

Programowanie sieciowe

Dane giełdowe - regresja MLP - Lab3

| | | | |
|------------------------------|-----------|-------------|---------------------|
| Data: | 8.05.2022 | Dzień: | Wtorek TN + 1/2 |
| Grupa: | Y02-15b | Godzina: | 17:05 |
| Numer indeksu: | 252889 | Prowadzący: | dr inż. Marek Bazan |
| Nazwisko i imię: Nowek Jakub | | | |

Spis treści

| | | |
|----------|--|-----------|
| 1 | Opis problemu | 2 |
| 2 | Opis użytych algorytmów | 2 |
| 3 | Testy numeryczne | 3 |
| 3.1 | Definicja testów | 3 |
| 4 | Wyniki. | 3 |
| 4.1 | Wyniki funkcji GridSearch oraz wykresy dla różnych wartości K | 3 |
| 4.2 | Błąd predykcji poniżej 5% | 7 |
| 4.3 | Wyniki predykcji dla najlepszych parametrów wybranych przez GridSearch dla poszczególnych solverów . . | 8 |
| 4.3.1 | ADAM | 8 |
| 4.3.2 | SGD | 9 |
| 4.3.3 | LBFGS | 9 |
| 5 | Wnioski. | 10 |

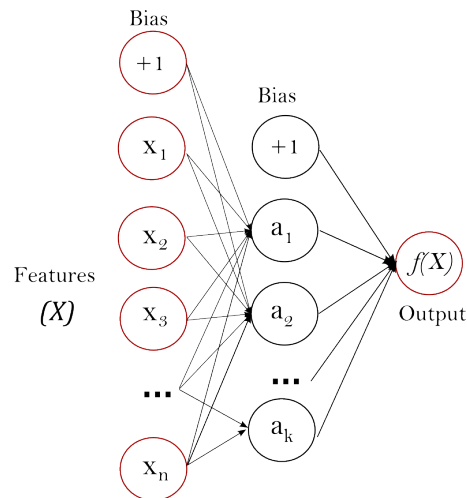
1. Opis problemu

Należało dokonać predykcji wartości jednego ze wskaźników giełdowych jeden dzień w przód na podstawie danych historycznych dla wybranej spółki giełdowej. Dodatkowo należało wyznaczyć ile procent danych wystarczy użyć, by osiągnąć dokładność predykcji większą lub równą 95%. Przy dokonywaniu predykcji należało wykorzystać wielowarstwowy perceptron z jedną warstwą ukrytą. Do zamodelowania wybrano spółkę KIM. Poniższa praca dotyczy przewidywania ceny otwarcia akcji. Poniżej umieszczono link do bazy danych zawierającej dane spółek, która została wykorzystana.

<https://www.kaggle.com/datasets/camnugent/sandp500/metadata>

2. Opis użytych algorytmów

Do stworzenia modelu wykorzystać należało funkcję `MLPRegressor`. Funkcja ta służy do tworzenia modelu, który optymalizuje wartość kwadratu błędu. Trenowanie dokonywane jest przy użyciu wstecznej propagacji bez funkcji aktywacji w warstwie wyjściowej, co można również postrzegać jako wykorzystanie funkcji tożsamościowej jako funkcji aktywacji. Wykorzystuje błąd kwadratowy jako funkcję straty, a wyjściem jest zestaw wartości ciągłych.



Rys. 1: Wielowarstwowy perceptron z jedną warstwą ukrytą i jednym wyjściem.

Do wyszukania najlepszych parametrów spośród podanych, do użycia w funkcji `MLPClassifier`, została użyta funkcja `GridSearchCV`.

Funkcja `GridSearchCV` wyszukuje optymalnych parametrów dla modelu, spośród podanych. Zwraca ona listę modeli w których każdy ma inne parametry. Liczba modeli zwróconych przez funkcję będzie zatem wynikiem iloczynu ilości różnych wartości danego parametru. Trenowanie jest więc wykonywane przy pomocy podanych zbiorów dla każdego wyznaczonego przez funkcję modelu. Po zakończeniu treningu możemy odczytać wyniki testów przeprowadzonych przez funkcję `GridSearch` (metoda `cv_results_`).

Najbardziej interesującym parametrem zwracanym przez `cv_results_` jest **mean_test_score**. Jest to kolumna zawierająca medianę wyznaczonych dokładności dla każdej próbki, dla każdego modelu.

3. Testy numeryczne

3.1. Definicja testów

Dla zestawu danych zawierającego łącznie 1259 różnych próbek, trenowano algorytm za pomocą zbioru treningowego (wartości wejść i wyjść), oraz zbioru testowego, który zawierał dane inne niż w zbiorze treningowym. Dane zbioru treningowego zawierały **(100-K)%** początkowych próbek, natomiast zbiór testowy składał się z **K%** ostatnich próbek.

