EXPERIMENTATION - SHORT DESCRIPTION:

Theoretical usage of **Transformers**-based models for high frequency FX trading.

At this point this draft model is just an experiment showing initial theoretical results to explore the model's general ability to predict profitable trades in the FX market. This experimentation is planned to be continued with further improvements based on initial observations and lessons learned.

LLM Coding Assistance: Throughout the development of this project, various LLMs such as GPT-4o, o1, Claude 3.5 Sonnet, Gemini 2.0 Flash were used as coding assistants for tasks such as code generation and debugging.

VERSION CONTROL

Description document			
Version	File name	Date of upload	Type
V01	README_Experimentation - model_FX_Transformers_v01 - Doc_v01.docx	January 2025	Initial draft

Code & model			
Version	File name	Date of upload	Type
V01	FX_TRANSFORMERS_v01.ipynb	January 2025	Initial draft

CONTENTS

CONCEPT

High level description: using a machine learning model for high frequency FX trading trained on historical hourly data with several technical indicators and some economic fundamental metrics as features.

Goal: to predict profitable actions (Buy, Sell) by forecasting the direction of price change.

Asset class: EURUSD. It typically carries more favorable characteristics for financial modeling compared to other asset classes where non-stationarity with a trend typically being involved; it's less of a concern with FX due to its inherent mean reversion tendency (especially for EURUSD, but broadly applicable to some other major FX pairs too).

Features: wealth of technical indicators and some relevant economic variables (EUR-USD interest rate differentials, Treasury yields, Inflation).

Model type: Transformers; a deep learning model with multi-head attention mechanism.

Evaluation: various accuracy and financial metrics on both training and testing datasets.

Transaction costs: transaction cost is implemented for the evaluation; however, the current model architecture does not include transaction cost in the target variable used for training. This is one of the key areas of further developments.

CURRENT STATE

This is an initial draft version, which aims to provide an infrastructure to experiment with various hyperparameters and set of features in a relatively simple environment with evaluation capabilities. As a next step the model can be enhanced based on the lessons learned and major observations of the results of the current version.

KEY RESULTS

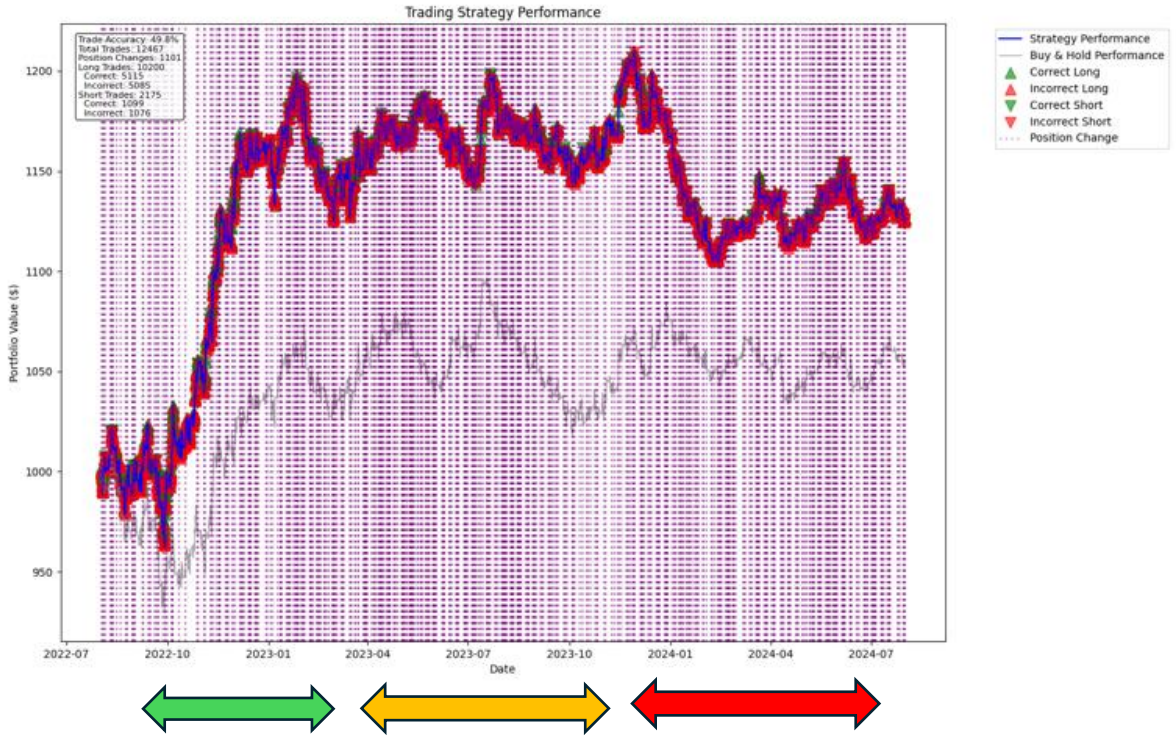
Please note that the results shown below assume a **1 pip transaction cost** for a round trip. The results are highly sensitive to the applied transaction cost, which is a crucial element in high-frequency trading. Depending on the state of the market, the trading volume and other factors, the transaction cost can be significantly higher rendering the results below rather optimistic.

