

BIRKBECK COLLEGE, UNIVERSITY OF LONDON

MSc. DATA SCIENCE PROJECT PROPOSAL

The Applicability of Machine Learning Methods For Currency Trading

Author

Edward GILL

Supervisor

Dr. George MAGOULAS

September 5, 2019

This proposal is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I give my permission for it to be submitted to the JISC Plagiarism Detection Service. The proposal may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

The availability of immense computing power has opened a veritable Pandora's box of highly complex and computationally intensive models which can now be applied to any dataset, whether appropriate or not. While machine learning models have led to rapid and impressive use cases across a host of different fields of research, this paper will focus on one oft researched area, global financial markets and specifically currency markets.

Applying complex methods within the financial domain will require a different approach from other datasets given the inherent non-stationarity and noisiness of financial time series. The aim of this project is to build on existing research and try to gauge the efficacy of applying machine learning methods to the investment process. This paper will explore various machine learning techniques to assist with the problem of currency forecasting across various time frequencies, whether by using a combination of models to handle different aspects of the trade generation process or as an outright prediction model.

Contents

1	Introduction	4
2	Background	4
3	Aims and Objectives	5
3.0.1	Data Retrieval, Cleaning and Standardisation	5
3.0.2	Feature Extraction	6
3.0.3	Model Training and Architecture Selection	6
3.0.4	Model Validation and Strategy Backtesting	6
3.0.5	Testing on Fully Withheld Data	6
4	Pitfalls of Machine Learning in Trading	6
4.1	Heteroskedasticity in Financial Markets	8
5	Building Blocks of Currency Trading Strategies	10
5.1	Drivers of Currency Markets	10
5.1.1	Economic Fundamentals	10
5.1.2	Price Momentum	10
5.1.3	Investor Positioning	10
5.1.4	Currency Valuation	10
5.2	A Multi Factor Model For Currency Prediction	11
5.3	The Myth of Machine Learning in Finance?	11
6	Machine Learning Model Overview	12
6.1	Artificial Neural Networks	12
6.1.1	Background Research	13
6.2	Support Vector Machines	13
6.2.1	Background Research	14
6.3	Ensemble Methods: Random Forests	14
6.3.1	Background Research	15
7	Proposed Methodology	16
7.0.1	Objective 1: Data Capture, Cleaning and Standardisation	17
7.0.2	Objective 2: Feature Extraction	18
7.0.3	Objective 3: Model Training/Fine Tuning and Architecture Selection . . .	18
7.0.4	Objective 4: Model Validation	19
7.0.5	Objective 5: Strategy Backtesting on Withheld Data	19
7.1	Performance Evaluation	19
7.1.1	Results Across All Model Iterations	19
7.1.2	Cross Validation	19
7.1.3	Problem of Dimensionality	20
7.1.4	Out-of-Sample Testing on Unseen Data	20
7.1.5	Simplicity Trumps Complexity	20
7.2	Assessing Model Performance	20
7.2.1	Mean Squared Error	20

7.2.2	Precision and Recall	20
7.2.3	Deflated Sharpe Ratio	21
8	Proposed Work Plan	22
8.1	Time Allocation and Potential Risks	24
8.1.1	Poor Feature Selection - <i>high risk</i>	24
8.1.2	Under Allocation of Resources - <i>moderate risk</i>	24
8.1.3	Data/Procedural Errors - <i>moderate risk</i>	24
8.1.4	Data Consistency - <i>moderate risk</i>	24

1 Introduction

The velocity of technological change that society has encountered since the turn of the millennium is striking, the world has embarked on a computational revolution which has brought the availability of immense processing power to the masses. Computational power at a scale that government sponsored computer science programs of the early nineties could only dream of. In fact, the processing power of most smartphones are larger than the supercomputers of yesteryear [1].

Moore’s Law, the well known law which states that the number of transistors in a dense integrated circuit doubles about every two years, has held for over fifty years [2].

