

LAB7 – Prognozowanie BTC (1D) – raport

Data: 2026-01-11

Celem ćwiczenia było zbudowanie modeli uczących się przewidywania dziennych log-zwrotów BTC na podstawie okna historycznego (window=90) oraz cech kalendarzowych (harmonicznych) i statystyk kroczyących. Porównano dwa podejścia: model gęsty (Dense/MLP) oraz sekwencyjny model rekurencyjny (RNN: GRU/LSTM), z dodatkowym strojeniem hiperparametrów (KerasTuner).

1. Dane i przygotowanie

Wejściem jest plik CSV z dziennymi danymi BTC (kolumny: „Open time”, „Close”). Z ceny tworzone są log-ceny i log-zwroty:

$$\log_price[t] = \ln(price[t] + \varepsilon), \varepsilon = 1e-8$$

$$r[t] = \log_price[t] - \log_price[t-1]$$

Uczenie odbywa się na $r[t]$ (to jest zmienna docelowa).

2. Inżynieria cech

Dla każdej obserwacji $r[t]$ budowane są cechy: $r[t]$ (opóŹniony log-zwrot jako cecha wejściowa) **harmoniczne roczne** z fazy w roku (sin/cos dla $k=1..harm_k$) **harmoniczne miesieczne** z fazy w miesiącu (sin/cos dla $k=1..harm_k$) opcjonalnie: **rolling mean** i **rolling std** z ostatnich rolling_window zwrotów Wszystkie cechy są następnie skalowane (scaler.pkl) i muszą być 1:1 identyczne w treningu i inferencji.

3. Modele

Porównano: **Dense (MLP)** – model działający punktowo na wektorze cech (bez jawnej pamięci sekwencji). **RNN (GRU/LSTM)** – model przyjmujący sekwencje długości window=90: (1, window, d). RNN był strojony KerasTunerem (kt_max_trials=10, kt_epochs=6), a potem trenowany pełniej (epochs=25).

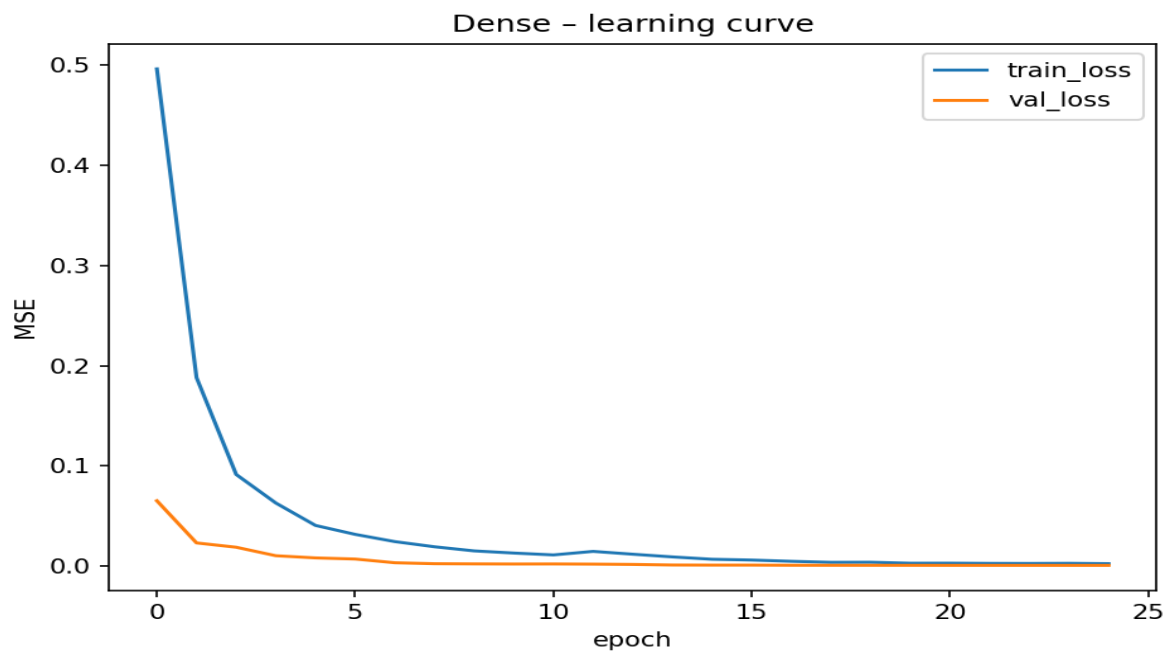
4. Wyniki ilościowe (test)

Model	Test MSE	Test MAE	Corr(y,ŷ)	Directional acc. (sign)
Dense	0.000602	0.018088	-0.035	0.489
RNN (LSTM tuned)	0.000505	0.015992	0.058	0.501

