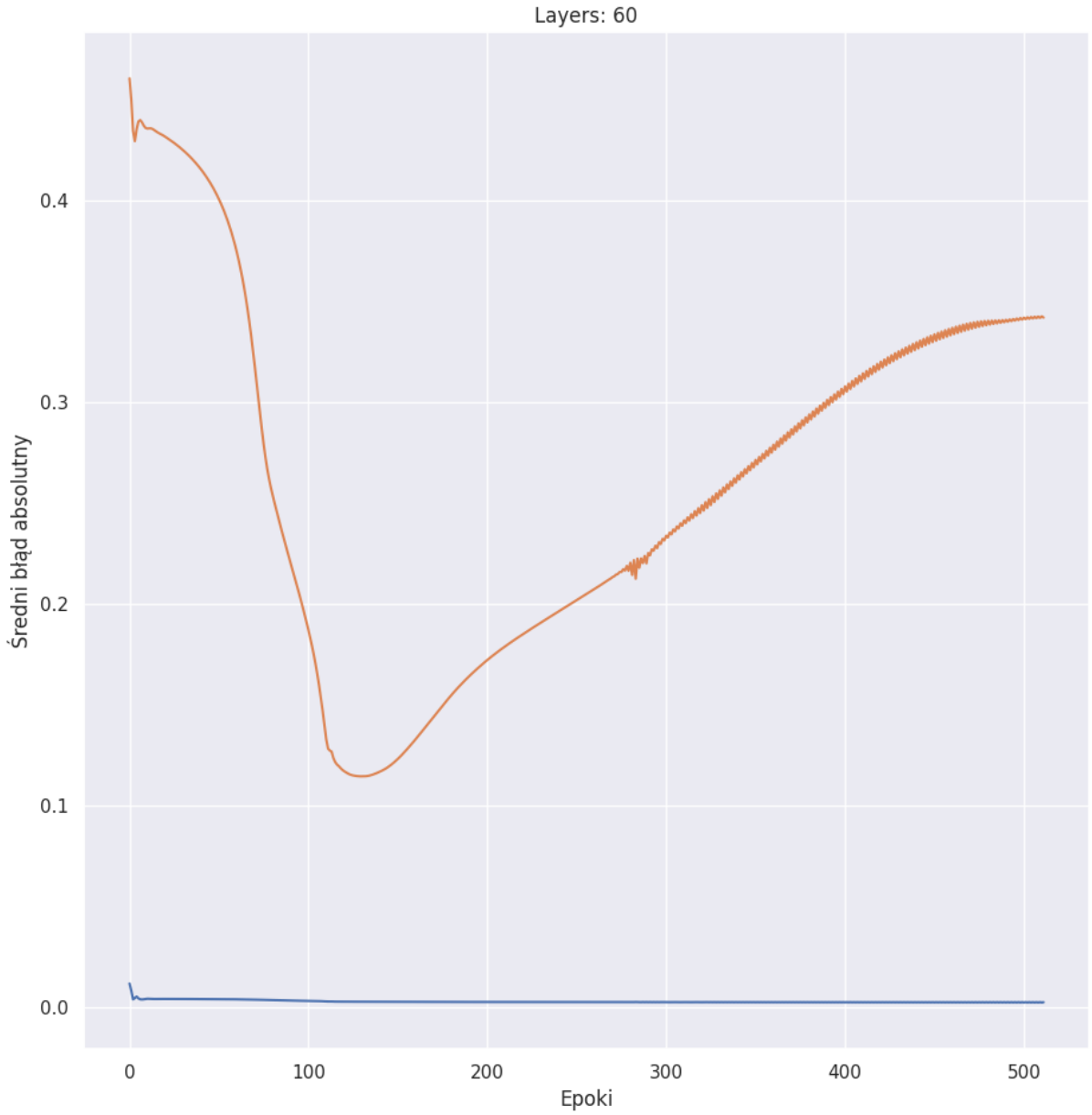


Rysunek 10: Błąd uczenia dla danych zagregowanych i wielkości warstw równej 60/120.

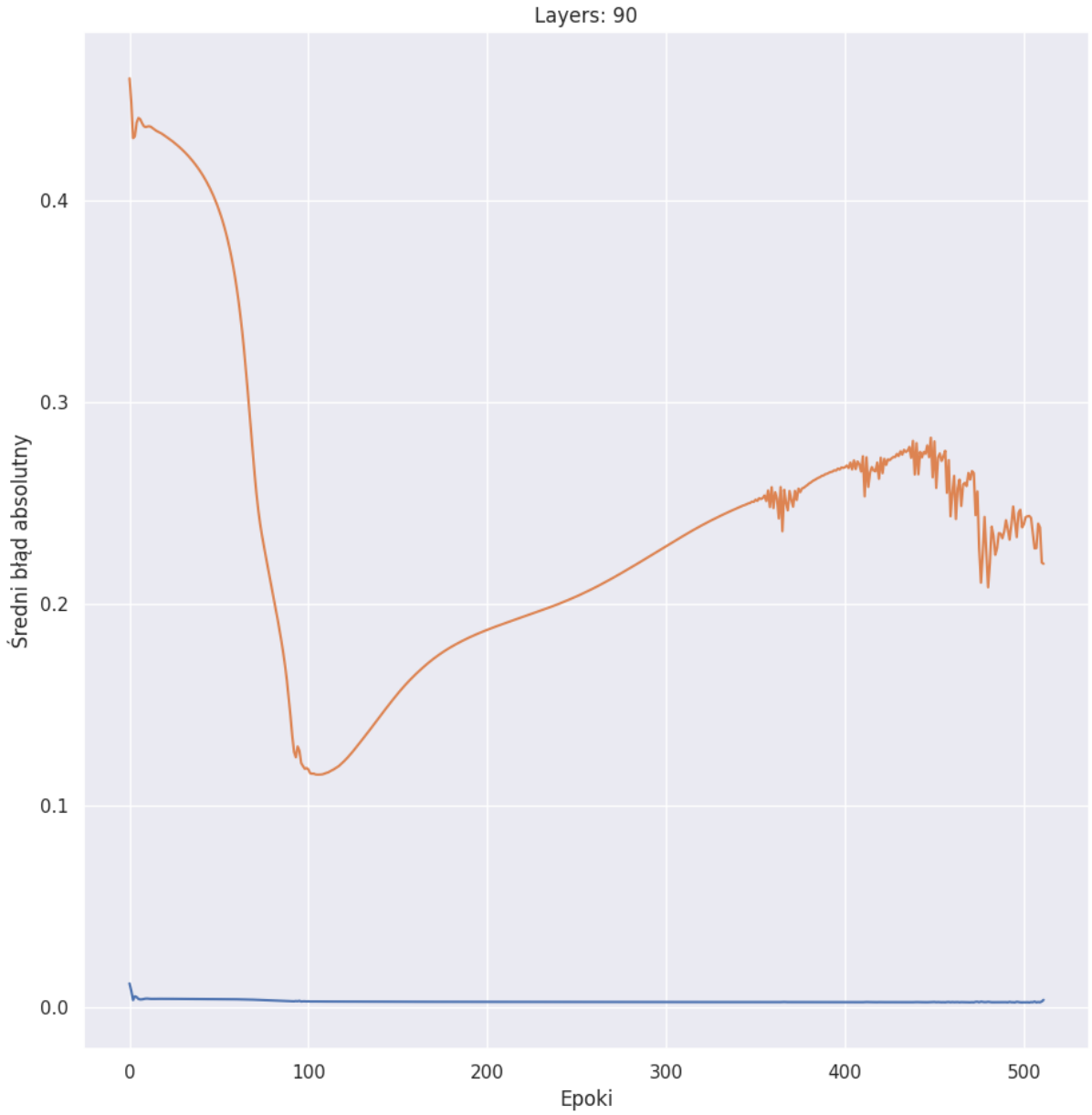
Na wszystkich wykresach widać bardzo szybki spadek wartości błędu na danych walidacyjnych, po którym następuje niewielkie pogorszenie i następnie znowu zmniejszanie wartości błędu. Najmniejszą wartość błędu osiągnięto dla 60 warstw.



