KIERUNEK STUDIÓW

Informatyka w Biznesie

Tryb i stopień studiów: niestacjonarny – magisterskie N2

Damian Kaliciak 189711

Piotr Michalak 189791

Maciej Nalepa 189794

Nazwa przedmiotu:

Sztuczna Inteligencja i Machine Learning

Rok akademicki 2023/2024 (semestr letni)

1. Wstęp

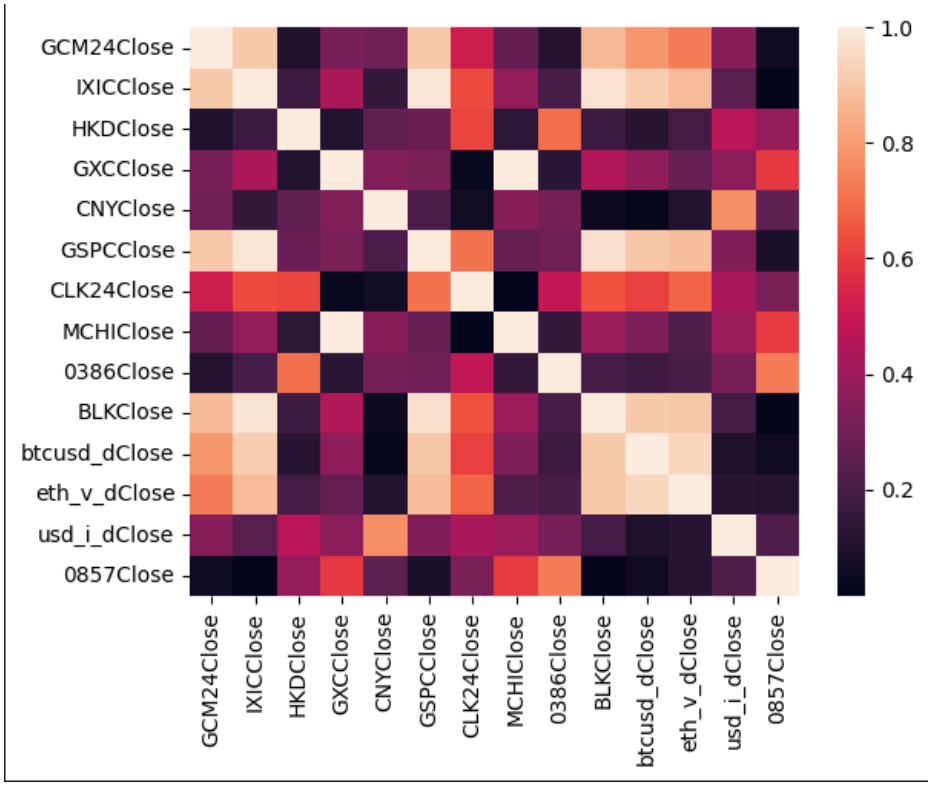
Celem projektu jest opracowanie modelu predykcyjnego Long-Short Term Memory (LSTM), który oszacuje za pomocą regresji liniowej wartość kursu BTCUSD na podstawie wprowadzonych danych wejściowych z innych instrumentów.

Następnie, model zostanie porównany z predykcją średniej kroczącej, a także zmodyfikowany do działania wyłącznie na podstawie kursu BTCUSD. Metryki wykorzystane do oceny jakości to Mean Squared Error (MSE) oraz Mean Absolute Error (MAE).

Ostatnim etapem projektu jest wprowadzenie elementu generalizacji za pomocą innych danych dotyczących. Nowymi danymi wejściowymi i wyjściowymi będzie kurs NASDAQ.