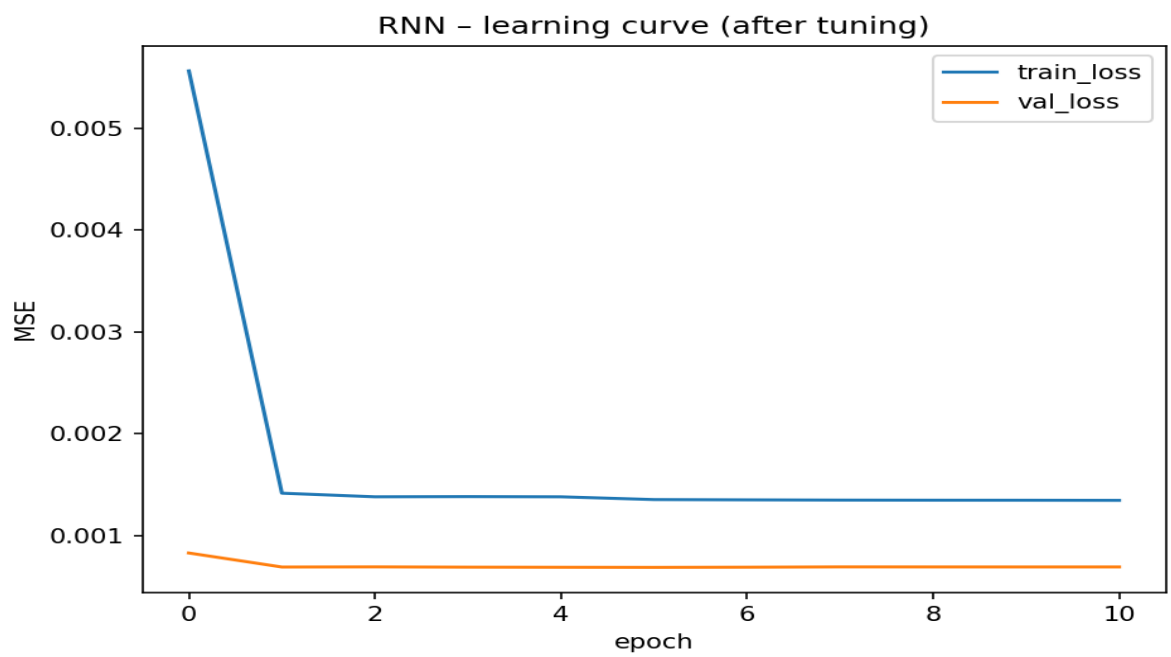
Dodatkowa obserwacja: predykcje RNN mają bardzo małą wariancję (odchylenie standardowe $\approx 1.08e-05$), czyli model w praktyce zwraca prawie stałą wartość. Dla Dense odchylenie standardowe ≈ 0.0091 . To wyjaśnia, dlaczego RNN może mieć niski MSE, ale na wykresie wygląda jak „linia prosta” – przewidywanie blisko średniej minimalizuje MSE, lecz nie śapie skoków zwrotów.

