Optymalną architekturę dla moduły LSTM wybraliśmy porównując wartości MSELoss dla zbioru testowego. Najniższa wartość (uzyskana w dowolnym epochu) wskazywała na najlepszy model. Zmian dokonywaliśmy w parametrach hidden\_size i num\_layers LSTM z biblioteki pytorch. Wszystkie ustawiane przez nas parametry oraz z input\_size mogły być zmieniane w poszukiwaniu najlepszego modelu.

Ostatecznym wynikiem była poniższa klasa:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Model trenowany był wg algorytmu:

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznieZapobiegliśmy przeuczeniu poprzez zachowanie jedynie modelu o najniższym mean square error.