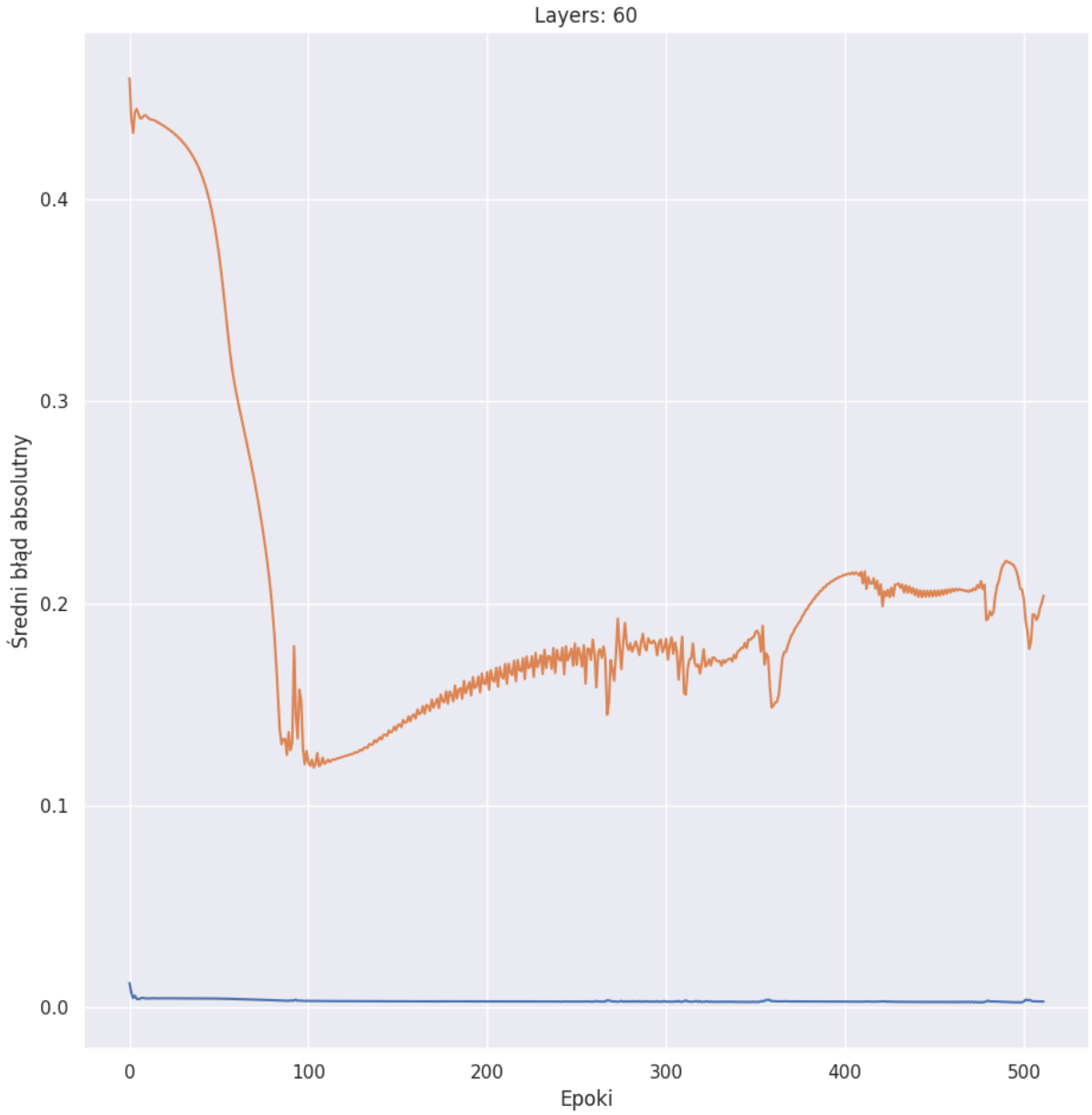
Rysunek 11: Błąd uczenia dla danych dziennych i wielkości warstwy równej 32.



Rysunek 12: Błąd uczenia dla danych dziennych i wielkości warstwy równej 60.

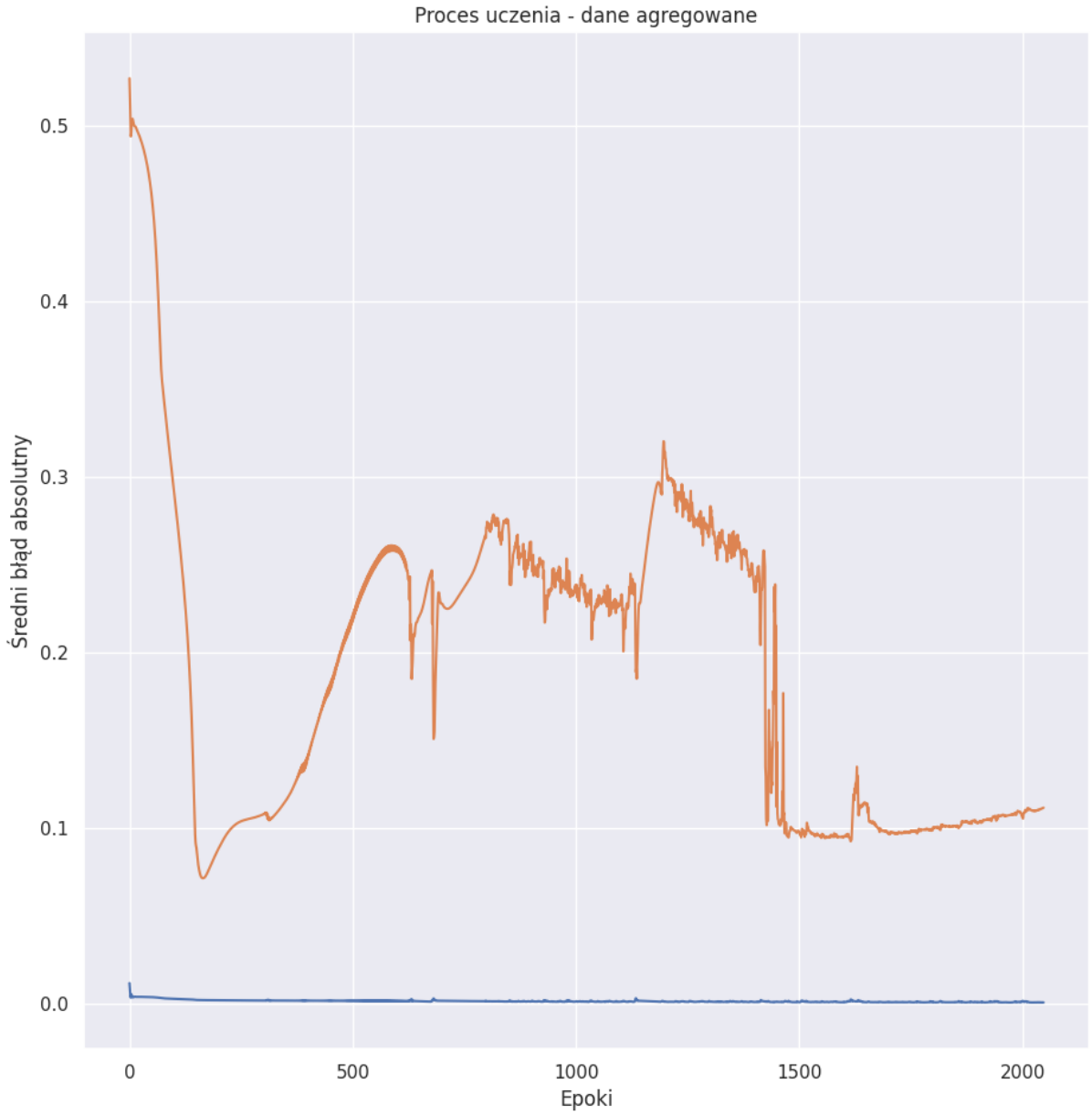


Rysunek 13: Błąd uczenia dla danych dziennych i wielkości warstwy równej 90.

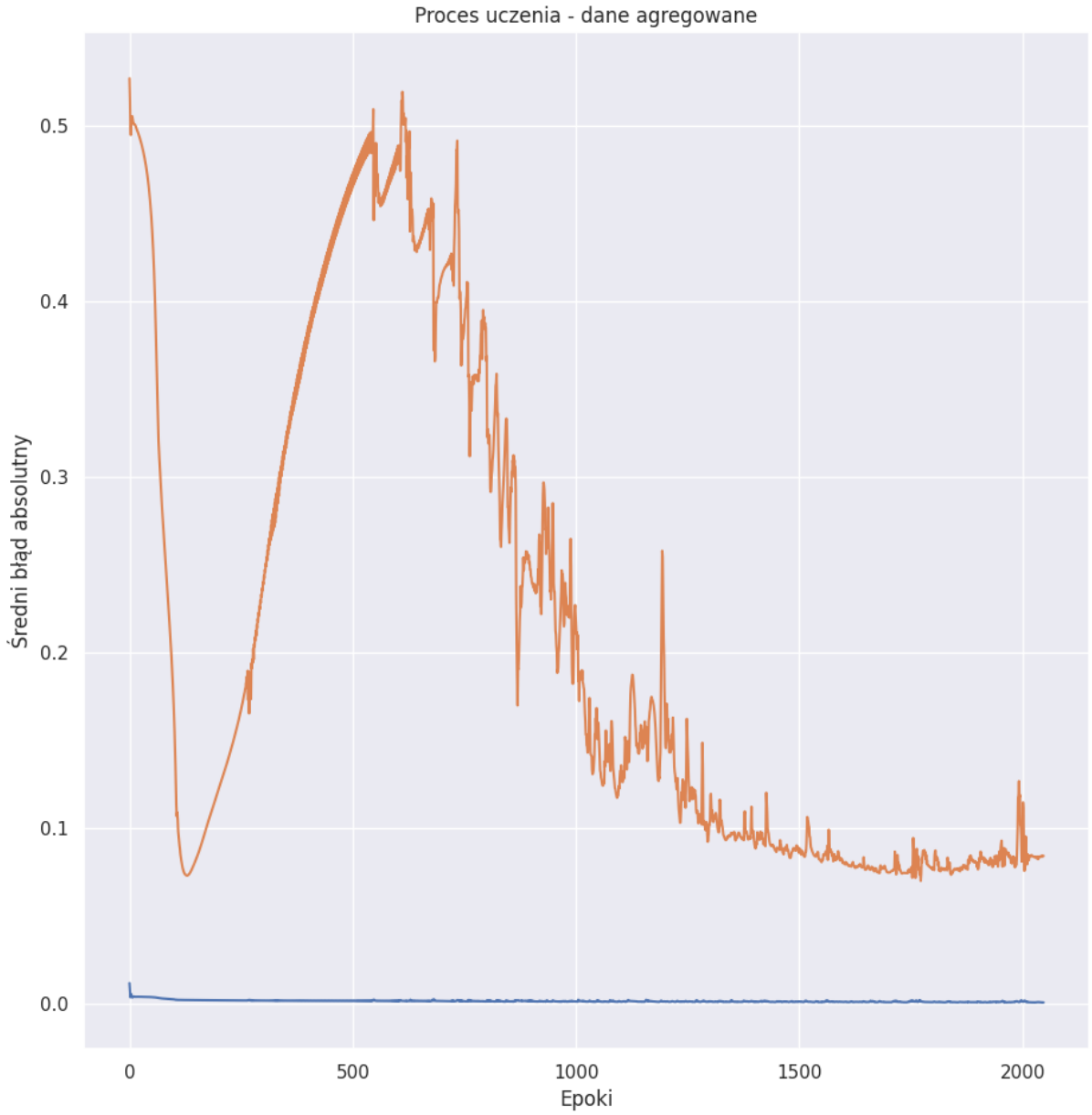


Rysunek 14: Błąd uczenia dla danych dziennych i wielkości warstw równej 60/120.