5. Wykresy

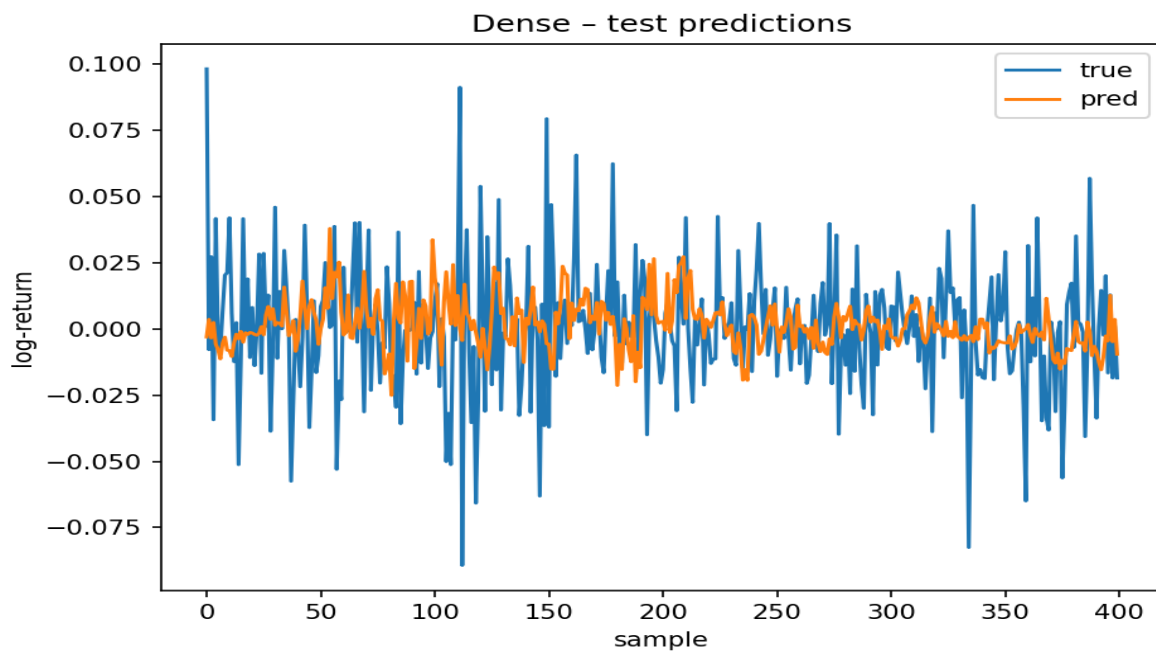
Krzywa uczenia: Dense



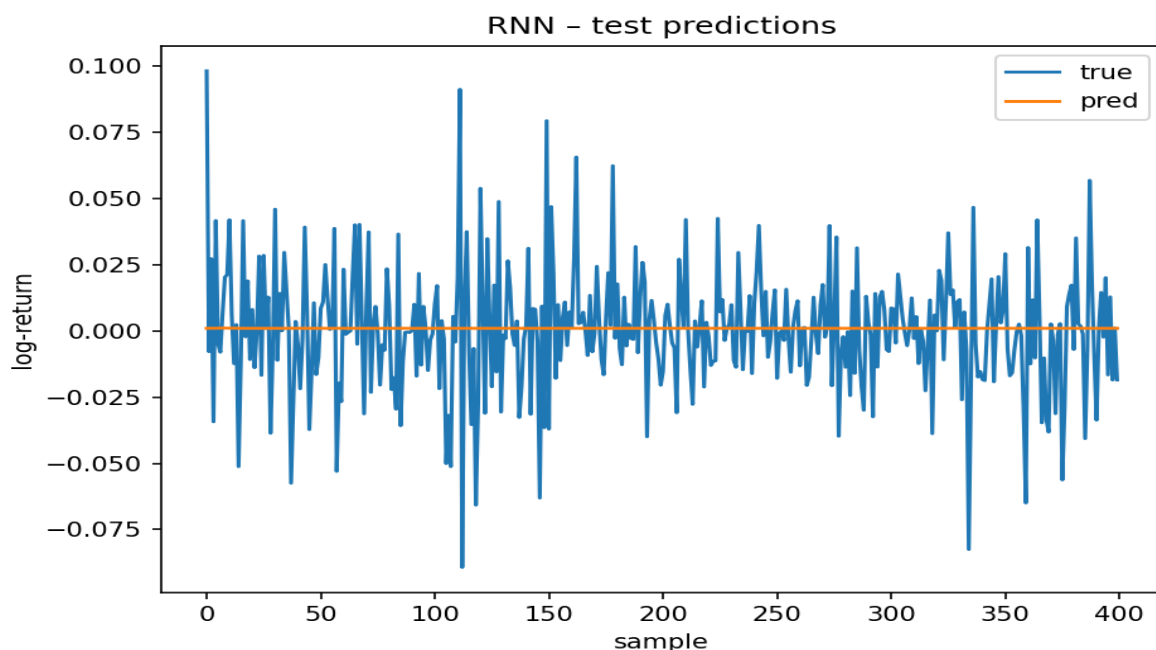
Krzywa uczenia: RNN (po tuningu)



Predykcje na teście: Dense (true vs pred)



Przykład na testach: RNN (true vs pred)



6. Interpretacja i wnioski

Dlaczego Dense i RNN dają inne prognozy? To inne klasy modeli (inna funkcja, inna inductive bias). RNN może „zadowolili się” przewidywaniem stałej wartości blisko średniej, co obniża MSE, ale daje mało użyteczne predykcje. **Dlaczego wyniki uruchomień różnią się między sobą?** Uczenie sieci jest stochastyczne (losowa inicjalizacja wag, losowe batchowanie), a na CPU dodatkowo czynniki operacji mogą być niedeterministyczne (oneDNN / różna kolejność sumowania FP). To normalne. **Czy dodanie harmonicznych miesiecznych pomaga?** Nie da się uczciwie stwierdzić bez ablacji i (A/B): trening „z” i „bez” miesiecznych na tym samym podziale danych, tych samych seedach i kilku powtórzeniach. W samych logach widać, że metryki testowe potrafią się wahać między uruchomieniami – to może przykrywać mały efekt cech. **Co jest realnie najlepszym kryterium?** Dla finansów sama MSE na log-zwrocie bywa myląca. Warto raportować też korelację, trafność kierunku oraz test strategii (np. prosta strategia long/short z kosztami) – dopiero wtedy widać, czy model ma „signal”.

7. Reprodukowalność (opcjonalnie)

Aby zmniejszyć rozrzuty między uruchomieniami: ustaw seedy: `PYTHONHASHSEED`, `np.random.seed`, `tf.random.set_seed` w TensorFlow: `tf.config.experimental.enable_op_determinism()` (jeśli dostępne)
dla CPU: można wyłączyć oneDNN: `TF_ENABLE_ONEDNN_OPTS=0` (kosztuje wydajności)