Relating Trade Conditions and Behavior of Currency Markets Using Technical Analysis

NAU Northern Arizona University
Office of Undergraduate Research and Creative Activity

Reilley Luedde * Anthony Simard * Christopher R. Keefe * Paul Wagner ‡

Northern Arizona University

* Equal Contributors ‡ Advisors

NAU Northern Arizona University
The W. A. Franke College of Business

Introduction

Background: Financial markets are subject to highly random behavior. The literature indicates that it might be possible to generate long-term positive results by identifying and trading on imbalances that repeat [3][2][1]. The currency market was chosen over other financial markets for its high liquidity (ease of entry and exit), which reduces the impact of "slippage" (differences between desired and market prices). Lower starting capital requirements and broker fees make small-scale automated trading more viable, making currency exchange more accessible to amateur traders.

Objective/Hypothesis: We present a software framework for backtesting technical-analysis-based trade strategies, alongside four strategies implemented by our team and tested against ten years of historical exchange rate data for 28 of the most liquid currency pairs. We hypothesize that successful trade strategies can be built on the principle of momentum, the idea that a change in price will continue in the same direction instead of reverting to the mean.

Data Provenance and Management

Few sources of high-quality, low-cost trade data exist. After extensive research, only one vendor was able to meet our needs. Our data were generated and published by FXCM Group, and accessed through Quandl with a paid, academic subscription. We chose daily Open, High, Low, Close (OHLC) bid/ask data, with volume for this analysis. Currency markets are decentralized, making 'tick' data available only broker-by-broker. Working with OHLC provides a broader look at market behavior in exchange for loss of fidelity. Data quality was high, so cleaning was limited to truncating data to the 10-year window from Sept 14, 2009-Sept 14, 2019

Software Architecture

We use a main model class to provide universal model functions (e.g. data access) to all trade strategies. This allows strategies to focus solely on using indicators to initiate trades. Indicators are also modularized, allowing users to compose strategies or more-complex indicators with them.

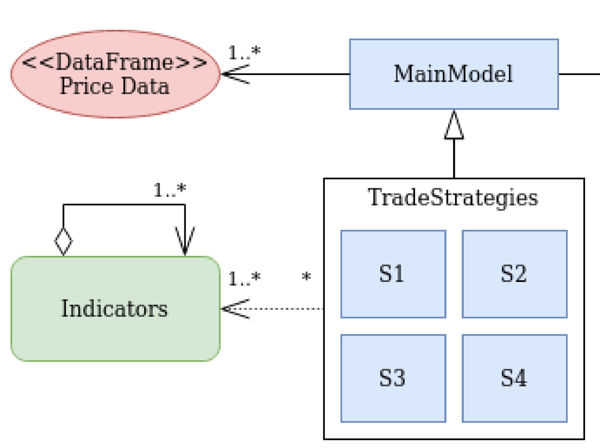


Figure 1: Our software framework allows for straightforward composition of strategies from implemented indicators.

All strategies derive from the MainModel class which provides universal functionality (buys, sells, equity tracking, etc.)

Indicators take a dataframe of the currency data over time and calculate the value of the given indicator on every day

The strategy uses the values of the selected indicators to determine its behavior on any given day

The strategy then calculates its current trade balance before moving on to the next day

```python
from main_model import MainModel
from indicators.alligator import alligator
from indicators.trend_direction_force_index import trend_direction_force_index
from indicators.heiken_ashi import heiken_ashi


class Strat1(MainModel):

    def run(self):
        alligator_df = alligator(self.df)
        TDFI_df = trend_direction_force_index(self.df, 13)
        heiken_ashi_df = heiken_ashi(self.df)

        for date_index in self.df.index:
            date = self.df['date'].loc[date_index]

            if not self.has_position:
                lip = alligator_df['lips'][date_index]
                teeth = alligator_df['teeth'][date_index]
                jaw = alligator_df['jaw'][date_index]

                TDI = TDFI_df['TDFI_ema'][date_index]

                if lip > teeth and jaw < teeth and TDI > .05:
                    self.enter_long(1000, date_index, date)
                elif jaw > teeth and lip < teeth and TDI < -.05:
                    self.enter_short(1000, date_index, date)
            else:
                HA_osc = heiken_ashi_df['difference'][date_index]
                HA_osc_prev = heiken_ashi_df['difference'][date_index - 1]

                if self.trades[-1]['type'] == 'long' and HA_osc < 0 \
                        and HA_osc_prev > 0:
                    self.exit_long(date_index, date)
                elif self.trades[-1]['type'] == 'short' and HA_osc > 0 \
                        and HA_osc_prev < 0:
                    self.exit_short(date_index, date)

            self.calculate_current_model_balance(date_index, date)