2. Przygotowanie danych

- Załadowano dane z wykorzystaniem klasy Featureset będącej elementem pomocniczej biblioteki projektu.
- Wczytane dane zostały zapisane w obiekcie DataFrame biblioteki Pandas, który jest strukturą danych umożliwiającą łatwą manipulację i analizę danych tabelarycznych.
- Sprawdzono typy danych, wszystkie okazały się być typu float64, co jest wymagane dla dalszej analizy. Dodatkowo, sprawdzono, czy w danych występują braki danych (NaN). W tym przypadku brakowało danych, co oznacza, że zbiór był kompletny i gotowy do dalszej analizy.
- Usunięto kolumny zawierające informacje o cenach otwarcia (Open), najwyższych (High) i najniższych (Low), pozostawiając tylko ceny zamknięcia (Close). Decyzja ta wynikała z założenia, że cena zamknięcia jest najbardziej reprezentatywna dla danego okresu i ma największe znaczenie dla predykcji przyszłych wartości.
- Przeprowadzono analizę korelacji między ceną zamknięcia Bitcoina a pozostałymi cechami. Celem było zidentyfikowanie cech, które mają najsilniejszy związek z ceną Bitcoina.

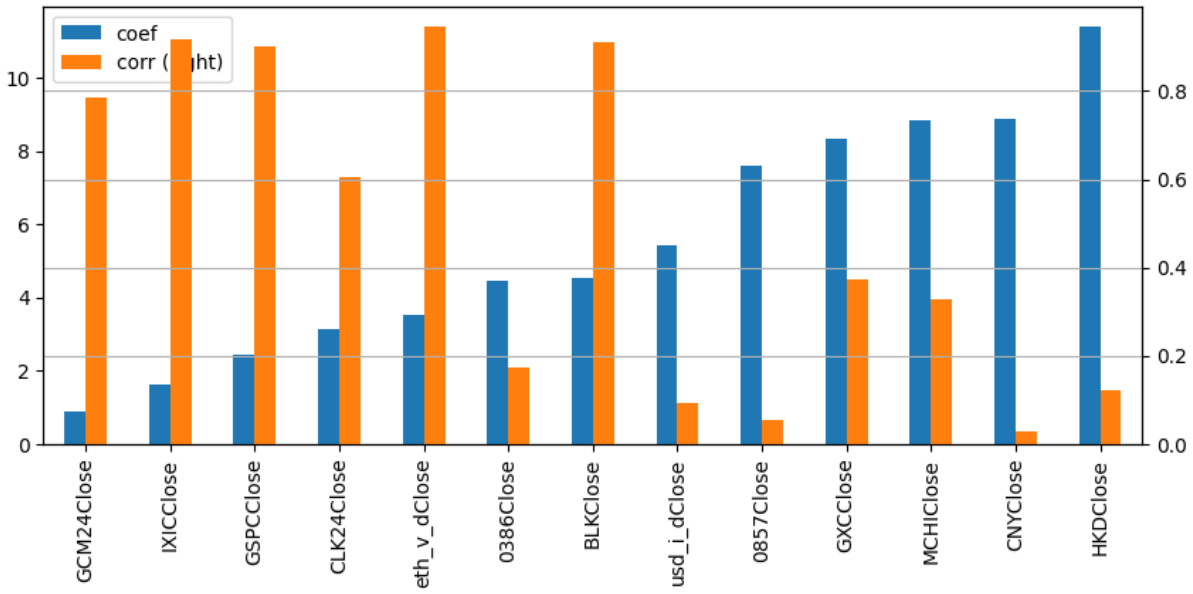


Rys. 1 Mapa cieplna macierzy korelacji.

```
Wartości korelacji

btcusd_dClose 1.000000 eth_v_dClose 0.945606 IXICClose 0.916503 BLKClose
0.912576 GSPCClose 0.902721 GCM24Close 0.786737 CLK24Close 0.605982 GXCClose
0.372193 MCHIClose 0.327378 0386Close 0.173504 Name: btcusd_dClose, dtype:
float64
```

- Na podstawie macierzy korelacji oraz współczynników regresji liniowej wybrano cechy o największym wpływie na cenę zamknięcia Bitcoina. Wybór ten opierał się na założeniu, że cechy o silnej korelacji z ceną Bitcoina będą najbardziej przydatne w procesie uczenia modelu. Natomiast współczynniki regresji liniowej pozwoliły wskazać cechy, które mimo niskiej korelacji pozwalają na skuteczne oszacowanie właściwiej wartości kursu Bitcoin.