The allure of applying machine learning methods to finance is well entrenched, where researchers dream of finding the holy grail of a repeatable pattern within the data which leads to guarantees of consistent alpha generation (returns unexplained by conventional market beta factors [3]). A cursory search of Google, another revolution of big data processing and clustering algorithms, uncovers a plethora of academic papers which profess of having found patterns in the data that could only have been uncovered by complex models with non-linear capabilities.

Using machine learning models in conjunction with big data can and do provide fantastic results in the area of the biological sciences or as recommender systems for movies or advertisements. However applying these same methods to financial markets is a different and more challenging task due to the underlying unpredictability of asset prices.

This proposal will firstly outline prior research undertaken alongside some of the more common pitfalls that can impact model performance and model generalisation. The rest of this proposal dives into the data to be used, model building framework, main pitfalls which make applying complex algorithms in finance susceptible to out of sample failure [4] and also a brief analysis of the potential machine learning methods to be used.

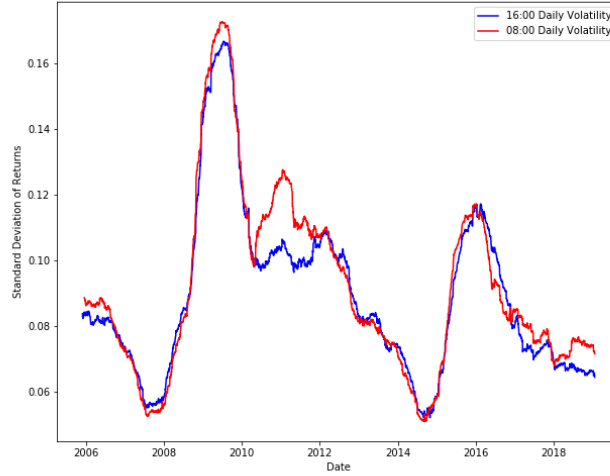
2 Background

The task of uncovering repeatable patterns in financial data remains incredibly precarious, financial markets are non-stationary, exhibit leptokurtic (fat tailed) distributions and can shift regimes at highly unpredictable speeds. Figure 1 shows the 1 year rolling volatility of EURUSD (with daily snapshots created at 8am and 4pm). We can clearly see how volatility remains non constant over time and that even the time of day used to calculate the volatility can allow large deviations in volatility to persist.

Much has been written on the complexity of financial markets and how researchers must overcome such dynamic systems [5]. While the explosion of increased computational power aided the application of machine learning in very useful tasks such as speech recognition and image processing, the applicability of these methods in finance are, on the face of it at least, questionable [6].

In the era of big data, financial markets are one domain which has some of the smallest datasets available, as one important factor is in short supply, time. Financial timeseries have relatively little history when compared to other fields of study. Most accurate data capture goes back to the early nineties and unless the domain is high frequency tick data, then there is generally sparse data to test with when compared to some non-financial data sets such as movie recommendation algorithms, which have many millions of users each with their own individual movie meta data attributes [7].

Figure 1: EURUSD Rolling Annualised Volatility Since 2002



Source: EURUSD FX Rate, Kaggle Database

3 Aims and Objectives

The main aim of the project is to understand how machine learning methods can be applied to currency forecasting and the classification of macroeconomic regimes in which a particular model (or combination of models) may perform best.

The project will try to outline best practices to use when applying machine learning techniques to currency trading, this may take the form of an outright prediction model or a set of models which also forecast shifts in market regimes. This will allow us to combine information from various sources to provide the highest probability of out of sample trading success.

The main objectives in order to achieve the stated aim of the project are shown below.

3.0.1 Data Retrieval, Cleaning and Standardisation

- One of the most important aspects of a trading model is to ensure that the underlying data is of high quality and doesn't contain erroneous values. The proposed model will use hourly spot exchange rates for G10 currencies ¹, various economic data from the Federal Reserve Economic Database (FRED), the International Monetary Fund (IMF) and the Organisation for Economic Cooperation and Development (OECD) will be used to try to uncover exploitable patterns in the data.
- This data will also need to be cleaned and mapped to correct date scales so that the model only uses data available at the time of trade signal generation. The raw data itself will need to be standardised to allow the model to understand the true relationships between the feature vectors and the output.