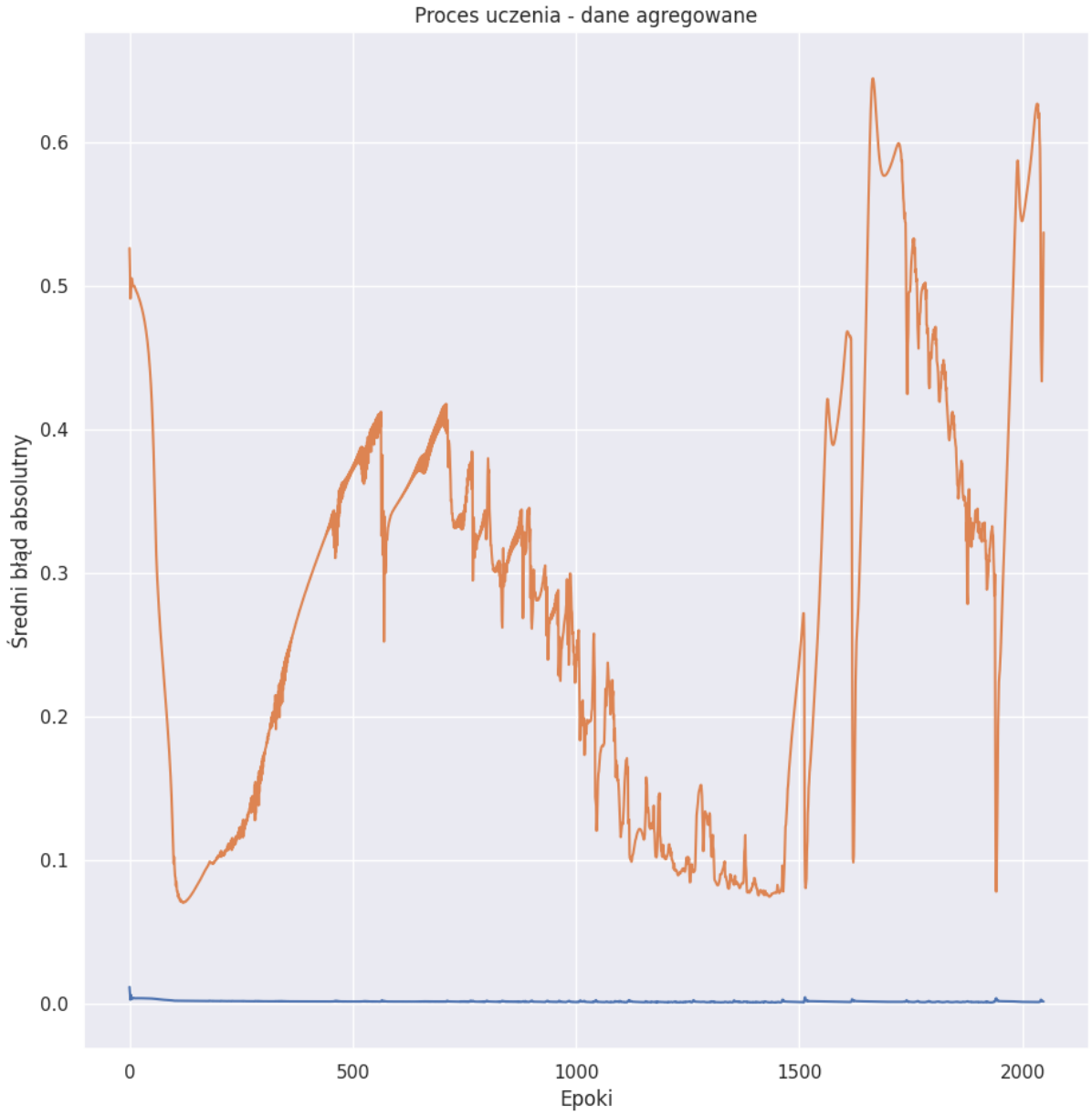
Na wszystkich wykresach widać bardzo szybki spadek wartości błędu na danych walidacyjnych, gdzie minimum osiągane jest po 100 do 200 epokach. Następnie następuje w miarę stabilne pogarszanie jakości modelu. Najmniejszą wartość błędu osiągnięto dla 60 warstw, jednak dokładność pozostałych modeli nie odstaje w sposób znaczący.



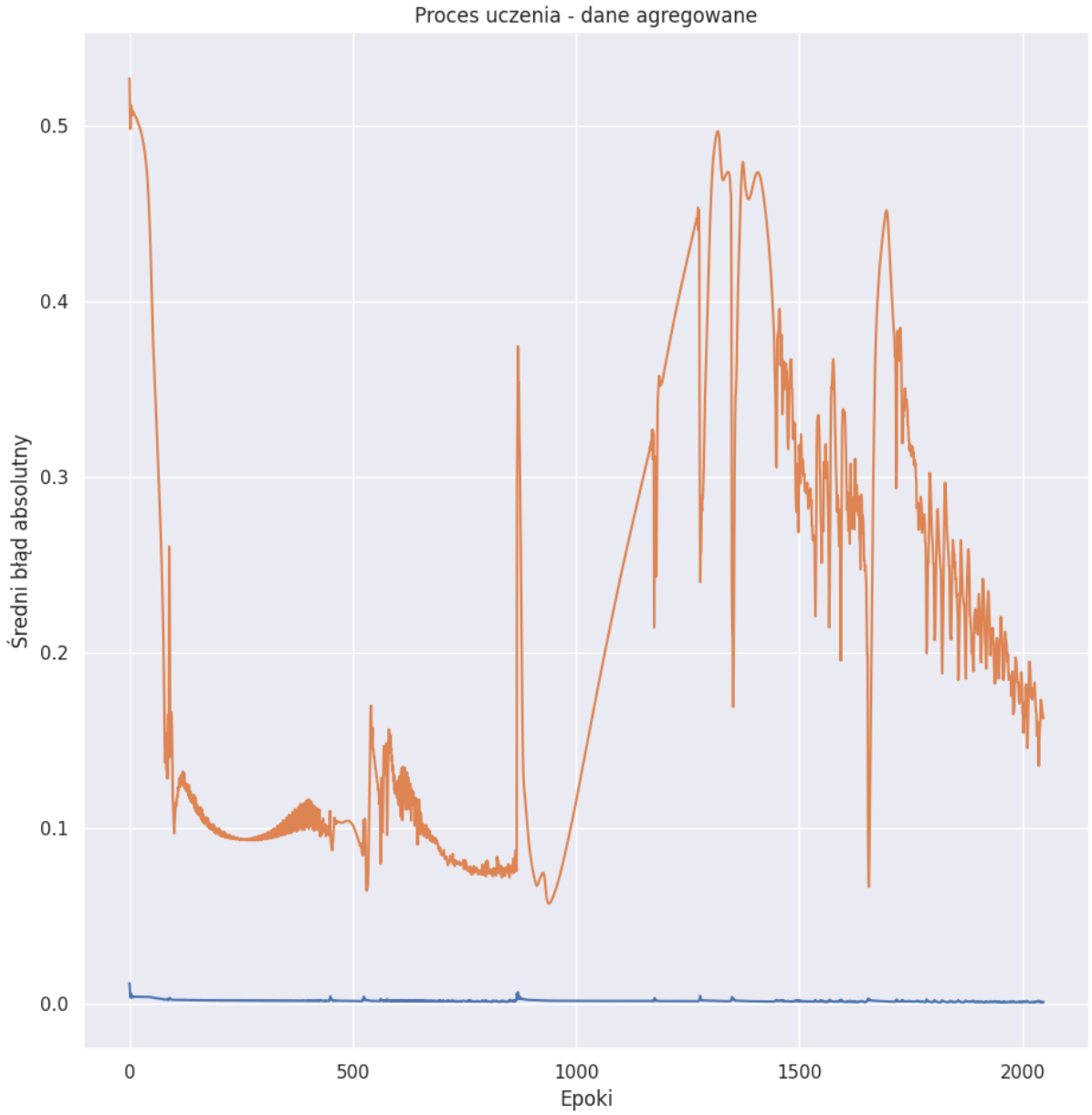
Rysunek 15: Błąd uczenia dla uśrednionych danych dziennych i wielkości warstwy równej 32.



Rysunek 16: Błąd uczenia dla uśrednionych danych dziennych i wielkości warstwy równej 60.



Rysunek 17: Błąd uczenia dla uśrednionych danych dziennych i wielkości warstwy równej 90.



