

Uniwersytet Łódzki
Wydział Fizyki i Informatyki Stosowanej

RAPORT Z ĆWICZEŃ
Programowanie zaawansowane II

**Ocena jakości prognoz kursu waluty
Porównanie modeli**

Ridge, ElasticNet, Random Forest, Extra Trees
HistGradientBoosting oraz ARIMA/SARIMAX

Autor: Katarzyna Stańczyk
Kierunek studiów: Informatyka
Data wykonania: 15 luty 2026
Rok akademicki: 2025/2026

Łódź, 2026

1 Cel projektu

Celem projektu było zbudowanie i ocenienie predyktora kursu **EUR/PLN** na podstawie danych dziennych NBP (tabela A). Porównano: (i) proste metody bazowe (m.in. *naive*, średnie kroczące), (ii) klasyczne modele uczenia maszynowego (Ridge, ElasticNet, RandomForest, ExtraTrees, HistGradientBoosting), oraz (iii) model szeregow czasowych SARIMAX (traktowany pomocniczo). Ocena została wykonana w czterech horyzontach prognozy: $H \in \{1, 7, 30, 60\}$ dni roboczych.

2 Dane

Źródłem danych jest API Narodowego Banku Polskiego (NBP), kurs średni EUR/PLN [1]. Zakres pobrania: od 2010-01-01 do 2026-02-15, przy czym ostatnia dostępna obserwacja w zbiorze wypada na 2026-02-13 (weekendy i święta nie są publikowane). Zapisano 4067 obserwacji, a brakujące dni (głównie weekendy/święta) nie były imputowane.

3 Przygotowanie danych i cechy

3.1 Definicja problemu

Z szeregu kursu y_t budujemy przykłady uczące w trybie *sliding window*. Dla każdej daty t tworzymy wektor cech x_t na podstawie ostatnich $W = 60$ notowań oraz cech pochodnych, a etykietą jest kurs w przyszłości:

$$\hat{y}_{t+H} \approx f(x_t), \quad H \in \{1, 7, 30, 60\}. \quad (1)$$

3.2 Cechy wejściowe

W projekcie użyto wyłącznie informacji pochodzących z kursu (bez zmiennych zewnętrznych):

- **lagi poziomu** y_{t-1}, \dots, y_{t-W} ,
- **zmiany/zwroty**: $\Delta y_t = y_t - y_{t-1}$, log-zwrot $\log(y_t/y_{t-1})$ i ich lagи,
- **statystyki kroczące** dla okien $K \in \{5, 10, 20, 60\}$: SMA, EMA, odchylenie std, min, max,
- **cechy kalendarzowe**: dzień tygodnia i miesiąc.

3.3 Podział train/val/test

Zastosowano podział czasowy bez mieszania próbek:

- **train**: wszystkie przykłady do 2024-07-24,
- **val**: 2024-07-25 – 2025-01-31 (130 próbek),
- **test**: 2025-02-03 – 2026-02-13 (260 próbek).

Dobór hiperparametrów wykonywano na zbiorze walidacyjnym oraz przy pomocy `TimeSeriesSplit` (5 podziałów) i `RandomizedSearchCV` (40 losowań).

4 Modele

4.1 Metody bazowe

Naive (persistence)

$$\hat{y}_{t+H} = y_t.$$

Założenia: silna inercja / random-walk.

Hiperparametry: brak.

Plusy: bardzo mocny i stabilny benchmark.

Minusy: nie przewiduje zwrotów ani trendów.

Momentum

Ekstrapolacja na podstawie ostatniej zmiany (kontynuacja ruchu).

Hiperparametry: definicja zmiany (np. 1-dniowa vs uśredniona).

Plusy: proste „kierunkowe” zachowanie.

Minusy: bardzo wrażliwy na szum.

SMA

Średnia z ostatnich K obserwacji.

Hiperparametry: K (dobierane na walidacji).

Plusy: stabilne, często dobre dla dużego H .

Minusy: opóźnienie reakcji (lag).