Dla każdego zestawu danych sprawdzana była dokładność klasyfikacji przy następujących parametrach:

- Liczba neuronów (hidden layer sizes=(N,)) – $N \in \{20, 40, \dots, 100\}$,
- Współczynnik uczenia sieci (learning rate init = η) – $\eta = \{0.1, 0.01, 0.001\}$,
- Algorytm optymalizacji wag (solver = name) – name $\in \{\text{adam, lbfgs, sgd}\}$.
- Typ modyfikacji współczynnika uczenia - 'constant' lub 'adaptive'
- Funkcja aktywacji w warstwie ukrytej - 'tanh' lub 'logistic'

Wynik przedstawiany był w postaci tabeli mean_test_score oraz macierzy pomyłek dla najskuteczniejszego z modeli. Zbiór treningowy zawierał wartości następujących wskaźników:

- open - cena otwarcia akcji w danym dniu;
- high - najwyższa cena osiągnięta w danym dniu;
- low - najniższa cena osiągnięta w danym dniu;
- close - cena zamknięcia akcji w danym dniu;
- volume - ilość transakcji

Dla uproszczenia obliczeń, wszystkie dane zostały znormalizowane do zakresu [0,1].

Wyjściem zbioru treningowego były ceny otwarcia akcji kolejnego dnia - była to zatem kolumna wartości wskaźnika 'open', przesunięta jeden dzień do przodu.

4. Wyniki.

Wyniki testów zaprezentowano na poniższych rysunkach.

4.1. Wyniki funkcji GridSearch oraz wykresy dla różnych wartości K.

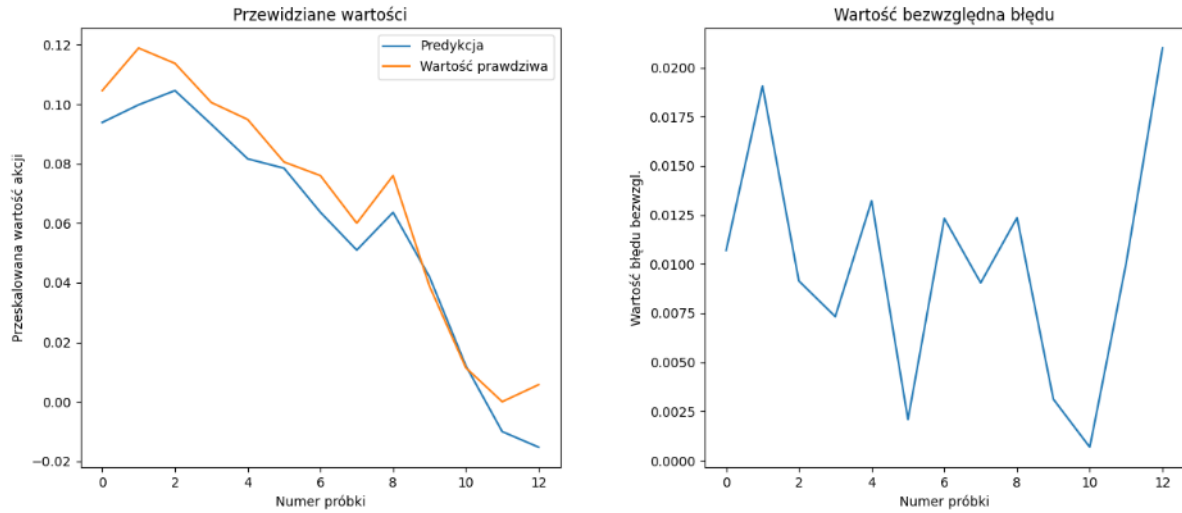
```
confidence: 0.9163005493215914
Grid search results:

      params  mean_test_score  rank_test_score
4  {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -2.002554      120
10 {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -1.336570      119
34 {'activation': 'logistic', 'hidden_layer_sizes': 60, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -0.160320      118
48 {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'adam'} -0.007640      117
54 {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'adam'} -0.006016      116
..      ...      ...      ...
55 {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} 0.983167         5
25 {'activation': 'logistic', 'hidden_layer_sizes': 60, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} 0.983622         4
117 {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} 0.983777         3
97   {'activation': 'tanh', 'hidden_layer_sizes': 80, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} 0.983799         2
111 {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} 0.985137         1

[120 rows x 3 columns]
Przewidziano na podstawie 12 ostatnich dni.
```

Rys. 2: Skuteczność dla K = 1%.

Przewidziano 12 dni (1 dzień do przodu) na podstawie 1246 dni.