Rys. 2 Selekcja danych wejściowych. wykres słupkowy, który wizualizuje istotność cech (współczynniki cech) oraz ich korelację z zmienną docelową btcusd_dClose.

- Dane zostały podzielone na dwa zbiory: uczący i testowy. Zbiór uczący służy do trenowania modelu, natomiast zbiór testowy do oceny jego skuteczności na danych, które nie zostały jeszcze przekazane do modelu. Miało to na celu zapobieżeniu sytuacji przeuczenia modelu, a więc sytuacji, w której model zapamiętuje dane uczące (zamiast wykorzystania próbki do ekstrapolacji / analizy innych danych).

Liczba trenowalnych parametrów powinna mieć podobny rząd wartości co rozmiar zbioru uczącego, aby uniknąć sytuacji, że model swoją liczbą parametrów jest w stanie opisać każdą próbkę uczącą i "zapamiętać" ją.

```
Total params: 6,309

Trainable params: 6,309

Non-trainable params: 0
```

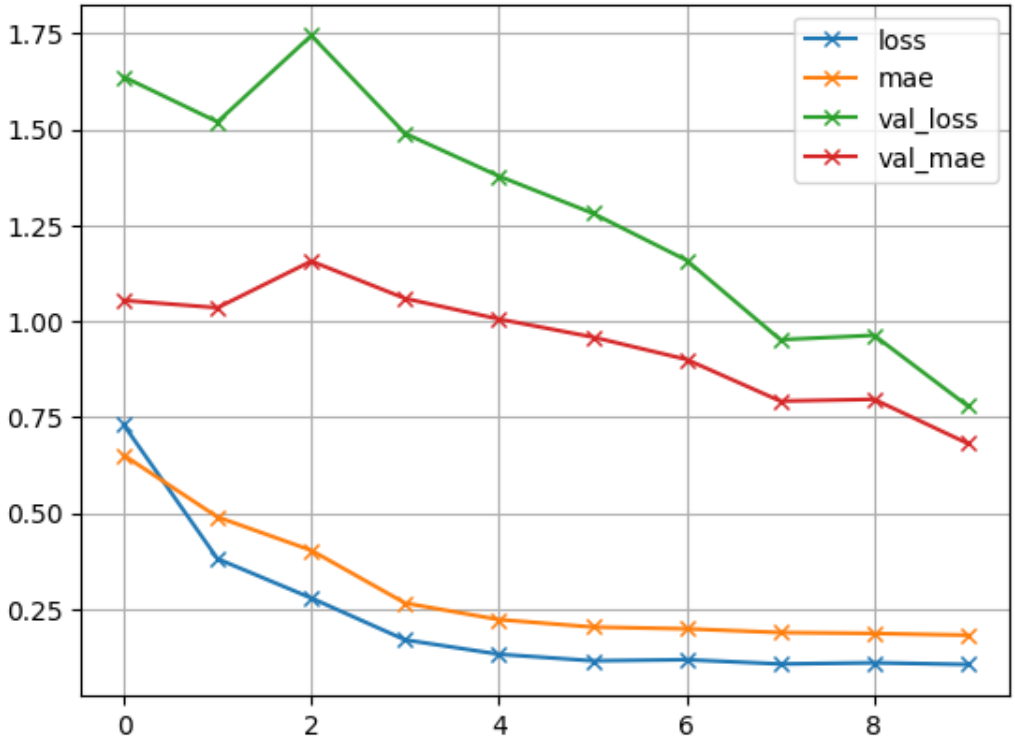
- Dane zostały przekształcone do formatu odpowiedniego dla modelu LSTM. Model ten wymaga danych wejściowych w postaci sekwencji (okien czasowych) o stałej długości.
- Każde okno czasowe zawierało wartości wybranych cech z określonej liczby poprzednich dni.
- Wartości docelowe (ceny zamknięcia Bitcoina) zostały przesunięte o jeden dzień do przodu, aby model uczył się przewidywać przyszłe wartości na podstawie danych historycznych.