EMA

Wygładzanie wykładnicze (większa waga nowszych obserwacji).

Hiperparametry: K / parametr wygładzania (dobierany na walidacji).

Plusy: bardziej adaptacyjne niż SMA.

Minusy: nadal słabe w „przewidywaniu” zwrotów.

4.2 Modele liniowe (scikit-learn)

Ridge (regresja liniowa + $L2$)

Co robi: minimalizuje błąd + kara $L2$ stabilizująca wagi przy wielu skorelowanych cechach.

Założenia: relacja w przybliżeniu liniowa; względnie stały reżim.

Hiperparametry: alpha.

Plusy: szybki, odporny na współliniowość, dobry baseline ML.

Minusy: nie łapie nieliniowości.

ElasticNet ($L1+L2$)

Co robi: łączy Ridge i Lasso (stabilizacja + częściowa selekcja cech).

Założenia: tylko część cech niesie sygnał, reszta jest redundantna.

Hiperparametry: alpha, l1_ratio.

Plusy: ogranicza przeuczenie, może „wyzerować” część cech.

Minusy: nadal liniowy, wrażliwy na dobór regularyzacji.

4.3 Zespoły drzew (scikit-learn)

RandomForest

Co robi: uśrednia wiele drzew trenowanych na próbkach bootstrap (model nieliniowy).

Kiedy działa dobrze: krótkie horyzonty, nieliniowe zależności w cechach.

Hiperparametry: n_estimators, max_depth, min_samples_leaf, max_features.

Plusy: łapie nieliniowości, nie wymaga skalowania.

Minusy: kosztowny; w szeregach czasowych może się przeuczać; słaba ekstrapolacja.

ExtraTrees

Co robi: jak RF, ale z bardziej losowymi podziałami (często mniejsza wariancja).

Hiperparametry: jak RF.

Plusy: bywa bardziej „odporny” niż RF.

Minusy: większy bias przy zbyt dużej losowości; podobne ograniczenia jak RF.

4.4 Boosting drzew (scikit-learn)

HistGradientBoostingRegressor

Co robi: sekwencyjnie dodaje drzewa korygujące błędy poprzednich (silny model tablicowy).

Założenia: złożony sygnał i nieliniowości; konieczne strojenie i kontrola przeuczenia.

Hiperparametry: learning_rate, max_iter, max_depth/max_leaf_nodes, min_samples_leaf, (opc.) early_stopping.

Plusy: często top jakość przy dobrym tuningu.

Minusy: czuły na hiperparametry i drift; dla dużego H może „wygładzać” i tworzyć bias.

4.5 Modele szeregów czasowych (statsmodels)

SARIMAX

Co robi: ARIMA z komponentem sezonowym; model liniowy zależności czasowych (AR/MA) + różnicowanie.

Założenia: (po różnicowaniu) stacjonarność; poprawne odwrócenie transformacji do poziomu.

Hiperparametry: order=(p, d, q), seasonal_order=(P, D, Q, s).

Plusy: klasyczny benchmark, diagnostyka reszt.

Minusy: bardzo wrażliwy na konfigurację; przy złych ustawieniach może dawać skrajne błędy.

Modele liniowe trenowane na cechach standaryzowanych, natomiast modele drzewiaste nie wymagają skalowania.

5 Metryki oceny

Błąd pojedynczej prognozy definiujemy jako:

$$e_i = \hat{y}_i - y_i. \quad (2)$$

W raporcie wykorzystano następujące metryki (na zbiorach val i test):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |e_i|. \quad (3)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}. \quad (4)$$

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{e_i}{y_i} \right|. \quad (5)$$

$$\text{sMAPE} = \frac{100\%}{n} \sum_{i=1}^n \frac{2|e_i|}{|\hat{y}_i| + |y_i|}. \quad (6)$$

Dla $H = 1$ dodatkowo raportowano *directional accuracy* (czy model poprawnie przewidziały znak zmiany względem y_t).