Rys. 3: Wykresy błędów dla $K = 1\%$.

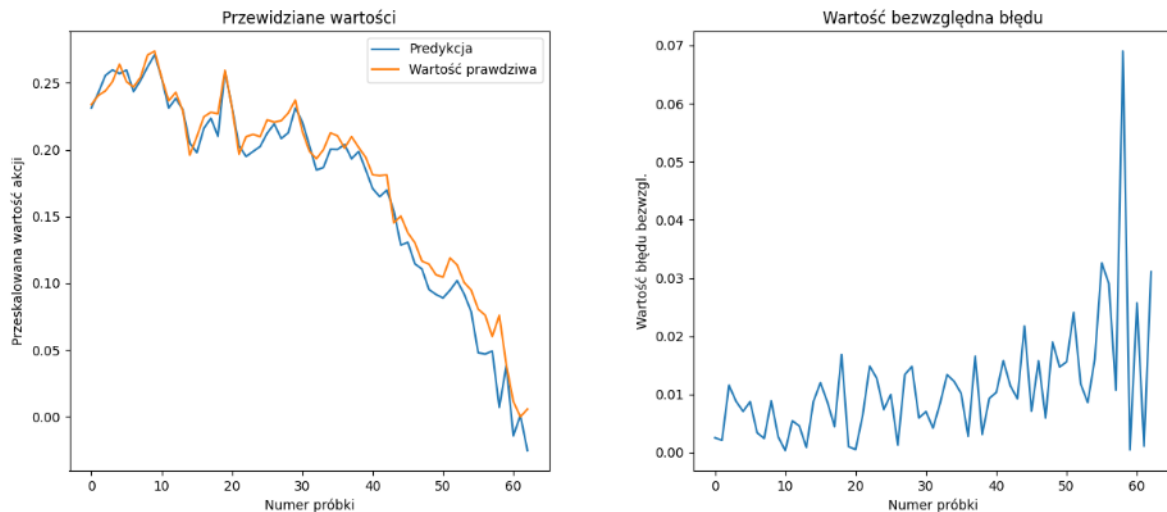
```
confidence: 0.9525574838510038
Grid search results:
```

| | params | mean_test_score | rank_test_score |
|-----|--|-----------------|-----------------|
| 4 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} | -1.643426 | 120 |
| 10 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} | -1.385880 | 119 |
| 34 | {'activation': 'logistic', 'hidden_layer_sizes': 60, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} | -0.531505 | 118 |
| 28 | {'activation': 'logistic', 'hidden_layer_sizes': 60, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} | -0.373618 | 117 |
| 40 | {'activation': 'logistic', 'hidden_layer_sizes': 80, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} | -0.119599 | 116 |
| ... | ... | ... | ... |
| 109 | {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} | 0.983751 | 5 |
| 69 | {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.983933 | 4 |
| 116 | {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'adam'} | 0.984401 | 3 |
| 65 | {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.985686 | 2 |
| 71 | {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.985849 | 1 |

[120 rows x 3 columns]
Przewidziano na podstawie 62 ostatnich dni.

Rys. 4: Skuteczność dla $K = 5\%$.

Przewidziano 62 dni (1 dzień do przodu) na podstawie 1196 dni.



Rys. 5: Wykresy błędów dla $K = 5\%$.

```

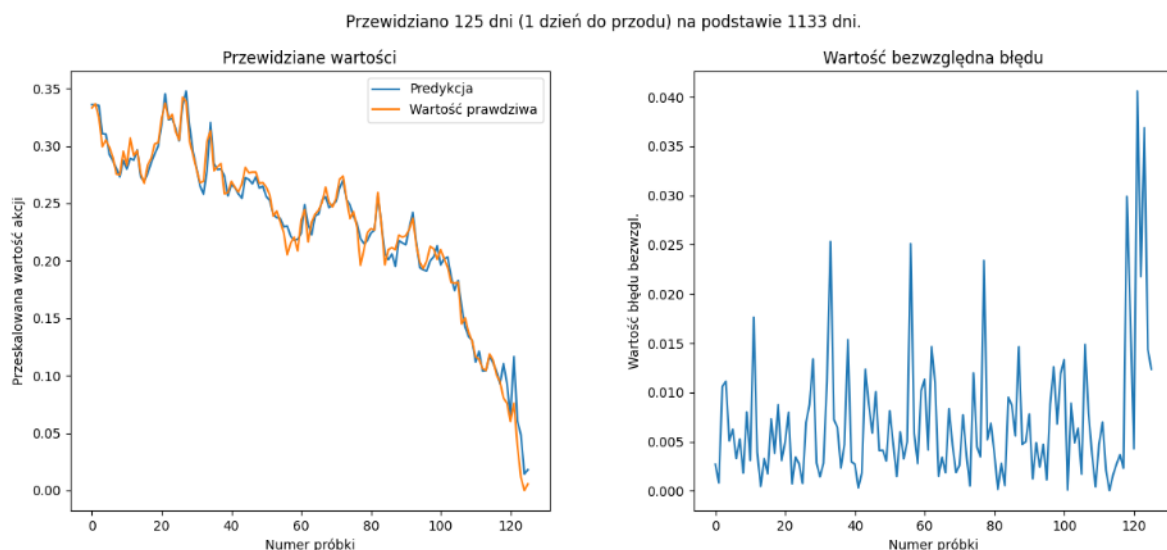
confidence: 0.9825050040330238
Grid search results:

params mean_test_score rank_test_score
4 {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -1.054084 120
10 {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -0.840024 119
22 {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} 0.017680 118
40 {'activation': 'logistic', 'hidden_layer_sizes': 80, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} 0.117359 117
16 {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} 0.191236 116
.. ..
99 {'activation': 'tanh', 'hidden_layer_sizes': 80, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} 0.984815 5
61 {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} 0.985332 4
75 {'activation': 'tanh', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} 0.985723 3
77 {'activation': 'tanh', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} 0.985902 2
65 {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} 0.987297 1

[120 rows x 3 columns]
Przewidziano na podstawie 125 ostatnich dni.