3. Budowa i trenowanie modelu LSTM

- Zbudowano model LSTM z wykorzystaniem biblioteki Keras. Jego definicję umiesz siml_model, określając przy tym liczbę dni uwzględnianych wstecz oraz liczbę cech wejściowych. Model składał się z warstw LSTM (przewidywanie przyszłości), Dropout (zapobieżenie przeuczenia się modelu poprzez losowe

wyłączenie neuronów modelu) i Dense (przetwarzanie i wygenerowanie przewidywanych wyników).

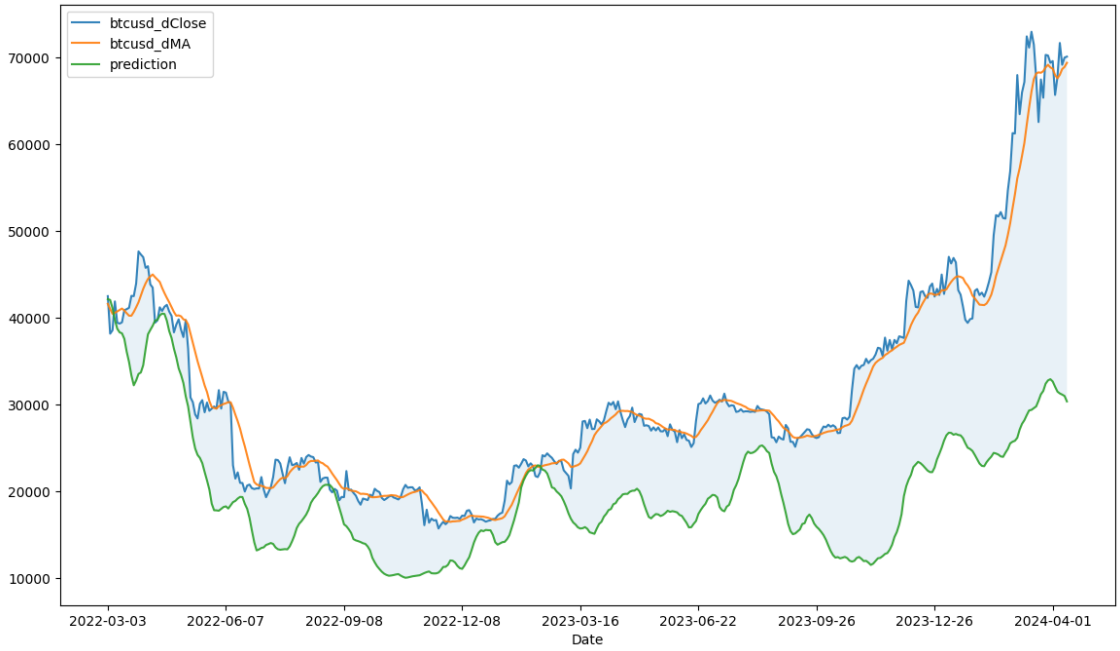
- Przeprowadzono trening modelu na zbiorze uczącym z wykorzystaniem model.fit i argumentami:
  - X_train: zbiór uczący zawierający dane wejściowe.
  - Y_train: zbiór uczący zawierający wartości docelowe.
  - validation_data: zbiór testowy (X_test, Y_test) używany do oceny modelu podczas treningu.
  - epochs: liczbę epok, czyli iteracji przez cały zbiór uczący.
  - batch_size: rozmiar partii, czyli liczba próbek przetwarzanych jednocześnie podczas jednej iteracji.
- Obserwowano **spadek** wartości funkcji straty (MSE) oraz błędu średniego bezwzględnego (MAE) zarówno na zbiorze uczącym, jak i testowym. **W modelu uczącym się prawidłowo, obie wartości powinny maleć dla każdej epoki.**



Rys. 3 Wyniki trenowania. Historia wartości MSE (loss) oraz dodatkowej metryki MAE na zbiorze trenującym i testowym (val).