6 Wyniki

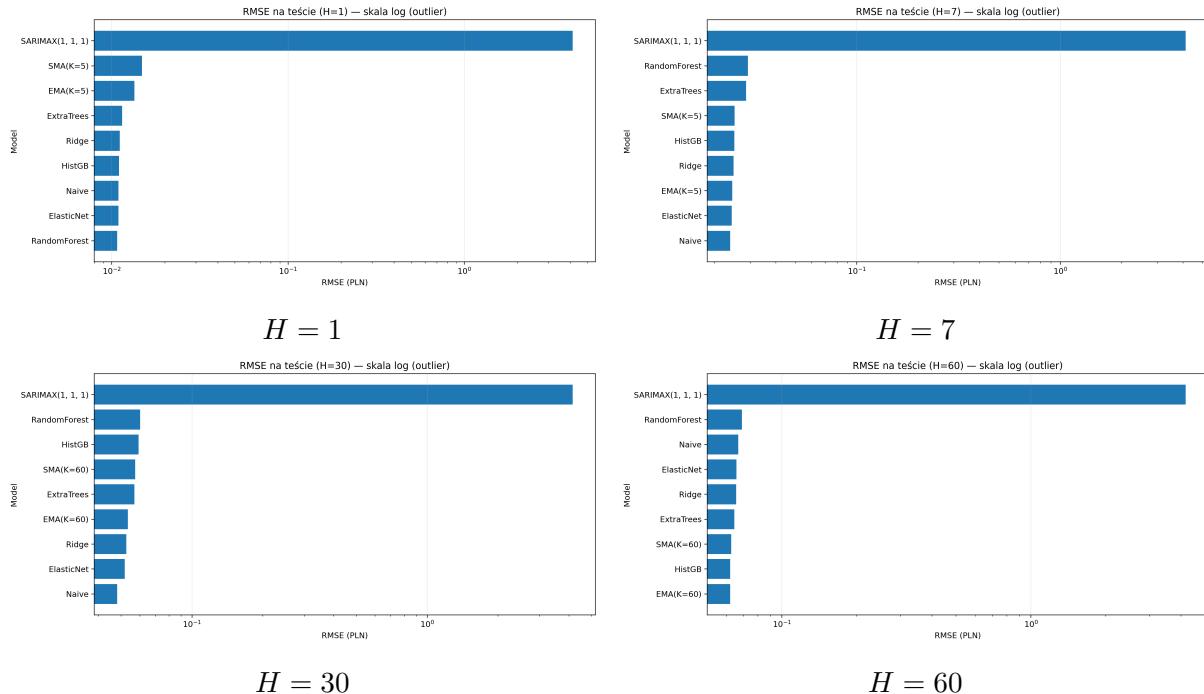
Poniższa tabela podsumowuje najlepszy model na teście dla każdego horyzontu. Wartości są w PLN (MAPE/sMAPE w %).

Tabela 1: Najlepszy model na zbiorze testowym dla każdego horyzontu (błąd $e = \hat{y} - y$).