```

Rys. 6: Skuteczność dla $K = 10\%$.



Rys. 7: Wykresy błędów dla $K = 10\%$.

```

confidence: 0.8926397396301431
Grid search results:

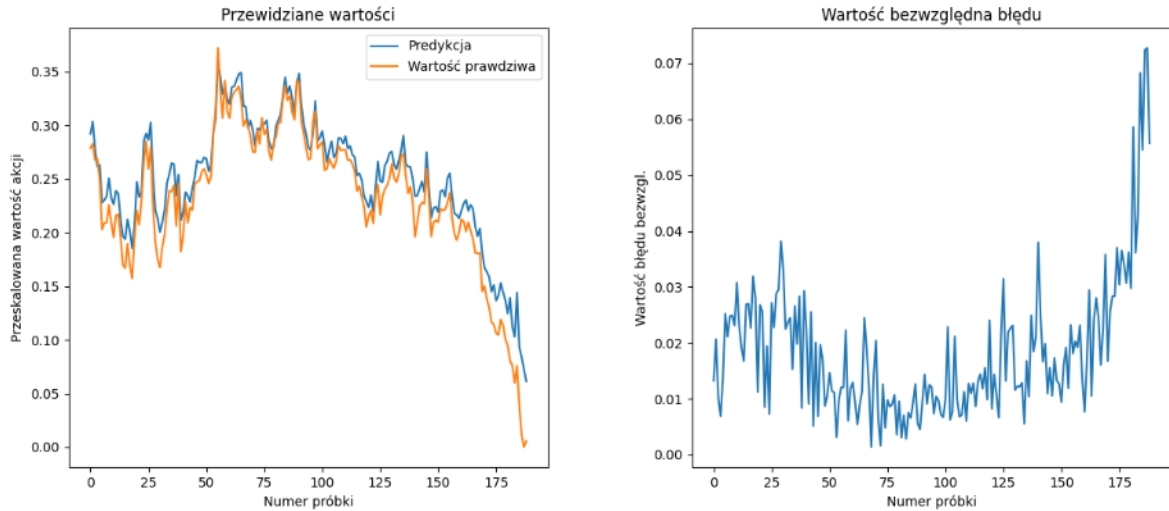
params mean_test_score rank_test_score
10 {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -2.043592 120
16 {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -1.513646 119
4 {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -1.362844 118
22 {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -1.037229 117
58 {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -0.534419 116
.. ..
93 {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} 0.985253 5
119 {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} 0.985522 4
69 {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} 0.986789 3
65 {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} 0.987150 2
67 {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} 0.989326 1

[120 rows x 3 columns]
Przewidziano na podstawie 188 ostatnich dni.

```

Rys. 8: Skuteczność dla $K = 15\%$.

Przewidziano 188 dni (1 dzień do przodu) na podstawie 1070 dni.



Rys. 9: Wykresy błędów dla $K = 15\%$.