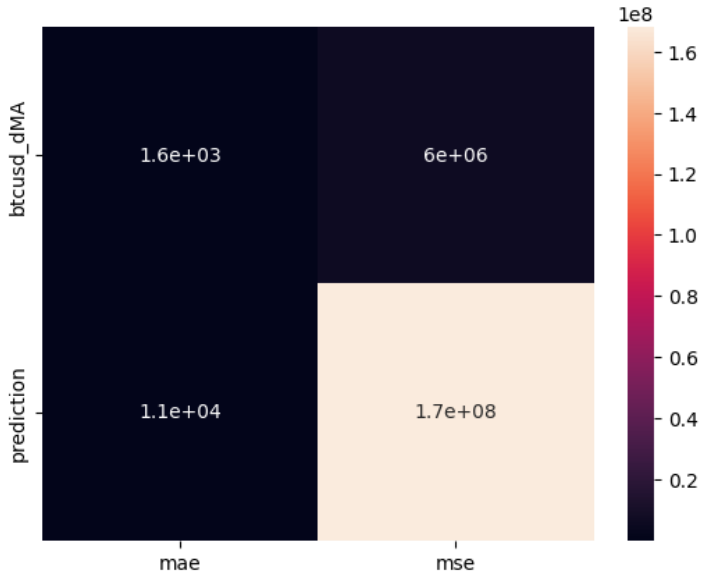
4. Ocena modelu

- Porównano predykcje modelu z rzeczywistymi cenami zamknięcia Bitcoina.
- Dodatkowo, wykreślono średnią kroczącą z 10 okresów dla lepszego zobrazowania trendu (jako punkt odniesienia na wykresie)



Rys. 4 Ewolucja modelu. Wizualizacja rzeczywistych cen bitcoina, średniej kroczącej oraz prognoz cenowych na wykresie.

- Obliczono błąd średni bezwzględny (MAE) oraz błąd średniokwadratowy (MSE) dla oceny dokładności predykcji (**im mniejsze wartości błędów, tym model jest lepszy**). Wartości są dużo większe, ponieważ nie stosowano w tym wypadku normalizowania wartości. Błąd modelu jest znacznie wyższy niż błąd średniej kroczącej w tej próbce danych.



Rys. 5 Wizualizacja przy pomocy mapy cieplnej dla wyników predykcji MAE i MSE. Analiza wartości prognoz (ich dokładność).

5. Predykcja na podstawie kursu Bitcoina

- Przeprowadzono dodatkową analizę, w której do predykcji wykorzystano wyłącznie kurs Bitcoina (bez dodatkowych zmiennych z innych instrumentów).

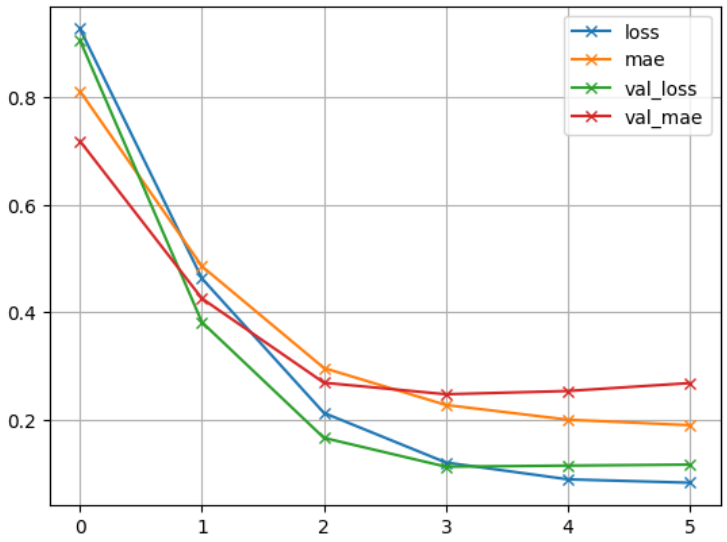
```
Total params: 5,426

Trainable params: 5,426

Non-trainable params: 0
```