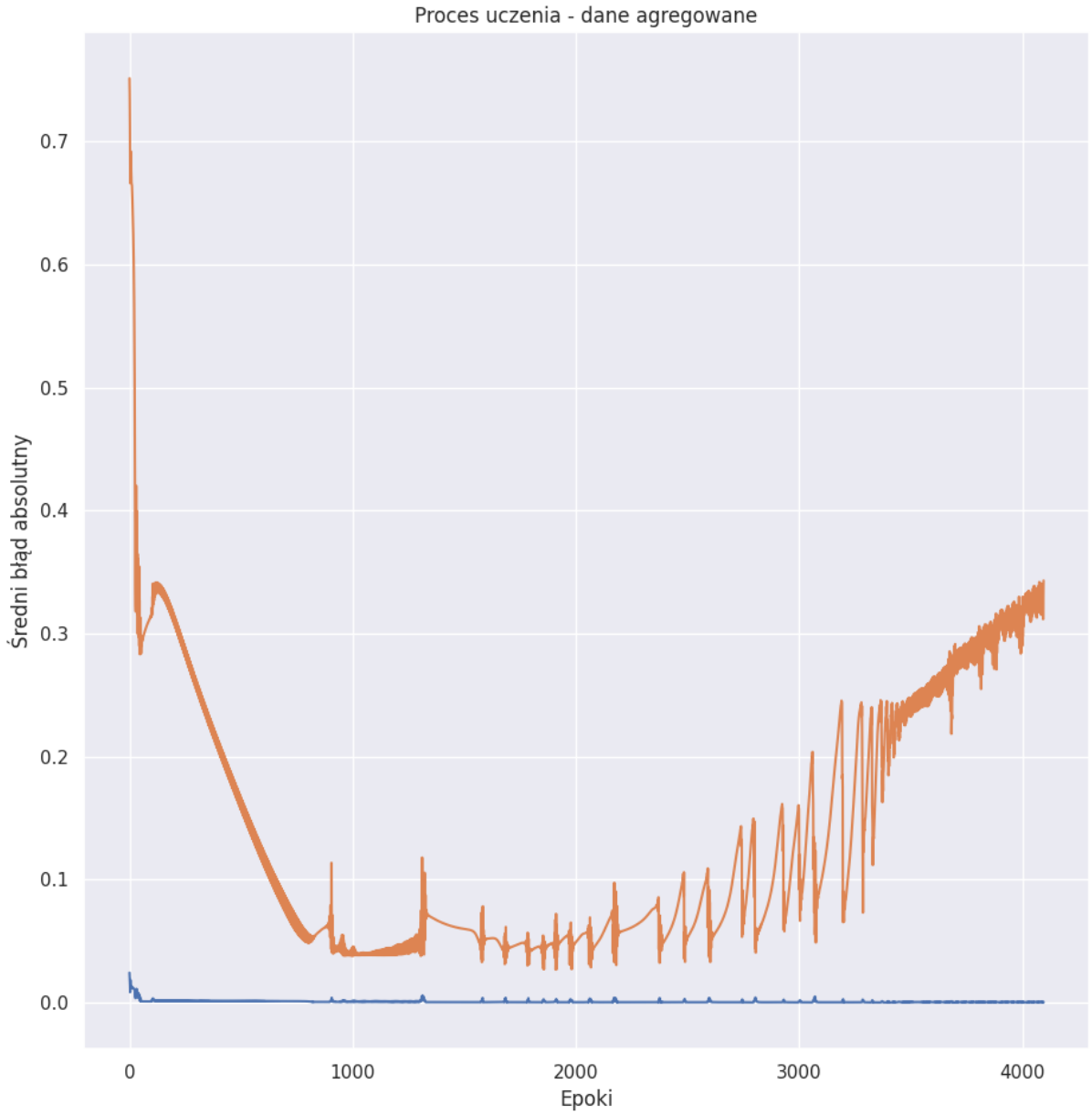
Rysunek 18: Błąd uczenia dla uśrednionych danych dziennych i wielkości warstw równej 60/120.

Na początku uczenia wartość błędu bardzo szybko spada i osiąga wartość minimalną lub jej bliską. W tym przypadku najlepszy okazał się być model z dwuwarstwowy z 60 i 120 jednostkami.

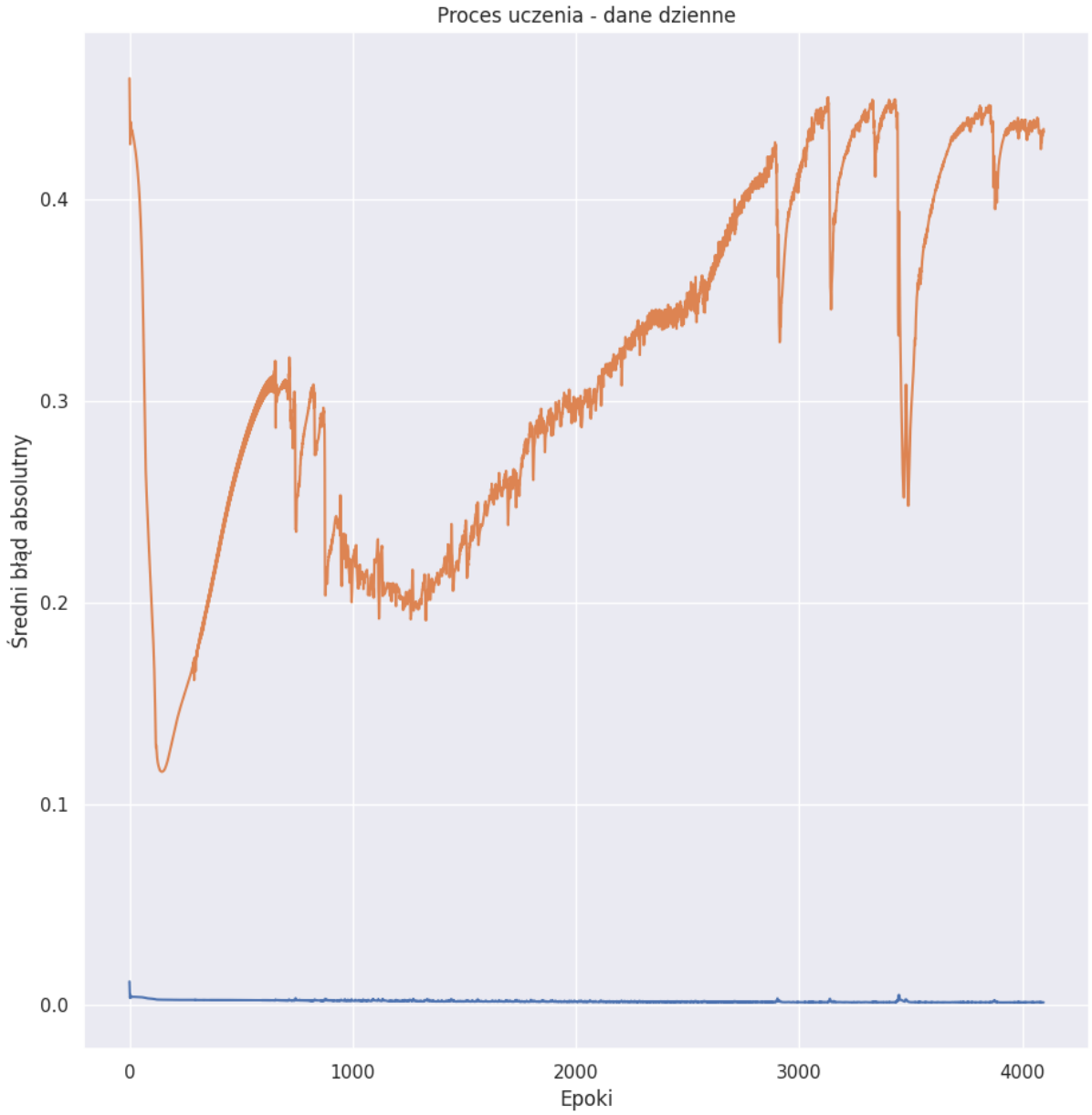
Najczęściej najlepsze rezultaty dawał model z 60 jednostkami, co równe jest ilości dni wejściowych do sieci neuronowej, i dlatego on został wybrany do dalszych prac.