The metrics and charts presented here are the early results of the initial draft version of the current model infrastructure.

Testing results on unseen data: some of the key metrics can be seen below.

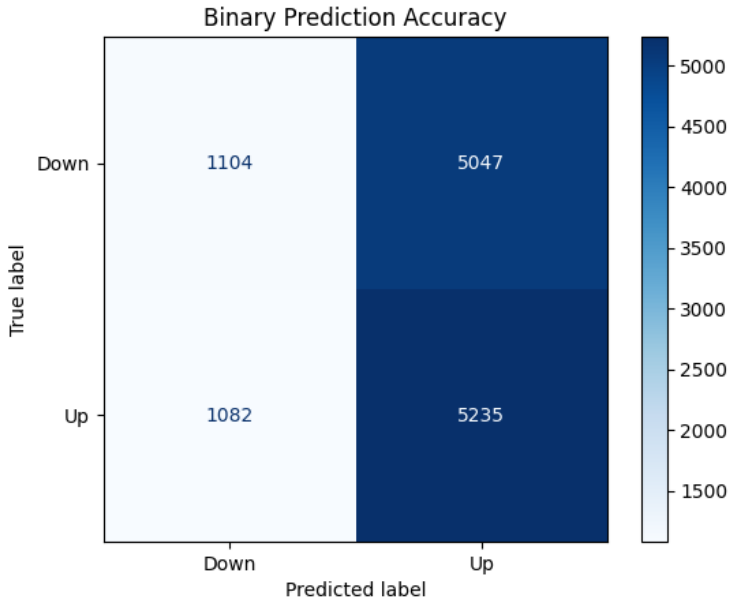
- Initial balance: EUR 1,000
- Testing time window: Aug 2022 – Aug 2024

- Transaction cost: 1 pip for a round trip
- **Annualized return: 5.95%**
- **Sharpe Ratio: 0.46** (assuming a 2% risk free rate)
- Maximum Drawdown: 8.7%
- Annual Volatility: 8.2%
- Total Holding Period Return: 12.6%
- Direction Prediction Metrics (based on actual trading actions, not the original binary predictions, which latter is on a 6 hour forward basis):
    - Accuracy: 49.35%
    - Precision: 49.85%
    - Recall: 82.12%
    - F1 Score: 62.04%