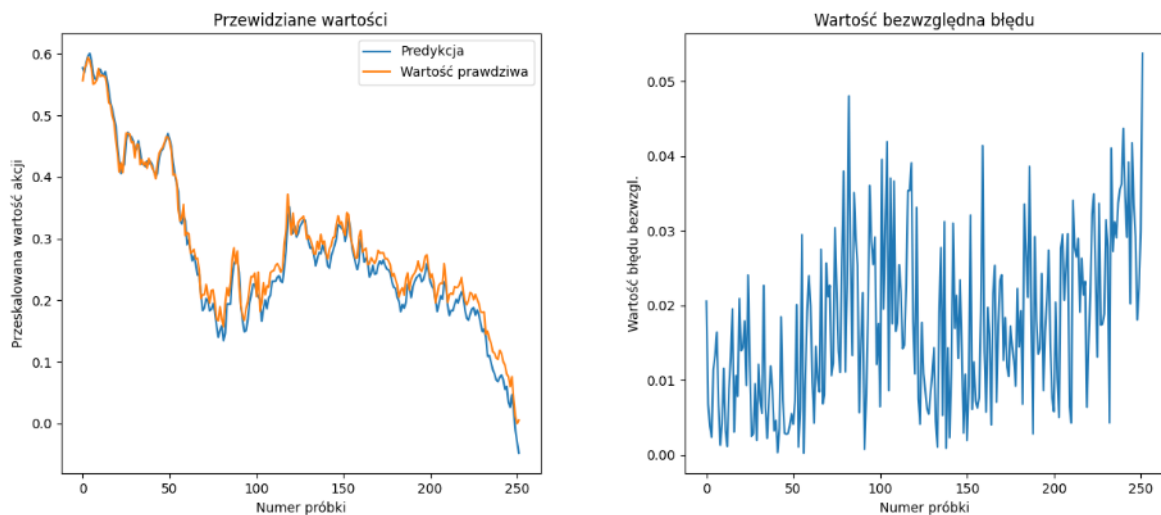
```
confidence: 0.970159839020294
Grid search results:
```

| | params | mean_test_score | rank_test_score |
|-----|---|-----------------|-----------------|
| 56 | {'activation': 'logistic', 'hidden_layer_sizes': 80, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'sgd'} | -5.336416 | 180 |
| 65 | {'activation': 'logistic', 'hidden_layer_sizes': 80, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'sgd'} | -5.181767 | 179 |
| 17 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'sgd'} | -4.430844 | 178 |
| 8 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'sgd'} | -4.424939 | 177 |
| 26 | {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'sgd'} | -4.422465 | 176 |
| .. | ... | ... | ... |
| 64 | {'activation': 'logistic', 'hidden_layer_sizes': 80, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} | 0.983654 | 5 |
| 85 | {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.983941 | 4 |
| 97 | {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.984598 | 3 |
| 133 | {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.984880 | 2 |
| 139 | {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.985102 | 1 |

[180 rows x 3 columns]
Przewidziano na podstawie 80 % dni.

Rys. 10: Skuteczność dla $K = 20\%$.

Przewidziano 251 dni (1 dzień do przodu) na podstawie 1007 dni.



Rys. 11: Wykresy błędów dla $K = 20\%$.

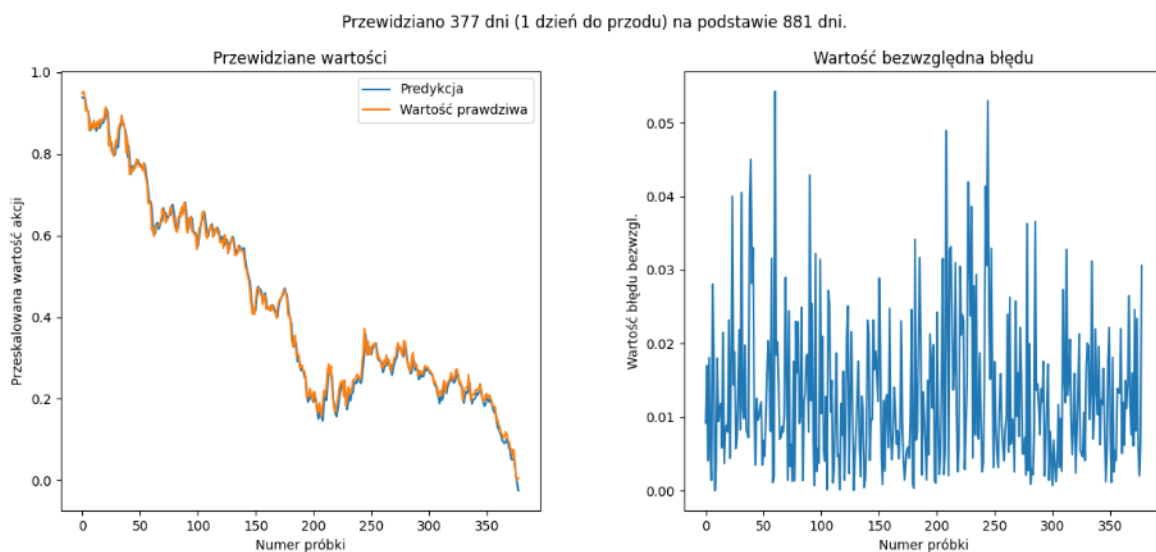
```

confidence: 0.9958952119639211
Grid search results:

