Financial time series forecasting with Transformers: Impact of additional features and cross-validation on model performance

Anton Zaitsev, University of Luxembourg, MADS

Motivation & Hypotheses

- Which data should be used to help the deep learning model grasp complex relationships within financial time series data so that it can yield more accurate results?
- How can we design a crossvalidation technique that better captures the temporal nature of nonstationary time series?

Cross-validation

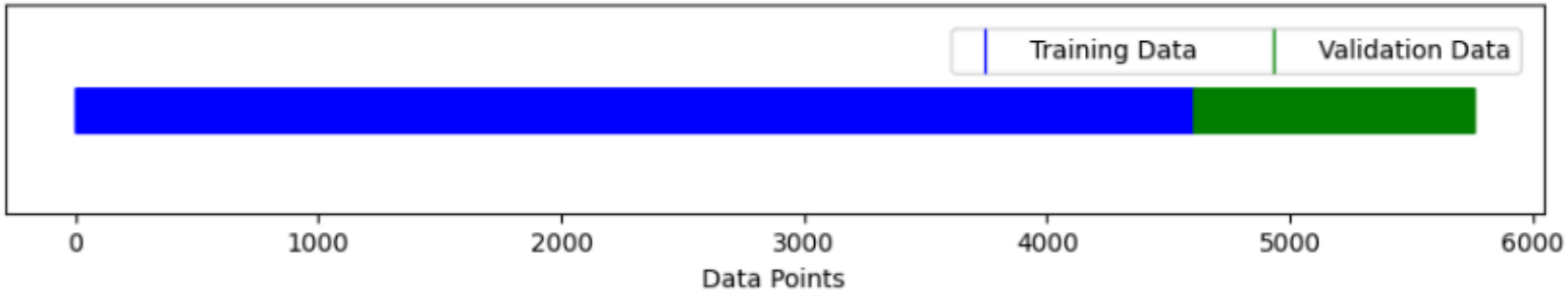

Figure 3: Default train / validation data splitting approach.
Ratio of the validation data is 20%.

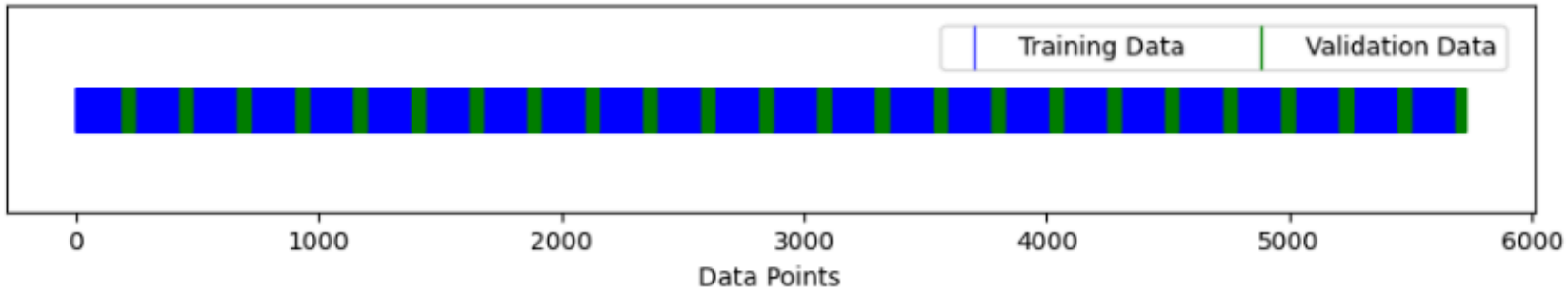

Figure 4: Custom train / validation data splitting approach.
Calculated training batch size is 190 and validation batch size is 46; ratio of the validation data is $\approx 19.7\%$.

Data

- The study is based on the historical stock market data (all data are available online via https://www.alphavantage.co/ API).
- The dataset comprises historical daily closing prices and volumes for three closely correlated US technology stocks: *ORCL*, *CSCO*, and *QCOM*.
- Additional data includes daily technical indicators (*MACDEXT*, *RSI*, *BBANDS*) and quarterly balance sheet statements (cash flow, earnings, income).