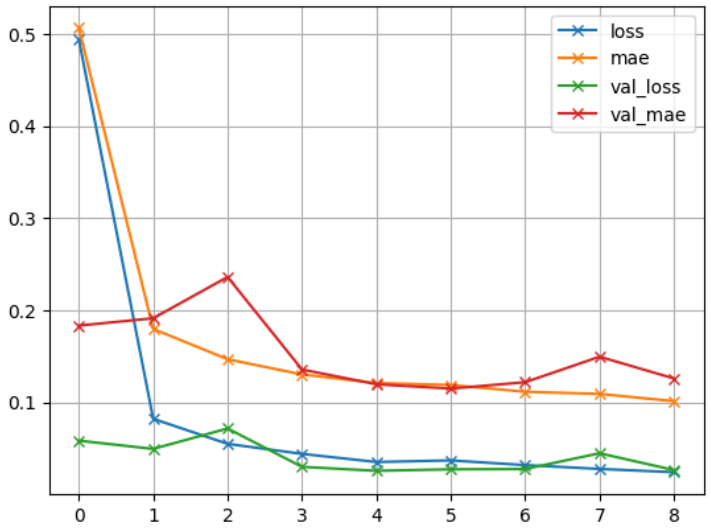
- Ponownie zbudowano i wytrenowano model LSTM, z zastosowaniem early stopping, aby zapobiec przeuczeniu. Parametr cierpliwości ustawiono na 2 dla wyszukiwania hiperparametrów oraz na 4 dla końcowego trenowania. Jeśli przez liczbę epok określoną w parametrze cierpliwości funkcja kosztu nie maleje to trenowanie zostanie zatrzymane.

```
from tensorflow.keras.callbacks import EarlyStopping
```

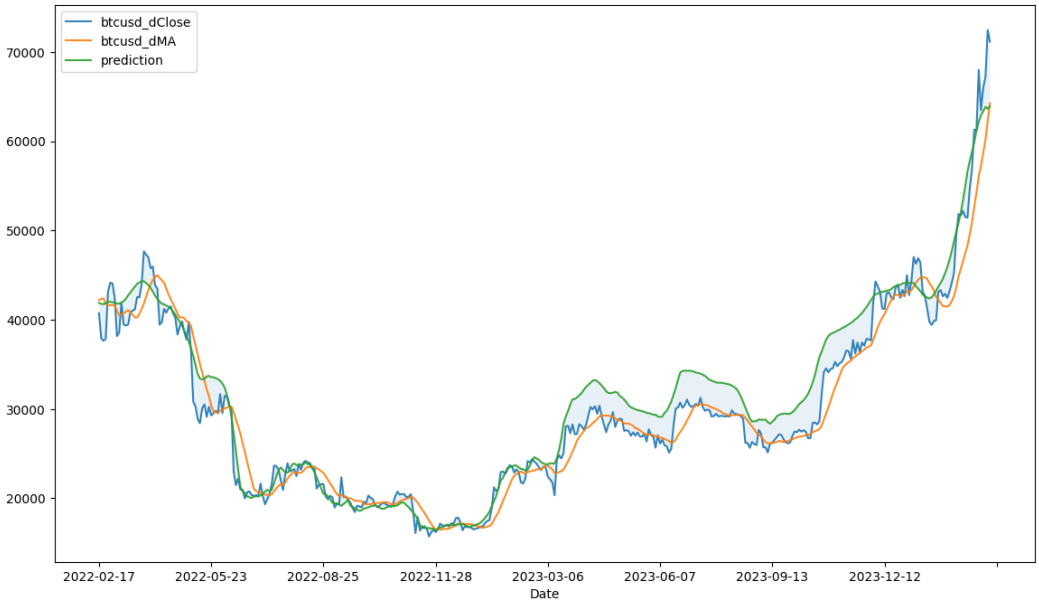


Rys. 6 Wynik trenowania z wykorzystaniem Early Stopping.