```

| | params | mean_test_score | rank_test_score |
|-----|--|-----------------|-----------------|
| 10 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} | -2.622382 | 120 |
| 16 | {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} | -2.430875 | 119 |
| 4 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} | -2.345477 | 118 |
| 22 | {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} | -2.056514 | 117 |
| 52 | {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} | -1.964839 | 116 |
| .. | ... | ... | ... |
| 51 | {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.980905 | 5 |
| 57 | {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.981149 | 4 |
| 107 | {'activation': 'tanh', 'hidden_layer_sizes': 80, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.981708 | 3 |
| 59 | {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.981807 | 2 |
| 113 | {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.983304 | 1 |

Rys. 12: Skuteczność dla $K = 30\%$.



Rys. 13: Wykresy błędów dla $K = 30\%$.

4.2. Błąd predykcji poniżej 5%

Parametry:

- ilość neuronów w warstwie ukrytej: 60
- funkcja aktywacji: tanh
- prędkość uczenia: stała = 0.01
- solver: adam

```

confidence: 0.998441045934779
Grid search results:

```

| | params | mean_test_score | rank_test_score |
|---|---|-----------------|-----------------|
| 0 | {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'adam'} | 0.944506 | 1 |

Przewidziano na podstawie 55 % dni.
ŚREDNIA BŁĘDÓW 0.007322144064127426
MEDIANA 0.004996338681165002

Rys. 14: Skuteczność dla $K = 45\%$, przy użyciu solvera adam.

```

confidence: 0.9953038561410505
Grid search results:

params mean_test_score rank_test_score
77 {'activation': 'tanh', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} 0.972843 1
91 {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} 0.971585 2
73 {'activation': 'tanh', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} 0.970560 3
95 {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} 0.967559 4
97 {'activation': 'tanh', 'hidden_layer_sizes': 80, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} 0.966701 5
.. ..
28 {'activation': 'logistic', 'hidden_layer_sizes': 60, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -3.846273 116
10 {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -3.897183 117
64 {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -3.915425 118
22 {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -3.992983 119
16 {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -4.035174 120

[120 rows x 3 columns]
Przewidziano na podstawie 55 % dni.
ŚREDNIA BŁĘDÓW 0.013541473397461553
MEDIANA 0.01125078097764226

```

Rys. 15: Skuteczność dla $K = 45\%$.

4.3. Wyniki predykcji dla najlepszych parametrów wybranych przez GridSearch dla poszczególnych solverów

4.3.1. ADAM

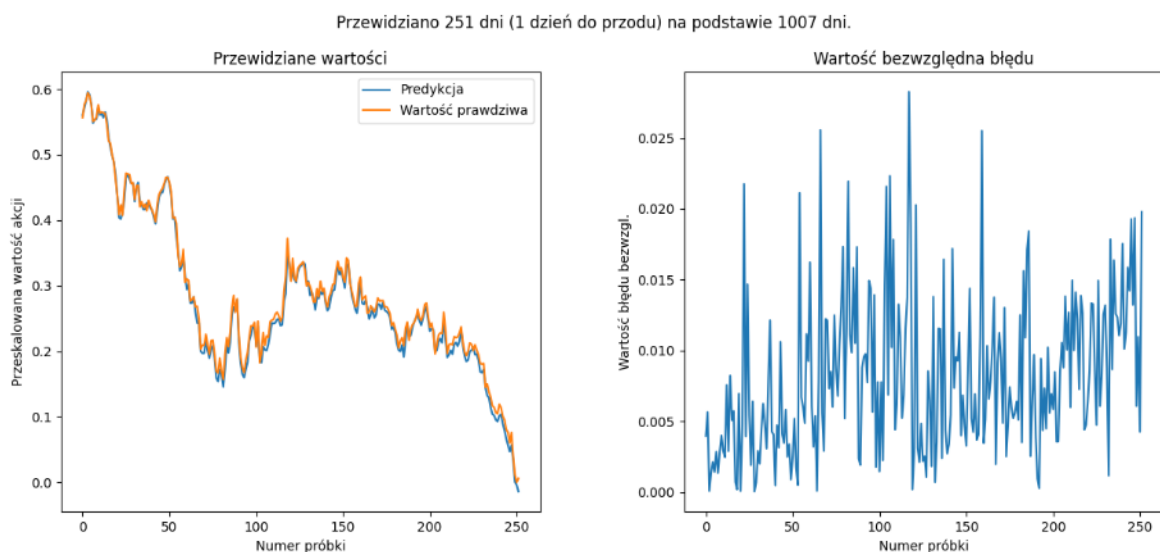
```

Grid search results:

params mean_test_score rank_test_score
5 {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -2.651767 60
14 {'activation': 'logistic', 'hidden_layer_sizes': 60, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -1.156810 59
11 {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'adam'} -0.821196 58
20 {'activation': 'logistic', 'hidden_layer_sizes': 80, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -0.794683 57
8 {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'adam'} -0.605805 56
.. ..
55 {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'adam'} 0.978461 5
43 {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'adam'} 0.979141 4
28 {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'adam'} 0.979552 3
52 {'activation': 'tanh', 'hidden_layer_sizes': 80, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'adam'} 0.980379 2
58 {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'adam'} 0.982831 1