4.3 Uczenie modelu

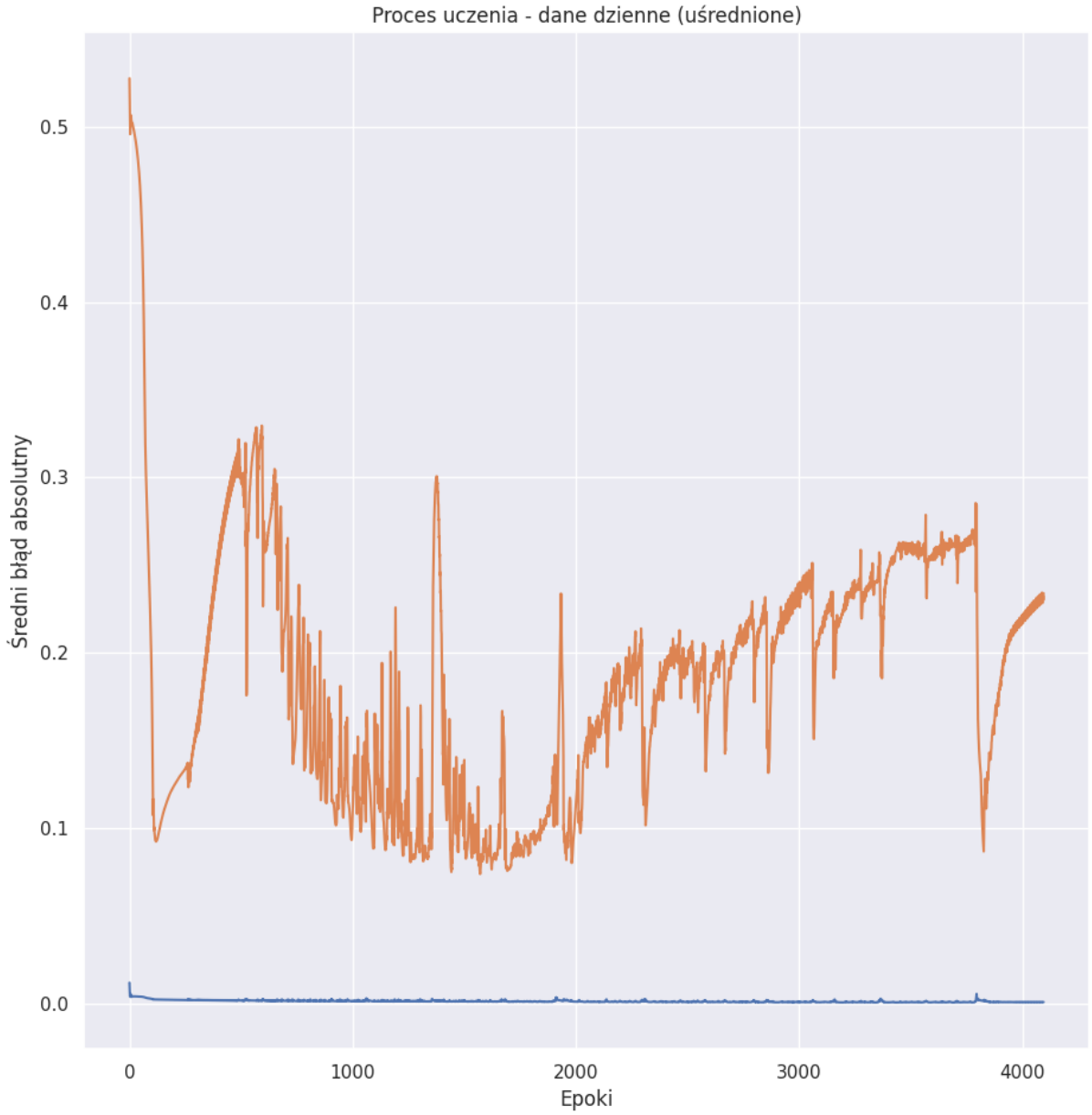
Modele sieci neuronowych wybrane w poprzednim podrozdziale należało nauczyć, sprawdzić ich zachowanie na większej liczbie iteracji oraz zapisać gotowe modele do plików w celu ich późniejszego wykorzystania. Dla danych dziennych i uśrednionych danych dziennych bardzo dobre dopasowanie jest już osiągane w pierwszych 200 epokach, a dla danych agregowanych jest to około 1000 epok. W dalszych epokach modele osiągają niewiele lepsze wartości błędu lub dochodzi do przeuczenia, dzięki czemu nie postanowiono zwiększać liczby epok.



Rysunek 19: Błąd uczenia dla danych agregowanych w 4096 epokach.



Rysunek 20: Błąd uczenia dla danych dziennych w 4096 epokach.



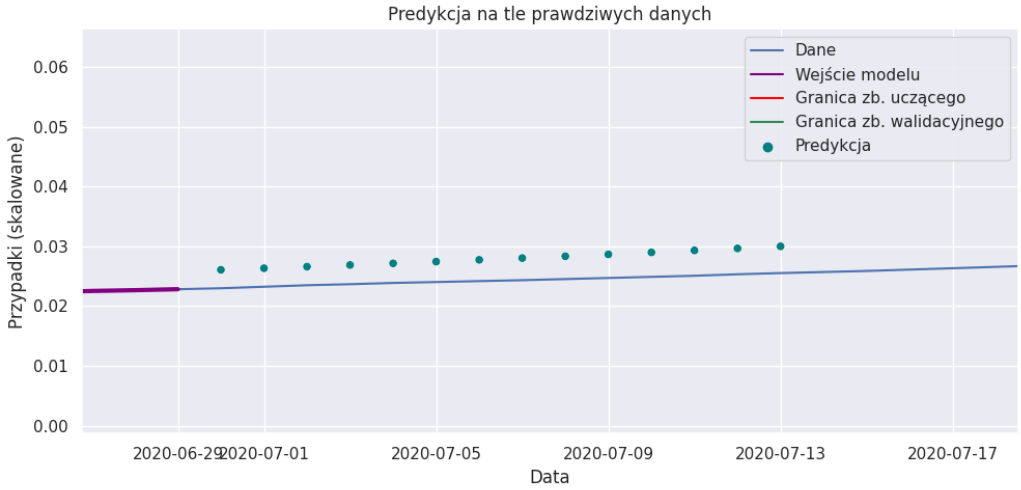
Rysunek 21: Błąd uczenia dla uśrednionych danych dziennych w 4096 epokach.

4.4 Ewaluacja modelu

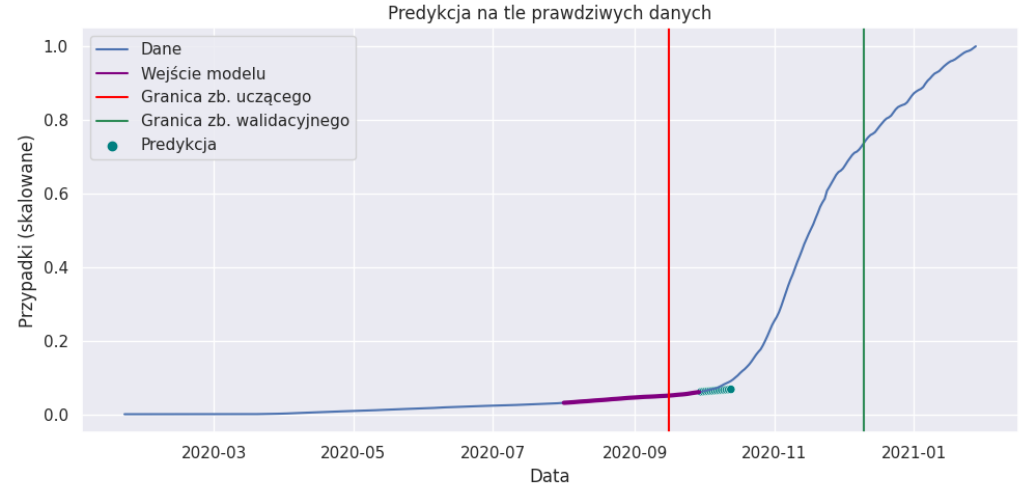
Ewaluacja modeli była wykonywana na 4 różnych zakresach dat, co daje łącznie 12 sprawdzeń działania modelu. Czerwona i zielona pionowa linia wskazują miejsca podziału danych na zbiory uczący, walidacyjny i testowy. Fioletowa linia oznacza wejścia modelu, a seledynowe kropki wartości przewidywane. Przedstawiono również zbliżenie na przewidywany fragment liczby zachorowań.

4.4.1 Wyniki dla danych agregowanych

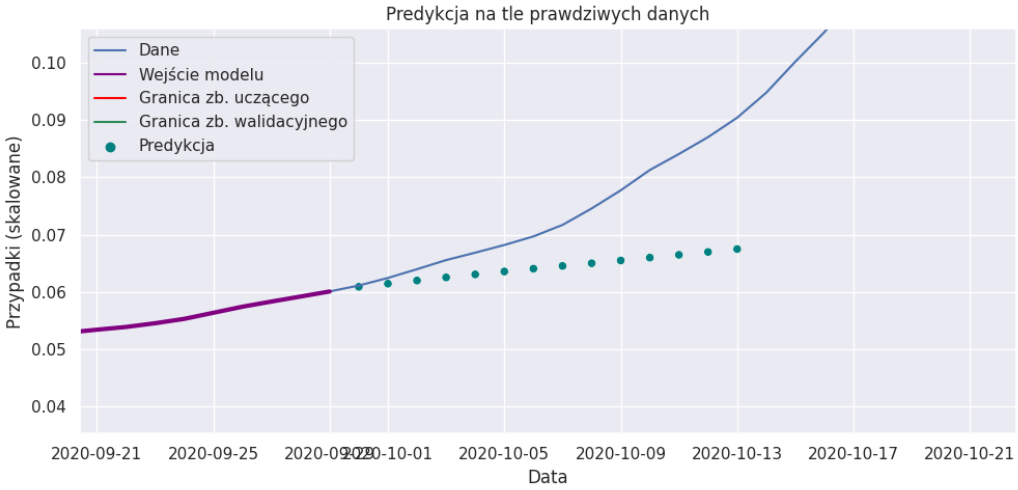
Rysunek 22: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



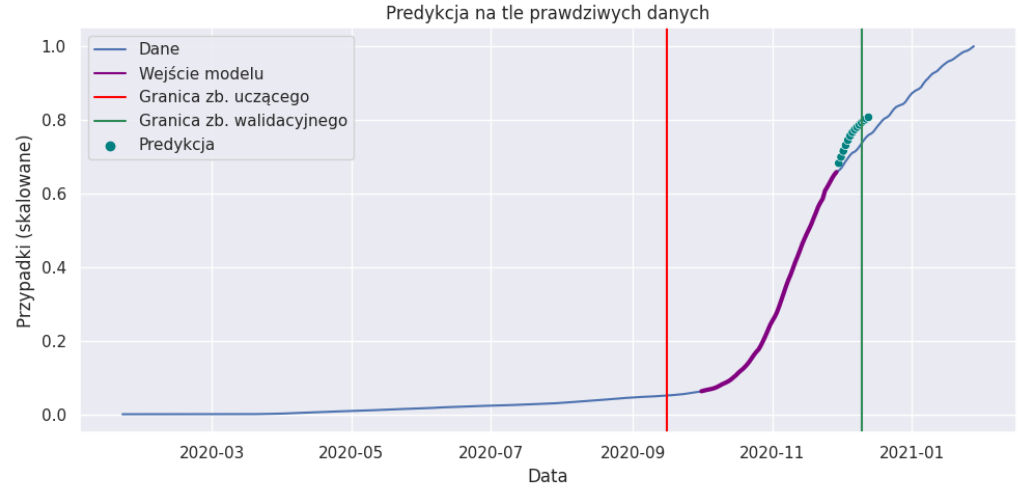
Rysunek 23: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).