- Przeprowadzono optymalizację hiperparametrów modelu (batch_size, num_days, look_ahead) za pomocą metody Grid Search (przeszukanie parametrów i wytrenowanie modelu, dla losowo wybranych 10% z wszystkich możliwych kombinacji parametrów).
- Po optymalizacji i poznaniu najlepszego zestawu parametrów, ponownie wytrenowano model i dokonano predykcji, którą następnie porównano z faktycznymi danymi.
- Otrzymany model zapisano wraz z hiperparametrami do plików h5 i json.



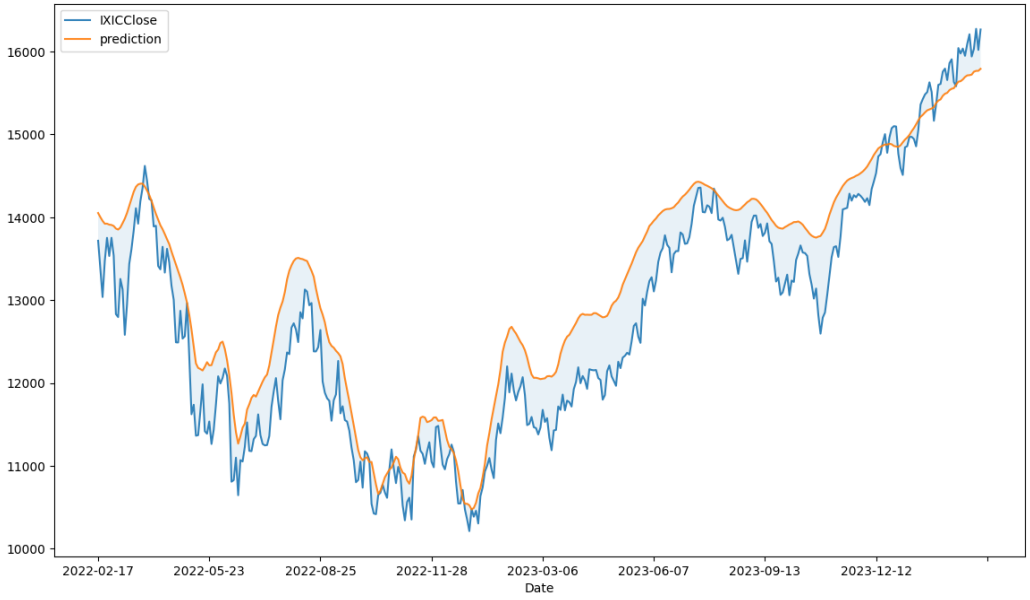
Rys. 7 Wynik trenowania po optymalizacji parametrów: batch_size, num_days, look_ahead.