```

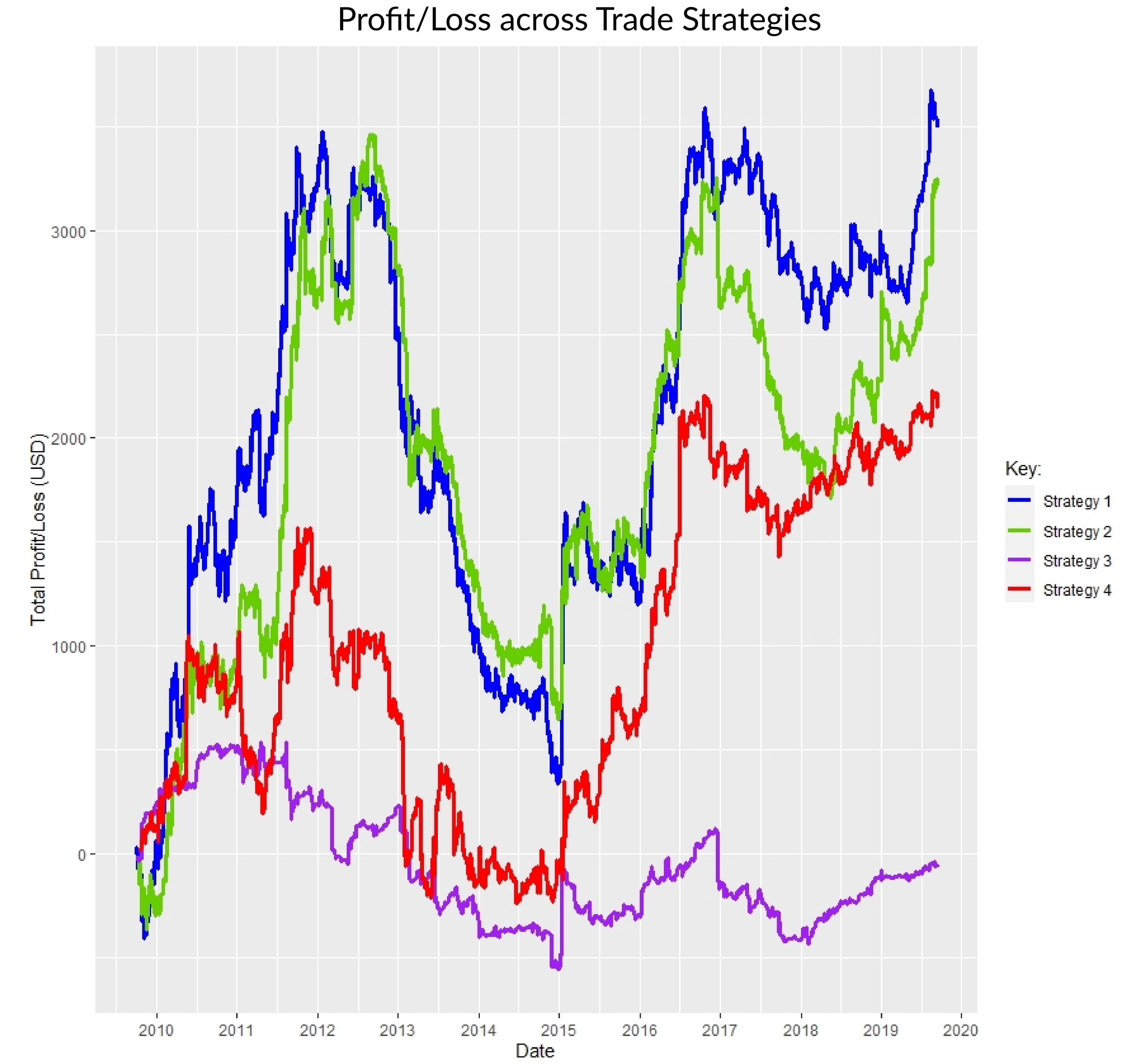
Figure 2: All trade strategies inherit high-level functionality from the Main Model. This reduces code duplication, and encourages consistency in data access and calculation of results.