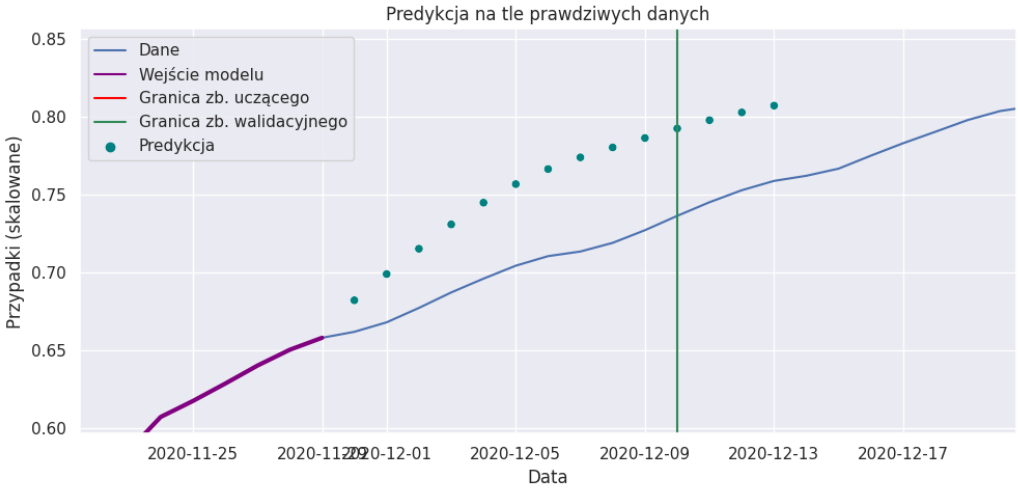
Rysunek 24: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



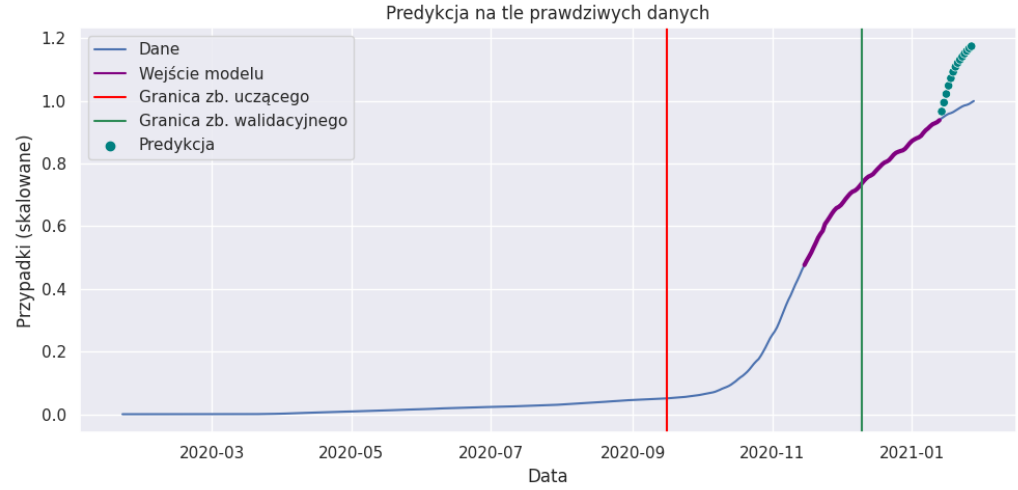
Rysunek 25: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).



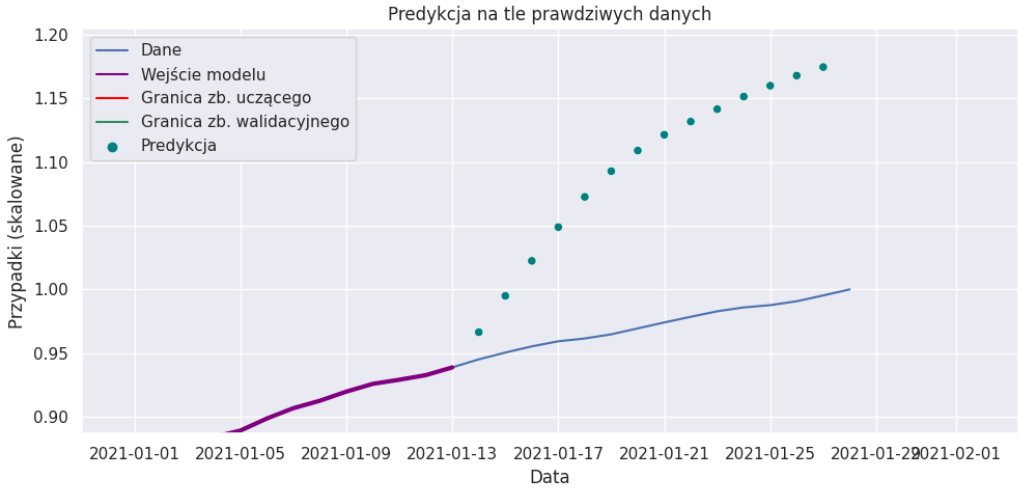
Rysunek 26: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



Rysunek 27: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

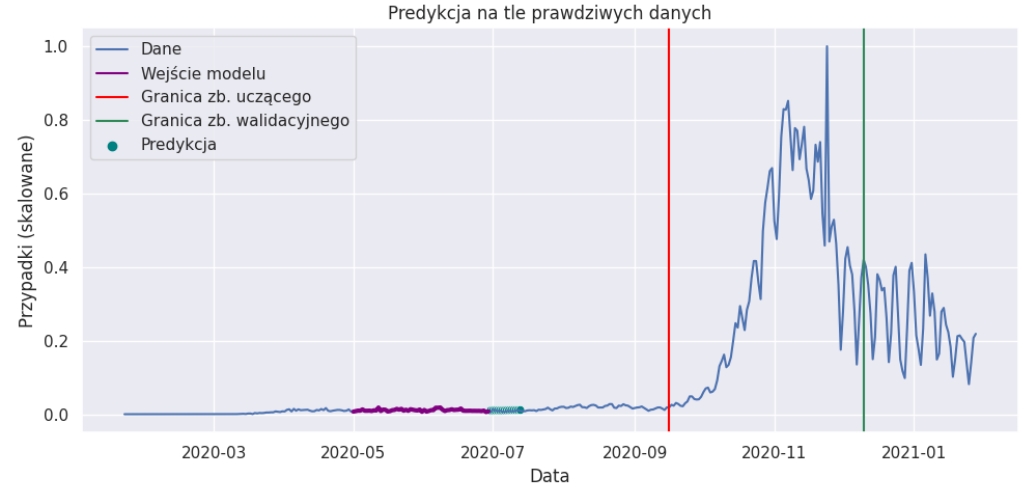


Rysunek 28: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.