```

Rys. 16: Skuteczność dla $K = 20\%$ przy użyciu solvera adam.



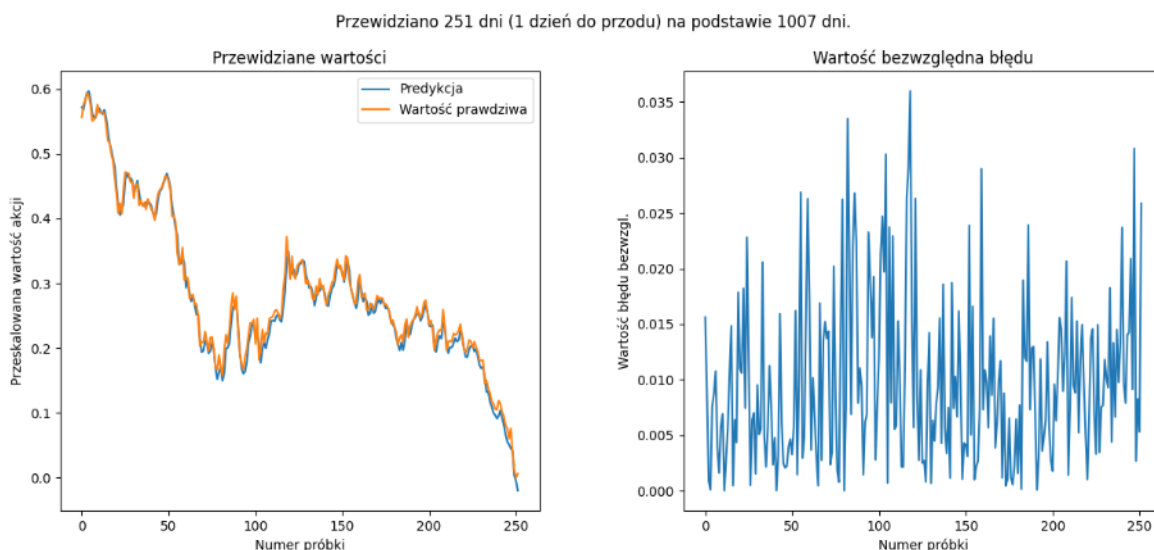
Rys. 17: Wykresy błędów dla $K = 20\%$ przy użyciu solvera adam.

4.3.2. SGD

Grid search results:

| | params | mean_test_score | rank_test_score |
|----|--|-----------------|-----------------|
| 18 | {'activation': 'logistic', 'hidden_layer_sizes': 80, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'sgd'} | -4.941281 | 60 |
| 24 | {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'sgd'} | -4.936532 | 59 |
| 26 | {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'sgd'} | -4.467497 | 58 |
| 11 | {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'sgd'} | -4.404678 | 57 |
| 23 | {'activation': 'logistic', 'hidden_layer_sizes': 80, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'sgd'} | -4.403567 | 56 |
| .. | ... | ... | ... |
| 54 | {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'sgd'} | 0.975460 | 5 |
| 9 | {'activation': 'logistic', 'hidden_layer_sizes': 40, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'sgd'} | 0.976578 | 4 |
| 57 | {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'sgd'} | 0.979300 | 3 |
| 45 | {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'sgd'} | 0.980233 | 2 |
| 51 | {'activation': 'tanh', 'hidden_layer_sizes': 80, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'sgd'} | 0.982053 | 1 |

Rys. 18: Skuteczność dla $K = 20\%$ przy użyciu solvera SGD.



Rys. 19: Wykresy błędów dla $K = 20\%$ przy użyciu solvera SGD.