Trade Model & Strategies

Our trade model externalizes brokerage costs, slippage, and inflation. Our starting equity figure was chosen post facto, a minimal amount that would allow all strategies to trade our ten-year window. Equity amounts and Sharpe Ratios are therefore arbitrary, but useful for intra-model comparison.

We tried to design trade strategies (S1, S2, etc) that capture market momentum. Each strategy uses at least one entry and exit indicator to determine when to open or close a trade. S1 and S2 attempt to enter the market at the beginning of a trend and exit when the trend weakens. S3 attempts to determine when a market is under/oversold, and trade accordingly. When the market exhibits behavior approaching randomness, S3 exits trades. S4 enters when volatility increases (risk and opportunity both increase), and exits on decreasing volatility.

Profit/Loss across Trade Strategies



Implemented Market Indicators

Indicator	Exit/Entry	Summary
Simple Moving Average (SMA)	Both	Calculates a rolling average of price
Average True Range (ATR)	Both	Determines a metric of market volatility
Exponential Moving Average (EMA)	Entry	Calculates a rolling average of price; recent prices are given more weight
Alligator	Entry	Determines trend direction fr
Trend Direction Force Index (TDFI)	Entry	Determines strength of trend based on magnitude of market activity
Money Flow Index (MFI)	Entry	Determines if a market underbought/undersold based on market activity
Heikin Ashi Oscillator (HAO)	Exit	Smoothes OHLC data to determine weakening of trend
Chandelier Exit	Exit	Determines exit based on volatility
Random Walk Index (RWI)	Exit	Determines how closely a market resembles randomness

Trade Strategy Implementations

Strategy	Entry Indicators	Entry Conditions	Exit Indicators	Exit Conditions
1	Alligator, TDFI	Shortest-term Alligator MA becomes most extreme |TDFI|>0.05	HAO	HAO crosses 0
2	Alligator, TDFI	Shortest-term Alligator MA becomes most extreme |TDFI|>0.05	Chandelier exit	Price crosses Chandelier Exit
3	MFI	MFI > 80 -> long, MFI < 20 -> short	Random Walk Index	Values Cross
4	SMA, EMA(ATR)	ATR > EMA(ATR) & Price > SMA -> long, ATR > EMA(ATR) & Price > SMA -> short	EMA(ATR)	ATR < EMA(ATR)

References

[1] Hayward, R.
Foreign exchange speculation: An event study.
International Journal of Financial Studies 6 (02 2018), 22.

[2] Menkhoff, L., and Taylor, M.
The obstinate passion of foreign exchange professionals: Technical analysis.
Journal of Economic Literature 45 (12 2007), 936--972.

[3] Shmilovici, A., Kahiri, Y., Ben-Gal, I., and Hauser, S.
Measuring the efficiency of the intraday forex market with a universal data compression algorithm.
Computational Economics 33 (02 2009), 131--154.

Project Funding

NAU Northern Arizona University
HURA Programs

Results

- S1, S2 & S4 were net profitable over the test period (externalizing fees). S1 was the most performant strategy per unit risk, based on Sharpe Ratio*.
- S2 had the largest expected trade size, but was also the most volatile (highest SD).
- The largest trades for S1, S2 & S4 were near-identical.