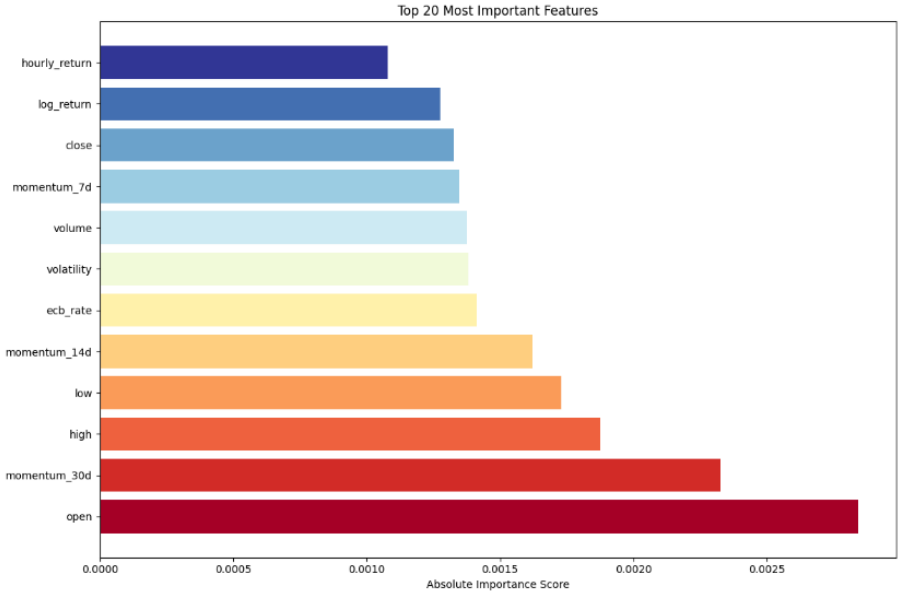


The model demonstrates some capabilities to make profitable trades on the unseen testing time window with a 1 pip round trip transaction cost. The predicted trading actions are skewed towards long trades, which is one of the drivers of the relatively high Recall ratio. The model took advantage of this behavior in the first few months of the testing time window, when substantial positive price moves occurred. On the other hand, however, this is also one of the main reasons why the model struggles with predicting profitable trades for the subsequent months, when the up and down price moves appear to be much more evenly distributed resulting in a substantial number of false positive predictions (Please see Confusion matrix below). Consequently, the model fails to maintain the initial relatively high win rate, which is also embodied in the rather poor overall Accuracy and Precision values. Having said that, a slight edge above 50% in

accuracy could be considered an acceptable model result due to the inherent high noise to signal ratio making financial modeling challenging. One of the main directions of investigation can be identifying the drivers behind the imbalance between long and short trade predictions (Please also see chart in the Appendix about prediction probabilities), which would have the potential to improve accuracy and precision metrics by reducing false positives.



Exploring the set of features contributing to the model performance the most is also a critical component allowing future improvements in model performance by identifying areas where enriching new metrics could present the highest added value potential. In this specific model setup price related features appear to be important drivers alongside metrics such as momentum, volatility and ECB rate. This is also an area to follow-up on and determine if the set of features could be better constructed and optimized.

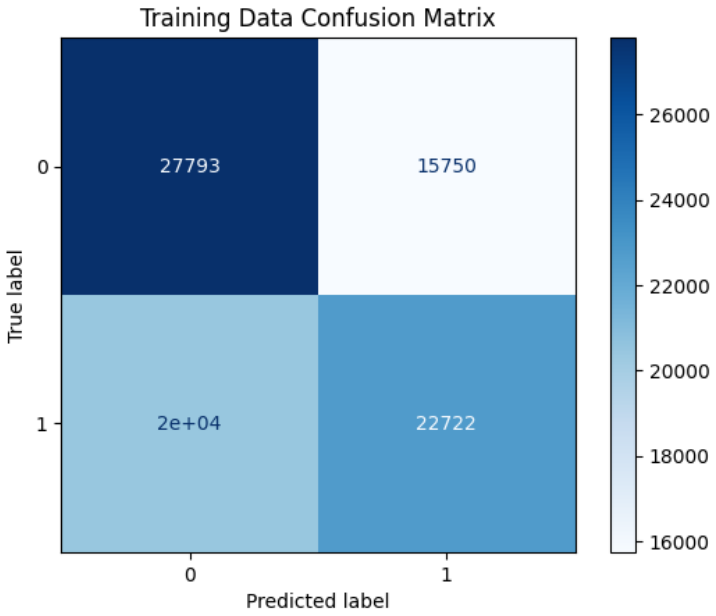


Training results: some of the key metrics can be seen below focusing on Confusion matrix and accompanying accuracy metrics.

- Training time window: Aug 2008 – Jul 2021
- Binary classification prediction metrics on the training dataset:
    - Accuracy: 58%
    - Precision: 59%
    - Recall: 53%
    - F1 Score: 56%

The focal point of assessing the training performance here is the analysis of the Confusion matrix, the balance of long-short trades, the binary prediction accuracy related metrics and the identification of potential over- or underfitting behaviour.

In contrast to the testing dataset, the long and short trade predictions appear to be more balanced in the training time window also resulting in higher Accuracy metrics, however, at the expense of lower Recall ratio (relative to what was observed with the testing metrics). Underfitting potential here is probably more likely than an overfitting tilt based on these statistics. It is, however, set by design to some degree to alleviate the negative consequences of the high noise to signal ratio. The applied regularization techniques to avoid overfitting are the high dropout ratio and relatively short training (low number of epochs). However, several other factors also play a significant role in this such as the learning rate, the batch- and window sizes to name a few. Finding the optimal combination of them given the available compute resources is one of the key areas of further investigation potentially using Parameter Grid Search tools. Overall, the current setup seems to carry more signs of underfitting than overfitting, which is subject to further investigation.