Model

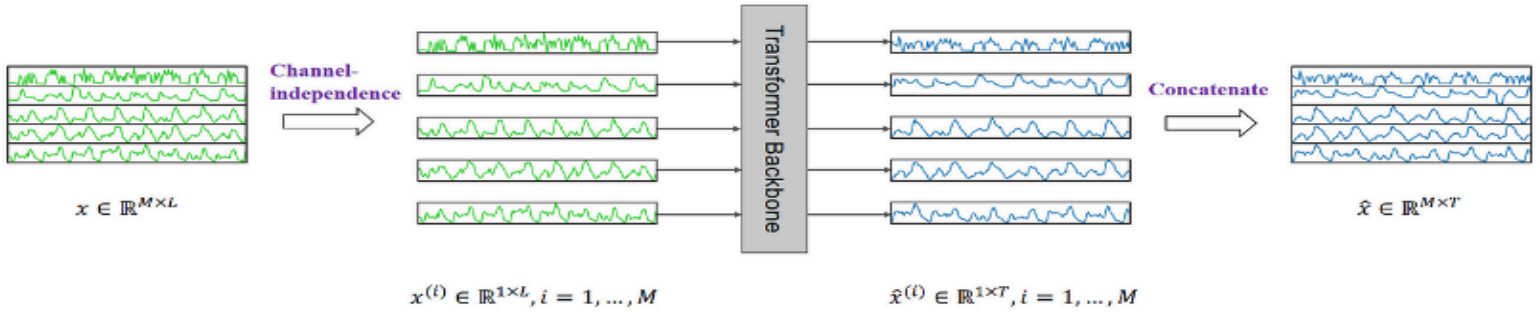

Figure 5: Original PatchTST model overview.
Here, M should be consistent before and after passing the data through the Transformer Backbone.

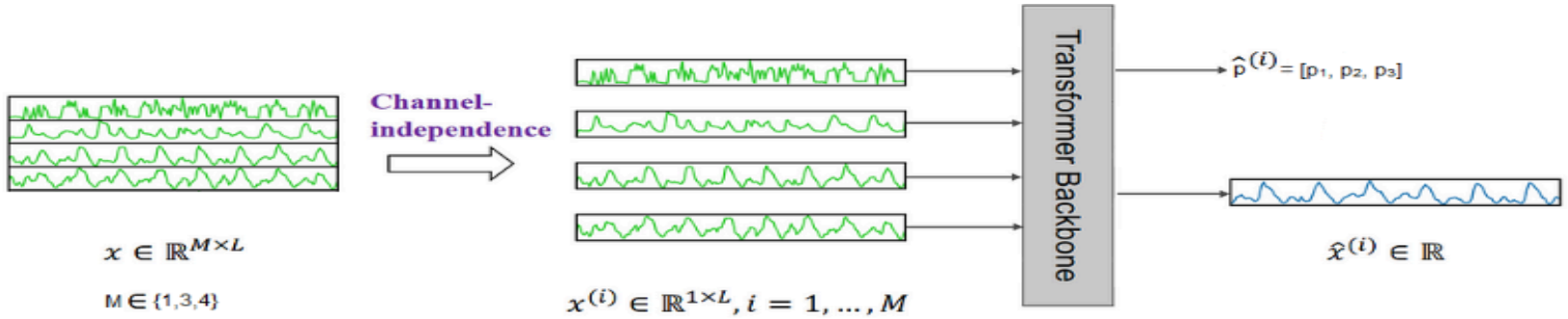

Figure 6: Model used: PatchTST backbone and custom head.

Labels

$$l_t = \begin{cases} 0, \frac{c_t}{c_{t-1}} - 1 < -0.05 \\ 1, -0.05 \leq \frac{c_t}{c_{t-1}} - 1 \leq 0.05 \\ 2, \frac{c_t}{c_{t-1}} - 1 > 0.05 \end{cases}$$

Results

Task / Model	Split	Price MAE	Price MAPE	Label CE
baseline	-	**0.4103**	**0.8052**	-
a	default	0.4526	0.8898	1.1026
a	custom	0.4820	0.9454	1.1017
b	default	0.5753	1.1200	1.1302
b	custom	0.5613	1.0995	1.1151
c	default	0.5418	1.0694	1.0856
c	custom	0.6102	1.2044	1.0640
d_baseline	-	**1.2497**	**1.1808**	-
d	default	1.3514	1.2776	-
d	custom	1.3765	1.3075	-

Table 2: Performance metrics for different model configurations.

- task a (univariate): price $\rightarrow$ price (single stock)
- task b (univariate): price $+$ additional features $\rightarrow$ price (single stock)
- task c (univariate): price $+$ correlated stocks $\rightarrow$ price (single stock)
- task d (multivariate): prices of three stocks $\rightarrow$ prices of three stocks

Conclusion

- Explored various strategies to enhance the performance of the Transformer-based model for financial time series forecasting task:
  - Utilized PatchTST as the backbone model with a custom-designed head for univariate and multivariate prediction.
  - Proposed a cross-validation technique for time series data
  - Tested whether incorporating additional features in the training data can improve the Transformer model performance

Conclusion

- Ultimately, none of these strategies are proven to be effective:
  - The models trained on the default train/validation splits consistently outperform those trained on custom splits
  - The models trained solely on price data perform better than those trained on prices as well as additional stock features.

Thank you for attention