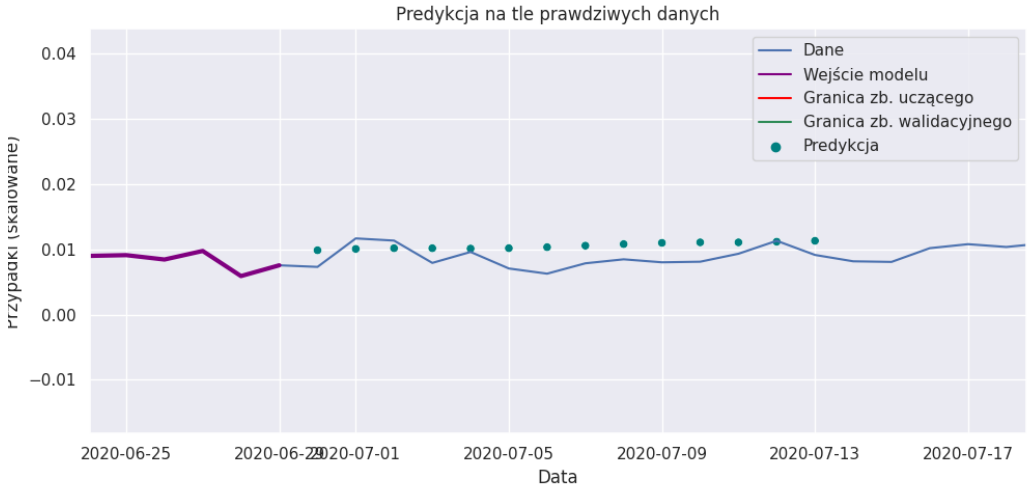


Rysunek 29: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

4.4.2 Wyniki dla danych dziennych

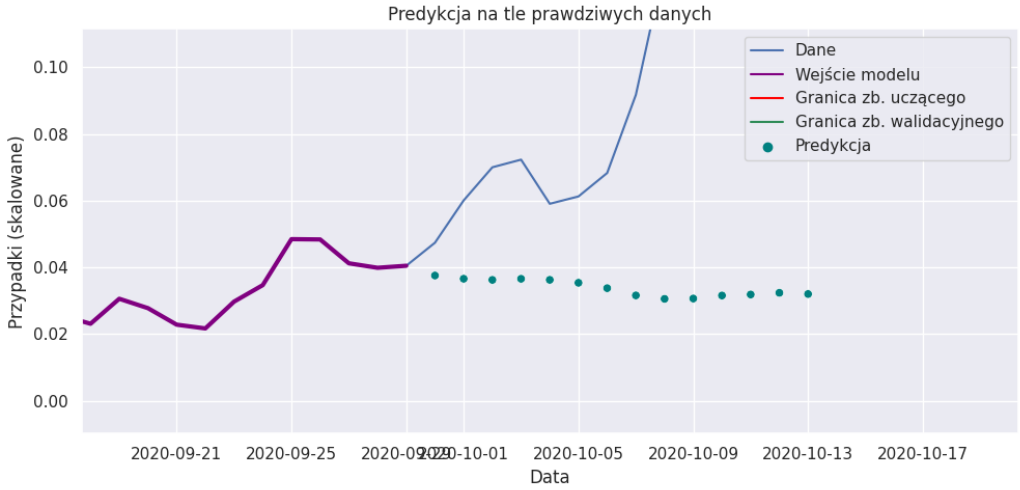


Rysunek 30: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



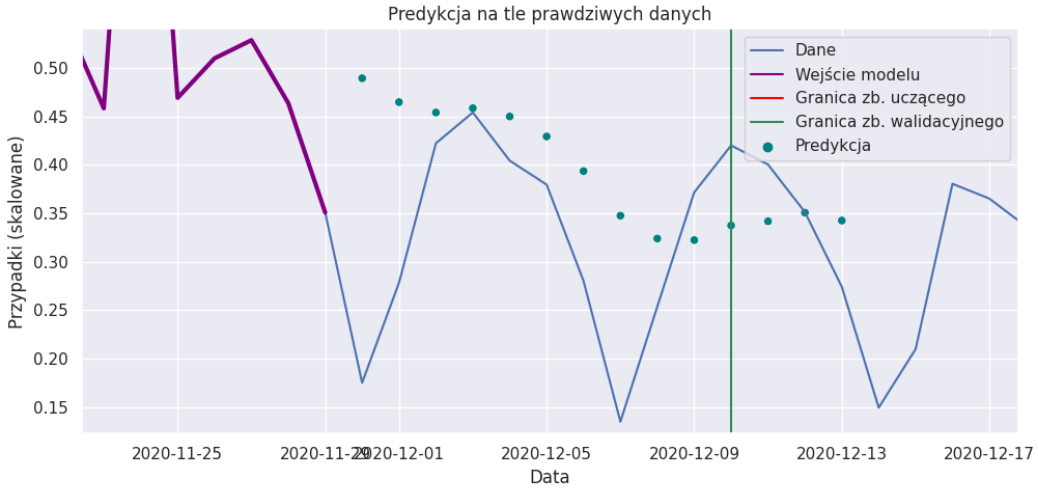
Rysunek 31: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

Rysunek 32: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



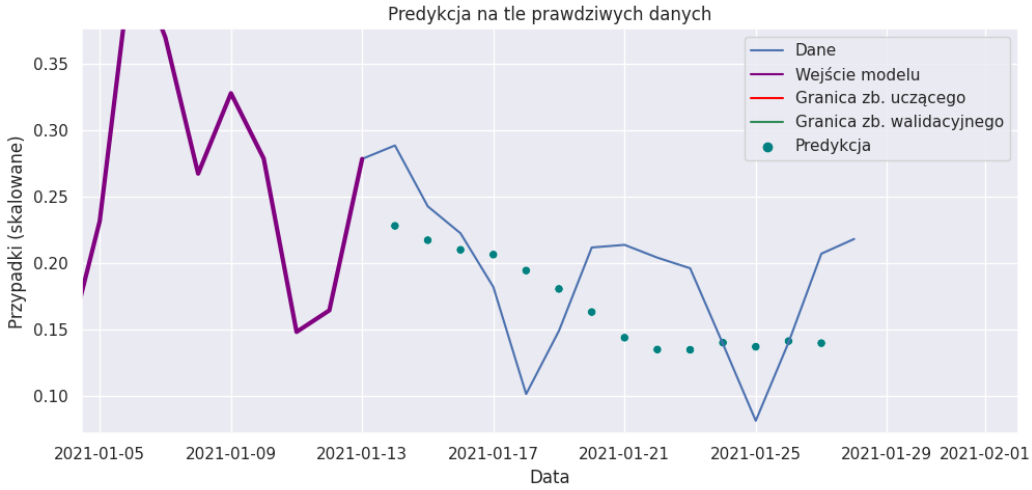
Rysunek 33: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

Rysunek 34: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



Rysunek 35: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

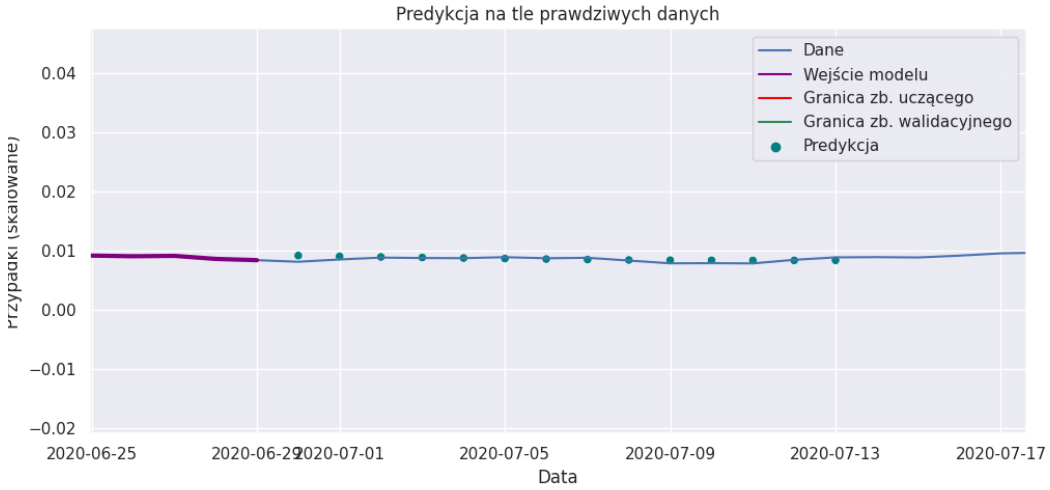
Rysunek 36: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



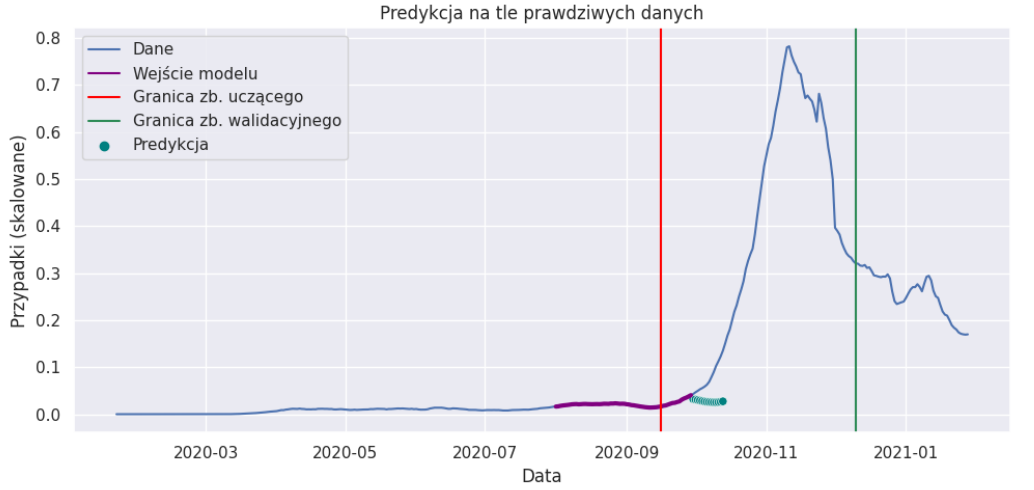
Rysunek 37: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

4.4.3 Wyniki dla uśrednionych danych dziennych

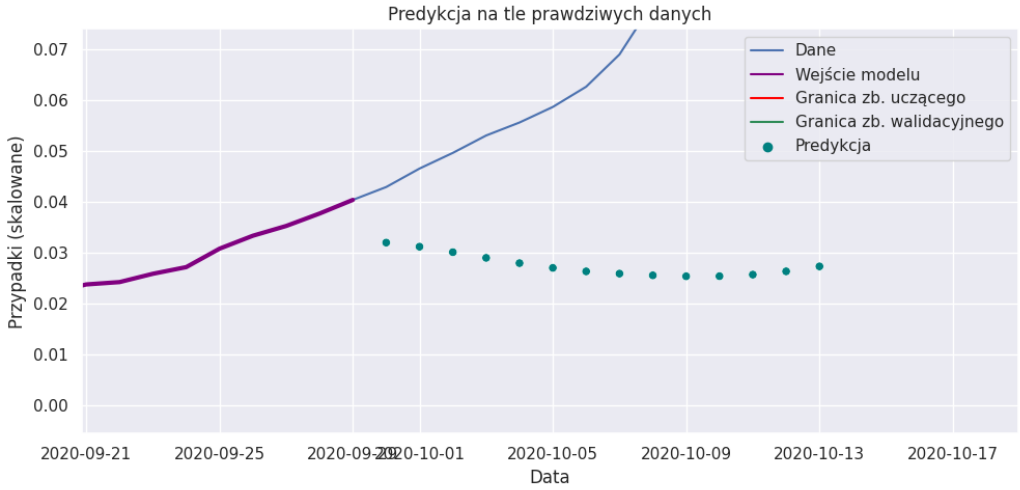
Rysunek 38: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



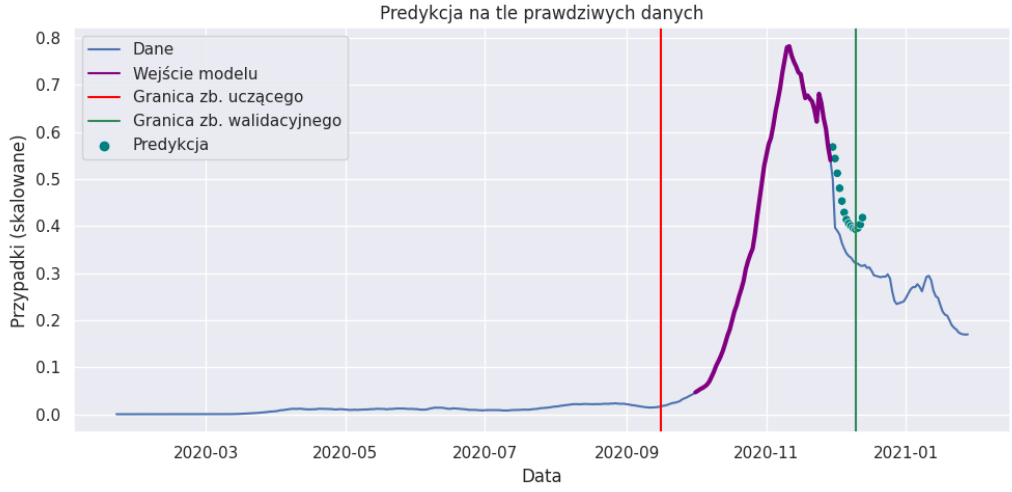
Rysunek 39: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).