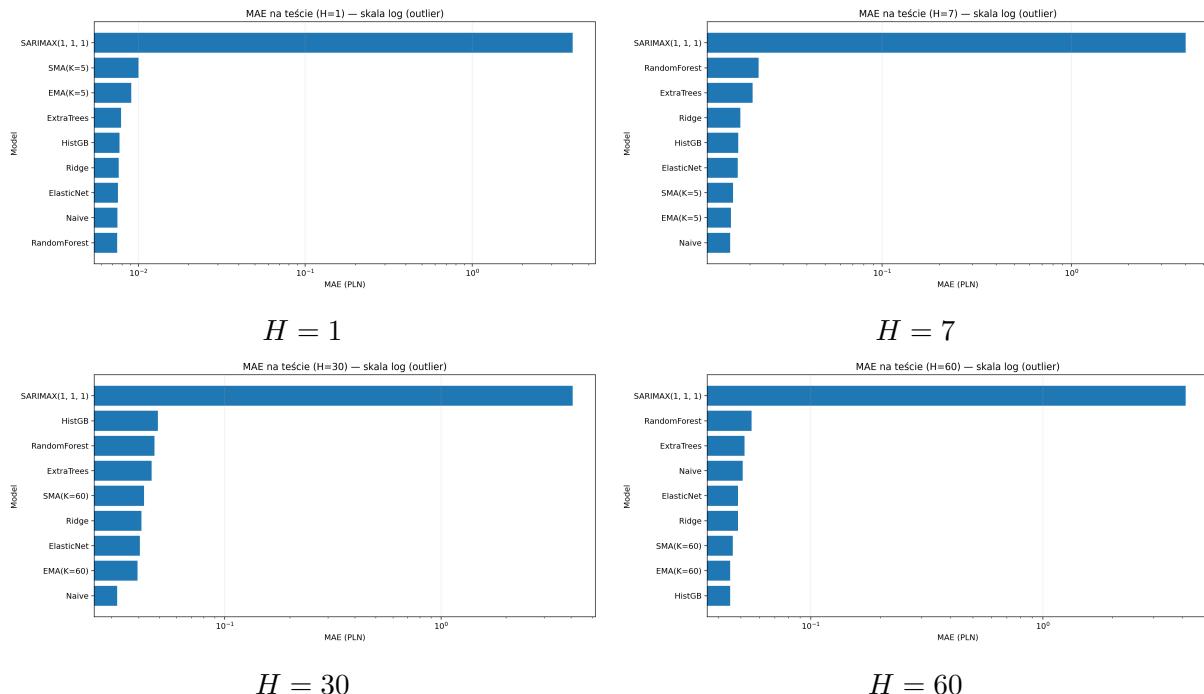
H	Model	RMSE	MAE	MAPE	sMAPE	DirAcc	MeanErr	P95 e	Max e
1	RandomForest	0.010734	0.008113	0.191052	0.191059	0.562	-0.001080	0.020090	0.061615
7	Naive	0.035038	0.026292	0.620318	0.620241		-0.004577	0.063073	0.118177
30	Naive	0.048398	0.038670	0.910825	0.909678		0.007420	0.112293	0.160986
60	EMA(K=60)	0.051438	0.042774	1.008289	1.006102		0.032204	0.154482	0.181174

6.1 Porównanie RMSE i MAE (test)

Na rys. 1 oraz 2 pokazano RMSE/MAE dla wszystkich modeli (skala logarytmiczna z uwagi na odstający SARIMAX).



Rysunek 1: RMSE na zbiorze testowym dla różnych horyzontów.



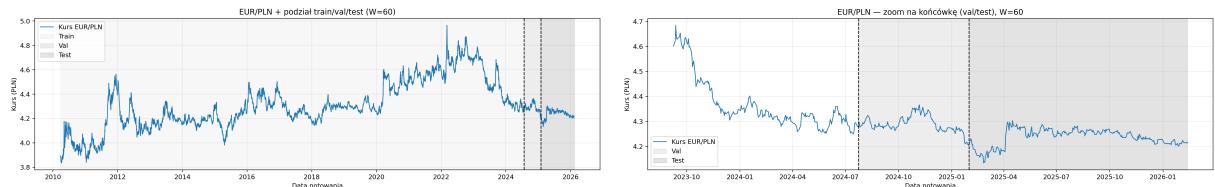
Rysunek 2: MAE na zbiorze testowym dla różnych horyzontów.

6.2 Prognoza vs rzeczywistość (test)

Dla krótkiego horyzontu ($H = 1$) wszystkie topowe modele są bardzo blisko wartości rzeczywistych, a różnice między nimi są niewielkie. Dla dłuższych horyzontów (np. $H = 30$) modele ML mają tendencję do nadmiernego wygładzania i/lub biasu, przez co baseline *naive* bywa trudny do pobicia.