Rys. 7 Wynik predykcji modelu po czynnościach optymalizacyjnych

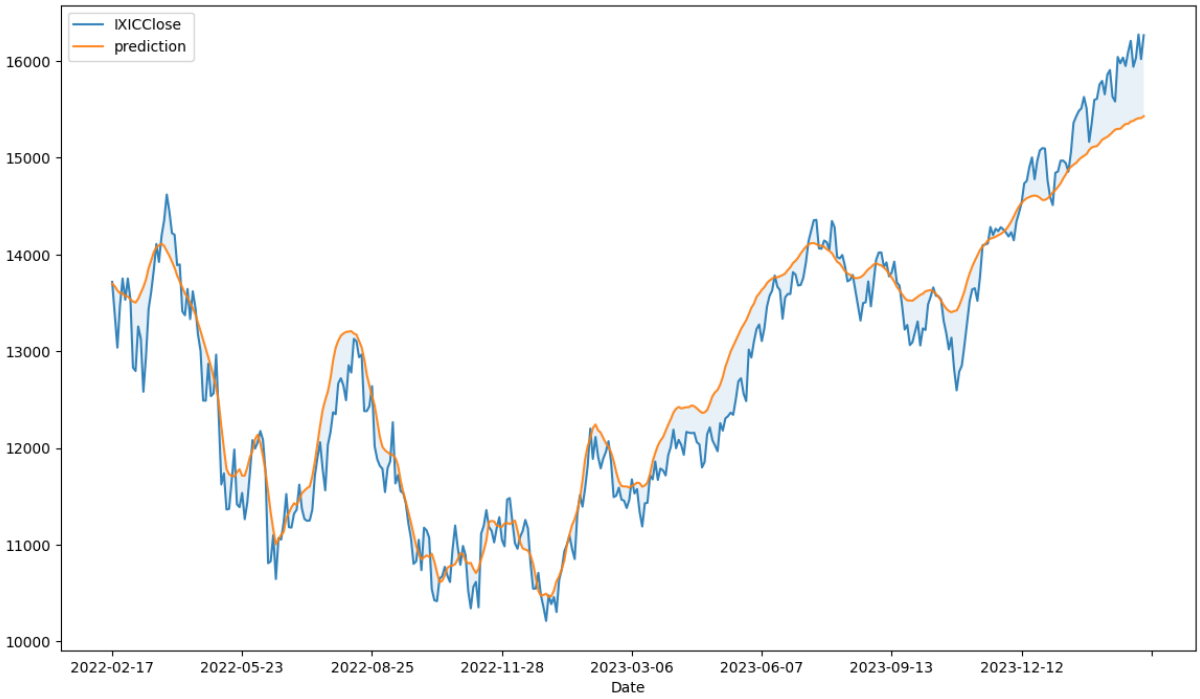
6. Generalizacja - Ewaluacja modelu na danych indeksu NASDAQ.

- Załadowano zapisany model LSTM (z model.json z trzema warstwami LSTM, trzema warstwami dropout i dwoma warstwami gęstymi) oraz jego najlepsze parametry (z pliku params.json – rozmiar batch 16, liczba dni wstecz 16, liczba dni do przodu 4).
- Wykorzystano model do predykcji cen zamknięcia indeksu NASDAQ.



Rys. 8 Wynik predykcji modelu dla indeksu NASDAQ.

- Przeprowadzono dodatkowy trening modelu przez 2 epoki na danych NASDAQ w celu dostosowania go do nowych danych.
- Porównano predykcje z rzeczywistymi wartościami indeksu. Dokładność predykcji widocznie poprawiła się.



Rys. 9 Wynik predykcji dla indeksu NASDAQ po dostosowaniu modelu do danych.

7. Wnioski

- Model LSTM nauczył się rozpoznawać trendy i wzorce w danych dotyczących cen Bitcoina, co pozwoliło mu przewidzieć, jak cena będzie się zmieniać w przyszłości. Optymalizacja hiperparametrów przeprowadzona podczas generalizacji pozwoliła na poprawę jakości predykcji.
- Co ciekawe, model stworzony do przewidywania cen Bitcoina dobrze sprawdził się również w przewidywaniu wartości indeksu NASDAQ, co pokazuje, że może być użyteczny także dla innych instrumentów finansowych (generalizacja).
- Predykcja wykonana modelem uniwersalnym, który wykorzystywał jedynie dane jednego indeksu do jego przewidywania w przyszłości była skuteczniejsza, niż model korzystający z wielu indeksów.
- Niższe wartości batch_size pozwalały uzyskać lepsze wyniki trenowania, niż większe wartości, co oznacza, że mniej uśrednione, bardziej czułe trenowanie jest w tym wypadku skuteczniejsze.