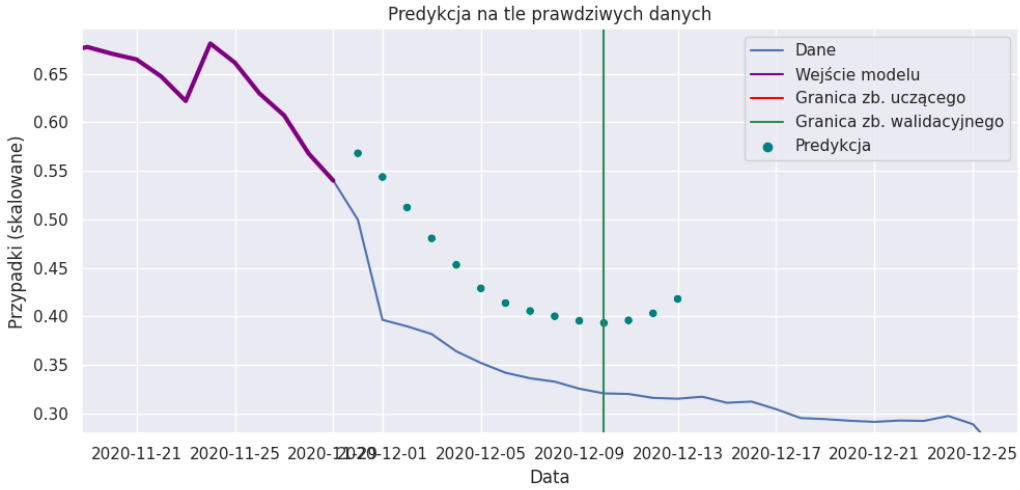
Rysunek 40: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



Rysunek 41: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

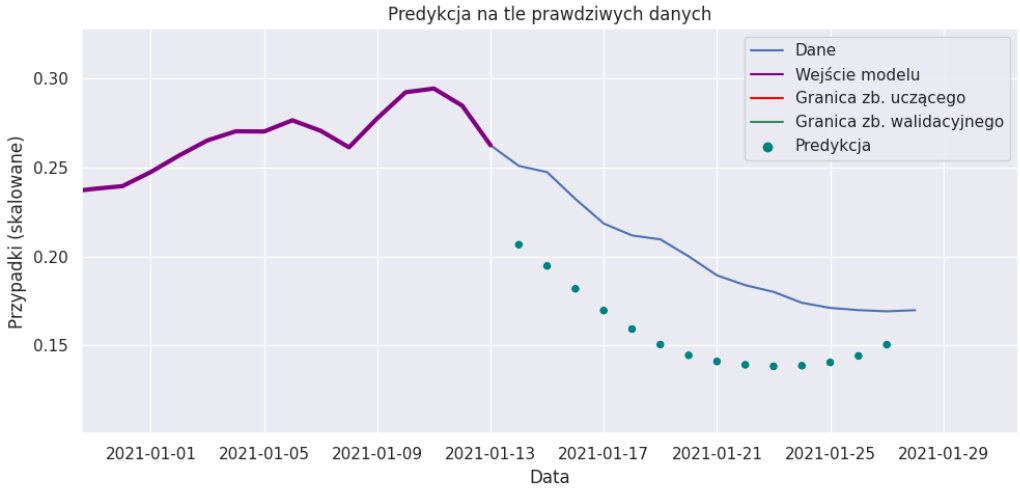


Rysunek 42: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



Rysunek 43: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

Rysunek 44: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych.



Rysunek 45: Porównanie modelu z danymi rzeczywistymi dla danych agregowanych (zbliżenie).

5 Analiza

5.1 Analiza dla danych akumulowanych

Dla zakresu dat znajdującego się w pełni w zbiorze uczącym predykcja jest bardzo dokładna z raczej stałą wartością błędu przewidywania. Taki stan rzeczy nie jest zaskoczeniem. Dla następnego zakresu przewidywanie jest już obarczone większym błędem. Liniowy fragment wejściowy wygenerował liniowe przewidywania z zachowaniem rosnącego trendu wartości. Dla trzeciego zakresu dat model w miarę poprawnie przewidział wartości oraz trend ich zmian. W danych rzeczywistych dochodzi do wypłaszczenia i można to też zauważyć w predykcji modelu. Dla ostatniego zakresu jedyną poprawną rzeczą jest przewidywanie wzrostu liczby zachorowań, jednak jest ona w znaczny sposób odstająca od danych rzeczywistych. W tym przypadku również widać wypłaszczenie zarówno w danych rzeczywistych jak i predykowanych wartościach.

Istotną rzeczą wpływającą na jakość modelu jest fakt, że zbiór uczący jest w znacznym stopniu płaski, gdzie następnie następuje znacznie szybszy wzrost wartości. Pomimo tego model okazał się dość dokładny dla trzeciego zakresu, jednak uzyskał znacznie błędne wyniki dla ostatniego. Zwiększenie

zbioru uczącego o zbiór walidacyjny mogłoby pomóc modelowi w takich sytuacjach.

5.2 Analiza dla danych dziennych

Dla zakresu dat znajdującego się w pełni w zbiorze uczącym predykcja jest bardzo dokładna, z zachowaniem trendu. Dla następnego zakresu przewidywanie jest już obarczone dużym błędem. W znacznym stopniu liniowy fragment wejściowy wygenerował liniowe przewidywania z niepoprawnym malejącym trendem. Dla trzeciego zakresu dat model poprawnie przewidział malejący trend wartości, jednak w większości przypadków predykowane wartości są większe niż wartości rzeczywiste. Negatywny wpływ na jakość przewidywań ma duża zmienność danych. Dla ostatniego zakresu dochodzi do podobnej sytuacji. Trend zmian jest poprawny jednak dokładne przewidywanie wartości nie jest możliwe.

Istotną rzeczą wpływającą na jakość modelu jest fakt, że zbiór uczący jest w znacznym stopniu płaski, gdzie następnie pojawiają się znacznie większe zmiany wartości i znacznie większe zaszumienie. Pomimo tego model okazał się dość dokładny, w większości przypadków poprawnie przedstawia trend zmian wartości.

5.3 Analiza dla uśrednionych danych dziennych

Ostatni z proponowanych modeli operuje na danych dziennych z 7-dniową średnią kroczącą. Dla zakresu dat znajdującego się w pełni w zbiorze uczącym predykcja jest bardzo dokładna. Dla drugiego zakresu można zaobserwować podobne zachowanie jak dla danych dziennych, czyli niepoprawnie malejące wartości, gdzie w rzeczywistości był to początek szybkiego wzrostu dziennej liczby zarażonych. Być może model tak reaguje na niewielki wzrostowy fragment występujący na wejściu. Dla trzeciego zakresu dat model bardzo dobrze przewidział trend malejący z wypłaszczeniem, jednak wartości są większe niż rzeczywiste i pojawia się niepoprawny wzrost w końcówce przewidywań. Dla ostatniego zakresu model zaniża wartości, jednak zachowuje poprawny trend. Jak w poprzednim przypadku pojawia się niepoprawny wzrost.

Istotną rzeczą wpływającą na jakość modelu jest fakt, że zbiór uczący jest w znacznym stopniu płaski, gdzie następnie pojawiają się znacznie większa zmienność danych. Pomimo tego model okazał się dość dokładny, w większości przypadków poprawnie przedstawia trend. Wyniki osiągnięte dzięki

zastosowaniu średniej ruchomej lepiej przybliżają się do wartości rzeczywistych dzięki zmniejszeniu zaszumienia zbioru wejściowego.

6 Wnioski

Przedstawione modele dość dobrze przedstawiają trend wartości, jednak nie radzą sobie z dokładniejszym ich przewidywaniem. We wszystkich przypadkach problemem może być zawartość zbioru uczącego, który ma stosunkowo małą zmienność względem późniejszych danych. Model uczony na większym zbiorze lub gdyby się skupić jedynie na danych do października powinien osiągać lepszą dokładność. W trakcie prac nie eksperymentowano z ilością dni wejścia i wyjścia, co stanowi możliwy dalszy kierunek prac.

Literatura

[1] Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases*, 20(5):533–534, 2020.

[2] Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[3] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[4] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 7:1, 2017.

[5] Alaa Sagheer and Mostafa Kotb. Time series forecasting of petroleum production using deep lstm recurrent networks. *Neurocomputing*, 323:203 – 213, 2019.

[6] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401. IEEE, 2018.