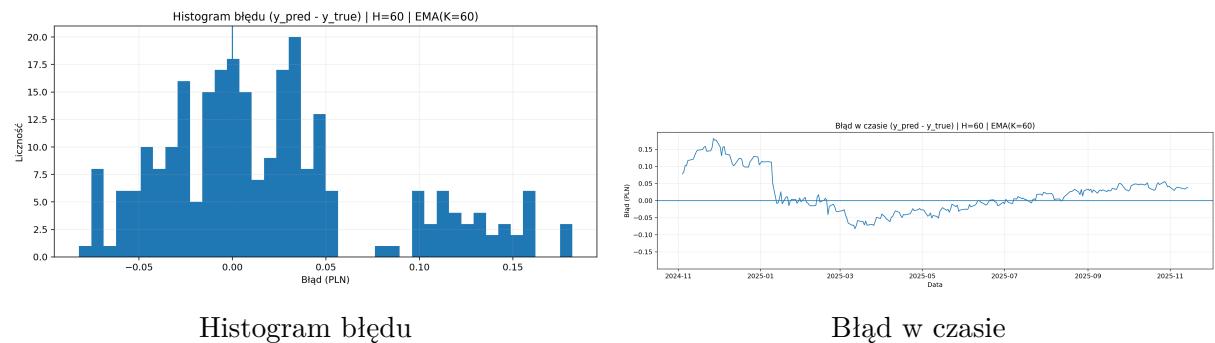


Rysunek 3: Porównanie predykcji i wartości rzeczywistych na teście (top 3 wg RMSE).



6.3 Analiza błędów

Na rys. 5 pokazano rozkład błędu i jego przebieg w czasie dla najlepszego modelu przy $H = 60$ (EMA). Widoczna jest zmiana biasu w czasie, co sugeruje *dryf* (zmianę reżimu kursu), którego proste baseline nie „nadążają” idealnie.



Rysunek 5: Analiza błędu dla $H = 60$ (najlepszy model na teście).

7 Omówienie wyników

- $H = 1$: RandomForest osiąga najlepszy RMSE, ale poprawa względem *naive* jest niewielka. W praktyce oznacza to, że dla kursu walutowego w horyzoncie 1 dnia bardzo trudno jest uzyskać stabilną przewagę nad modelem „jutro będzie jak dziś”.
- $H = 7$ i $H = 30$: najlepszy jest *naive*. Modele ML (zwłaszcza zespoły drzew) poprawiają wyniki na walidacji, ale pogarszają się na teście (ryzyko przeuczenia oraz zmiana reżimu w okresie testowym).
- $H = 60$: najlepszy jest baseline EMA($K=60$), a HistGradientBoosting jest bardzo blisko. Dla długich horyzontów modele tendencjonalnie wygładzają serię i trudno im uchwycić przyszłe zmiany bez dodatkowych danych zewnętrznych.
- **SARIMAX:** w tej konfiguracji daje błąd rzędu kilku PLN, co jest niefizyczne w porównaniu do pozostałych wyników. Najbardziej prawdopodobna przyczyna to niepoprawna konfiguracja (np. zła skala danych, problem z różnicowaniem/odwróceniem transformacji lub nieudane dopasowanie). W dalszej części projektu SARIMAX powinien być uruchamiany w wersji zweryfikowanej lub wyłączony z porównań.

8 Wnioski

Najważniejszy wniosek jest praktyczny: dla kursu EUR/PLN prognozowanie poziomu kursu na podstawie samych historycznych notowań w krótkim horyzoncie ($H = 1$) daje co najwyżej niewielką poprawę względem *naive*, a dla dłuższych horyzontów baseline (*naive/EMA*) bywa trudny do pobicia. Dalsze ulepszenia powinny iść w kierunku: (i) modelowania *zmiany* (delta/zwrotu) zamiast poziomu, oraz (ii) wykorzystania dodatkowych zmiennych zewnętrznych (jeśli regulamin na to pozwala).

Literatura

- [1] Api webservice narodowego banku polskiego (nbp). <https://api.nbp.pl/>, 2026. dostęp: 2026-02-16.