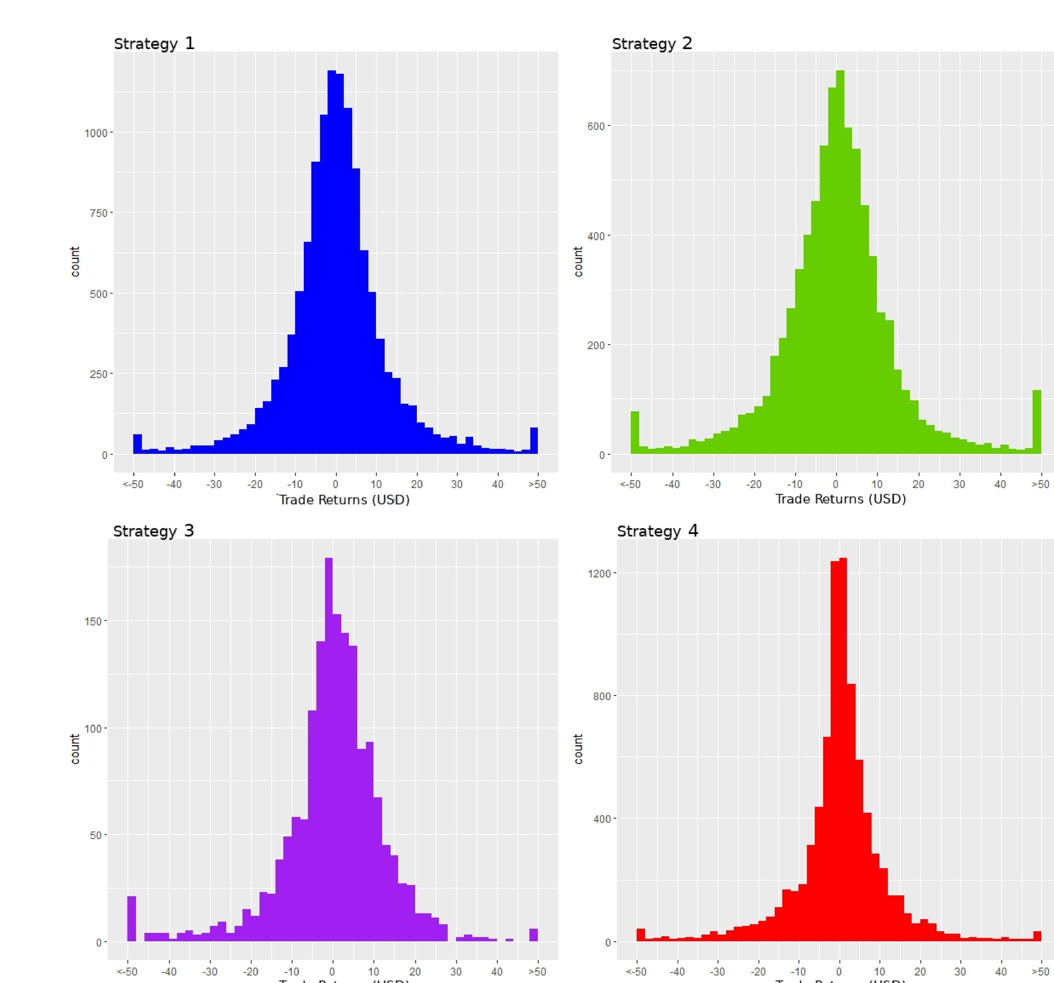
Trade Strategy Performance

	Strategy 1	Strategy 2	Strategy 3	Strategy 4
n	12061	7789	1663	8177
EV/Trade	$0.29	$0.41	($0.04)	$0.26
SD	$14.21	$17.59	$15.50	$12.82
Sharpe Ratio*	1.366	1.173	(.027)	1.072
% Return	223.96%	206.40%	(4.16)%	137.40%
Starting Equity	$1560.00	$1560.00	$1560.00	$1560.00
Final Equity	$5053.74	$4779.84	$1495.04	$3703.49
Min	($139.85)	($212.90)	($130.46)	($186.61)
Max	$211.71	$211.36	$199.80	$211.26

Figure 3: All currency values normalized to USD. Parentheses indicate negative values.

*Characteristics of our model prohibit the comparison of Sharpe Ratio with strategies in the wild, but this metric is useful for contrasting within-model performance.

Profit-Loss Distribution



Conclusions

The externalities of this preliminary model make our results inconclusive, but the success of S1, S2, and S4 within the model environment suggest that momentum-based trade strategies may provide a profitable direction for further study. Our software architecture will allow us to scale the number of trade strategies tested gracefully, allowing us to run more, better-target tests.

Our results are tightly coupled to the data window we chose, illustrating the impact of selection bias and model overfitting. Shorter data windows might have shown these strategies to be highly successful or unsuccessful, based on selection. Bootstrap testing these strategies against many randomly-selected windows might better inform strategy robustness. Validation against data from other time periods and currency pairs will increase our confidence in our results.

S1, S2, and S4's near-identical max-value trades occurred simultaneously, when those strategies were short during the 2015 "Swissie Flash Crash". Though likely coincidence, the ability of these models to capitalize successfully on a major market shift deserves further study, and could support the utility of technical analysis trade models in responding effectively to rapid market change.

Future Work

- **Improve model performance:** This preliminary model externalizes brokerage fees, market slippage, and other critical factors. Further development will better quantify profitability.
- **Test Additional Strategies:** Refine specific sategies: momentum and mean reversion, especially.
- **Hourly Data:** Transitioning to hourly instead of daily OHLC data would be straightforward with our framework, and would improve model fidelity