METHODOLOGY OUTLINE INCLUDING DATA PREPARATION

Infrastructure: TensorFlow, Keras, Python.

Model type: Transformers; a deep learning model with multi-head attention capable to capture longer-term dependencies. The number of attention heads in the current setup is four.

Please see paper: Attention is All You Need (Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017).

Target: binary classification of actions based on the average forward log return over a 6-hour time period. If this average forward return is positive, the assigned action is Buy, otherwise Sell. The idea behind using a forward time window as opposed to a one period return is the intention to reduce noise disincentivizing the model to learn on them, which would increase the probability of overfitting also leading to reduced testing performance.

Features: Several technical indicators including among others volatility, momentum, directional movement, moving averages, mean-reversion, price acceleration, trend strengths, volume related metrics. These technical indicators are enriched with fundamental economic variables of FED and ECB rates, treasury yields, inflation rates alongside some rate differential related indicators. The full list of features can be seen in the code. The total number of features is over 100.

Data preparation:

- The main source is IC Markets hourly EUR/USD close price spanning the time frame between 2008 and 2024. The number of data points is about 100k (~90% train and 10% test). The dataset includes basic price information (high, low, open, close, volume, spread).
- The rates related economic indicators are sourced from the ECB and FED official websites. The economic indicators resampled to an hourly frequency to adapt to the trading frequency. NaN values are forward filled to avoid lookahead bias.
- The data for the general macroeconomic variables such as GDP and inflation are sourced from Vantage Alpha (https://www.alphavantage.co/). These are also resampled to an hourly frequency and NaN values are forward filled.
- The technical indicators are calculated using the library Talib (some of the metrics are manually constructed).
- The up and down hourly price directions appear to be balanced, and the range of the price level historically is within a relatively narrow band (unlike in the case of stocks and other asset classes, where significant trend is usually present), rendering the classic financial modelling issue of non-stationarity less of a concern for EURUSD asset. Log-return of the close price nonetheless included as a feature to further mitigate any potential negative impact.
- To handle the high noise-to-signal ratio typically attributed to financial datasets, the 6-hour rolling average of the feature values (where meaningful) are applied encouraging the model to pick up signals rather than to react on noise.

Hyperparameters: The aim is to use values that are mainly within the industry standard ranges with some experiments and adjustments to adapt the given infrastructure and feature profile.

Evaluation: The model is evaluated on both the training and testing time window. The evaluation for the training time window focuses on accuracy related metrics mainly to gauge over/underfitting potential. The evaluation of the testing predictions is more comprehensive and includes three main metric types:

- Accuracy statistics such as precision, recall, F1 score.
- Distribution scanning metrics like the ratio of long and short positions.
- Trading metrics such as portfolio value evolution, Sharpe ratio, maximum drawdown, annual volatility, annual return.

The ideal situation is having a model that produces high risk-adjusted return on both the training and testing datasets capturing general trends appropriately. The assessment is based on the totality of the metric types.

LIMITATIONS & WEAKNESSES

Optimistic transaction cost assumption and no slippage rate: Although a transaction cost is applied for testing model performance, the assumed level may be too optimistic. Additionally, (opportunity) costs associated with potential slippage is currently not in scope of the model infrastructure; this can be an element to add in future improvement rounds.

Representativeness: It could be that due to some market regime shifts and change in trading behaviour/profile over time, the trained model is not sufficiently representative to the testing and potentially live trading environment. Training yet uses data up until 2021 and the model requires vast amounts of data to learn properly. Hence, this is a fine balance to strike between data availability and representativeness.

Underfitting potential: The fundamental building blocks of validation (during training) is already implemented in the current infrastructure; however, it's not yet being used mainly for GPU constraints and training time related reasons. However, based on the accuracy statistics the model performance depicts more signs of underfitting than overfitting.

Features: The list currently does not contain sentiment indicators of economic news, and forecasts for fundamental economic variables, which all have the potential to improve the model accuracy even to a significant extent.

Hyperparameters: Given the nature of neural network-based models, finding the optimal set of hyperparameters is crucial. The currently set parameters may not be close enough to the optimal ones.