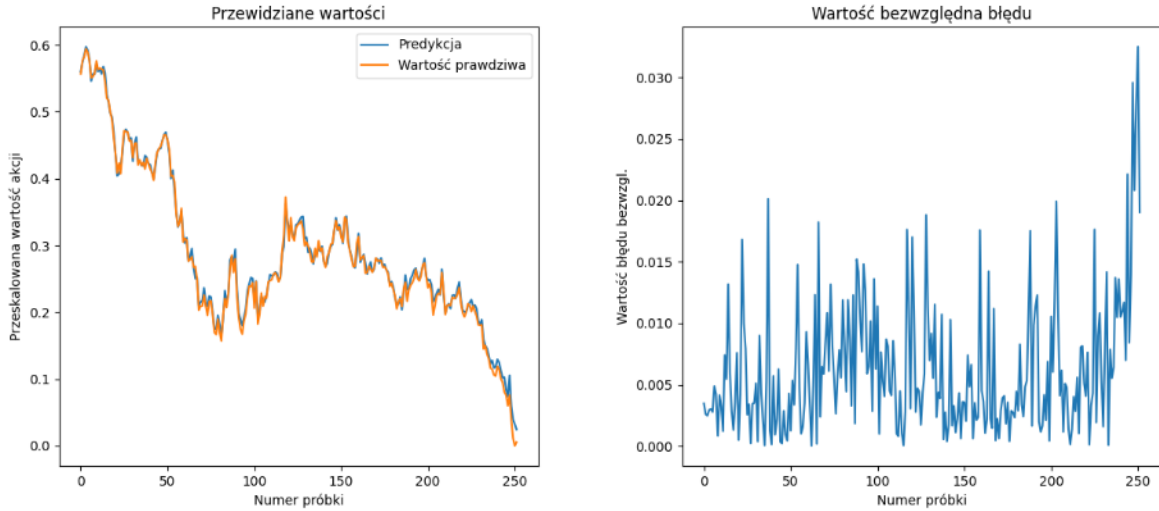
4.3.3. LBFGS

Grid search results:

| | params | mean_test_score | rank_test_score |
|----|---|-----------------|-----------------|
| 34 | {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.971448 | 60 |
| 3 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} | 0.973049 | 59 |
| 1 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.975385 | 58 |
| 0 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} | 0.975973 | 57 |
| 2 | {'activation': 'logistic', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.976259 | 56 |
| .. | ... | ... | ... |
| 59 | {'activation': 'tanh', 'hidden_layer_sizes': 100, 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'solver': 'lbfgs'} | 0.983664 | 5 |
| 25 | {'activation': 'logistic', 'hidden_layer_sizes': 100, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.983779 | 4 |
| 45 | {'activation': 'tanh', 'hidden_layer_sizes': 60, 'learning_rate': 'adaptive', 'learning_rate_init': 0.1, 'solver': 'lbfgs'} | 0.983941 | 3 |
| 52 | {'activation': 'tanh', 'hidden_layer_sizes': 80, 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.984012 | 2 |
| 31 | {'activation': 'tanh', 'hidden_layer_sizes': 20, 'learning_rate': 'constant', 'learning_rate_init': 0.01, 'solver': 'lbfgs'} | 0.986489 | 1 |

Rys. 20: Skuteczność dla $K = 20\%$ przy użyciu solvera LBFGS.

Przewidziano 251 dni (1 dzień do przodu) na podstawie 1007 dni.



Rys. 21: Wykresy błędów dla $K = 20\%$ przy użyciu solvera LBFGS.

5. Wnioski.

- Dla danych testowych, zawierających od 1% do 20% średnia dokładność przewidywania dla najlepszego modelu wybranego przy pomocy funkcji GridSearch wynosiła ponad 98%. Dla $K=30\%$ wartość ta również była podobna. Okazuje się, że przy tak dobranych danych i parametrach, dokładność przewidywania jest bardzo wysoka.
- Dokładność z jaką można przewidzieć przyszłe wartości zależy nie tylko od liczby próbek w zbiorze treningowym, ale też od samych wartości próbek. Funkcje, które były wolnozmiennie znacznie łatwiej jest przewidzieć niż funkcje szybkozmiennie.
- Najlepszy wynik spośród wszystkich solverów osiągnął solver LBFGS, który dla różnych warunków zaprezentowanych powyżej osiągał lepsze wyniki od modeli wykorzystujących metody gradientowe minimalizacji błędu.
- Wartość błędu przewidywania powyżej 5% została osiągnięta przy wykorzystaniu 55% danych historycznych (czyli dla $K=45$), z parametrami:
 - ilość neuronów w warstwie ukrytej: 60
 - funkcja aktywacji: tanh
 - prędkość uczenia: stała = 0.01
 - solver: adam Dla porównania, solver lbfgs w takich warunkach osiągnął dokładność równą 98,37%.
- Najniższy wynik spośród wszystkich solverów osiągnął solver SGD.