High noise to signal ratio: This is an inherent issue with almost every asset in financial modelling. To mitigate the high noise to signal ratio, 6-hour rolling average of the feature values have been applied at the expense of potentially reducing the signalling power of the features.

Compute resources: Due to compute resource related limitations the hyperparameters like the batch size, window size, number of hidden layers were set at levels that may not correspond to the optimal setup.

Set of evaluation metrics: the trading results are assessed based on the portfolio value, Sharpe ratio and similar metrics. No 'baseline' metrics are currently available to compare the model results against. For stock trading such a baseline metric would be the total return with a simple 'Buy&Hold' strategy. Such a metric is less meaningful in FX trading. Nonetheless, performance of a 'Buy&Hold' strategy is added to the evaluation framework for now.

Number of data points: Although the current number of datapoints of about 100k is considered to be sufficient, neural network-based models work better on large datasets

due to better generalization capabilities, which is especially applicable to Transformers-based models (whereas Long Short Term Memory based Recurrent Neural Network-models may also function well on smaller datasets).

FUTURE IMPROVEMENT DIRECTIONS

Also train with transaction costs and implement slippage rate: Transaction cost is currently applied only during evaluation. Besides applying transaction costs during model performance test, the improvement idea is to also train the model with transaction cost included. Implementing thresholds for training based on transaction costs might make it necessary to use three classes as opposed to the current binary classification (besides the classes Buy and Sell, the introduction of Hold may be necessary). That may entail further changes in the model and evaluation infrastructure. Additionally, slippage related opportunity costs should also be part of the total trading costs. These are areas of further exploration as explicitly adding transaction and slippage related cost to the model training is expected to improve the model performance to a substantial extent.

Hyperparameter tuning: The results are highly sensitive to the set of hyperparameters. One direction of improvement could be to use Grid Search to automate the exploration of hyperparameter combinations to find the optimal group of parameters. On the other hand, it should be carefully designed because such a function would entail significant increase in compute resource and training time. Given the testing performance, hyperparameters should potentially support further learning (to reduce current potential underfitting). Experimenting with different number of attention heads could also improve model performance.

Features: additional improvement potential can be to enrich the set of features with more economic data and ideally with sentiment indicators of social and economic news. Additionally, some feature selection mechanism could also be implemented to reduce the highly correlated ones, however, this is less of a concern with neural network-based models compared to 'classic' models like linear regressions.

Training dataset: to increase the number of data points, a data history with a higher frequency (e.g. minute data) could be considered. Alternatively additional FX pairs could be added to the dataset that are expected to behave in a similar fashion to the EUR/USD. The idea behind that would be for the model to find some general relationships that are present for all the major FX pairs. It would bring the benefit of more data points and reduced possibility of overfitting at the expense of likely lower representativeness with respect to individual FX pairs. Finding a careful balance between the two factors can be a sensible, possible step ahead.

Additionally, a more robust data preparation process could be added to the framework with stationarity testing and representativeness assessment functions.

CONFIGURATION

Applied files:

These csv files below are used in the code to source historical price data as well as creating technical and economic features.

- ECB policy rate: file ECB Data Portal_20241228020814.
  - Source: https://data.ecb.europa.eu/data/data-categories/ecbeurosystem-policy-and-exchange-rates/official-interest-rates
- Fundamental economic data: file Alpha_Vantage_econ_data_features
  - Source: Alpha Vantage, https://www.alphavantage.co/
- EURUSD hourly rate: file EURUSD_H1_200806301600_202408232300date_format
  - Source: IC Markets

How to run it:

- Have the underlying csv files stored in a single folder.
- In the code 'FX_TRANSFORMERS_v01.ipynb' replace the file path references with yours in section '1. Configuration setup' at the beginning of the code.
- Set the applied parameters in section '# 4. Setting parameters' at the end of the code:
  - Hyperparameters (e.g. learning rate, batch size, window size)
  - Training parameters (e.g. training start and end data, future period)
  - Trading parameters (e.g. transaction cost)
- Run the Python code. It will save the relevant training and testing files in the reference folder.
- Depending on the available GPU resources, it might be that the model run is completed, but the testing part fails to run due to resource exhaustion. In such a case run the short second code below the main one to see the main test results on the testing time window.

APPENDICES

Testing time window – prediction probabilities and trading signals