¹Hourly FX Spot Data Provided by Citi Foreign Exchange

3.0.2 Feature Extraction

- Feature extraction for financial time series prediction is a very challenging task and given the inherent noise in the data, including features which have an intuitive justification for predicting future prices is important [8]. Feature selection driven by mining the training set greatly increases the risk of overfitting and should be avoided when working with financial times series.

3.0.3 Model Training and Architecture Selection

- The project itself will test the efficacy of a range of algorithms in trying to predict currency movements, this project will also make use of non price data in order to assist the model in understanding potential regime changes in the trading environment.

3.0.4 Model Validation and Strategy Backtesting

- Once a model architecture has been selected, it is important to understand not only classification rates but also how the model would perform in a live market setting. This can be approximated by creating a backtesting framework which depicts the model's trading performance and also includes the impact of transaction costs.

3.0.5 Testing on Fully Withheld Data

- While the test data allows us to understand how the model performs, there will also be an opportunity to analyse how the selected model will work on new unseen data which would provide a useful guide as to how the model will generalise in the future.

4 Pitfalls of Machine Learning in Trading

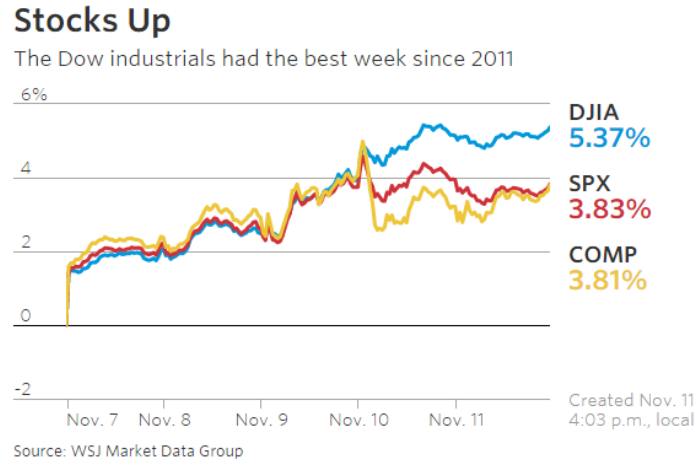
The problem to overcome when applying ML to trading is the level of signal to noise in the data. To try to differentiate the problems ML practitioners face in the financial domain vs others, lets look at an example. Take the case of image processing, if a neural network is fed a matrix of pixels which form the image of cats and dogs, and then trains based on the historical patterns it observes, the trained neural network will be able to identify cats and dogs with high classification rates. However, if we provide an image of a rhino, the model likely classifies the rhino as a large dog due to the historical patterns that have been learned, hence it can only learn what it has trained on historically.

Similarly with the Alpha Go project [9] at the company DeepMind, the developers created an excellent model which would calculate all possible combinations of moves to find the most optimal route to victory. This model convincingly beat the worlds best human Go players [10].

That extremely well built model is beholden to the past patterns it has learned over millions of iterations playing within the well defined rules of the game of Go. In finance, which one could assume is somewhat akin to playing the ancient game of Go (or more closely related would possibly be poker) you can learn all the historical patterns and probabilities surrounding the asset you are trading. This model will likely work very well even in the testing period. However, in finance the rules can change rapidly, causing structural breaks to occur in past relationships. Alpha Go does not need to worry about the impact of a 280 character tweet from Donald Trump entirely changing the rules of the game.

For example, in the run up to the 2016 US presidential election, equity markets would sell off due to increases in Donald Trump’s odds of winning the presidency. The market was horrified to see a president who openly criticised western allies, spoke of protectionism and of building walls against major trading partners.

Figure 2: US Equity Market Performance During US Election Week



Source: Wall Street Journal, Nov 2016

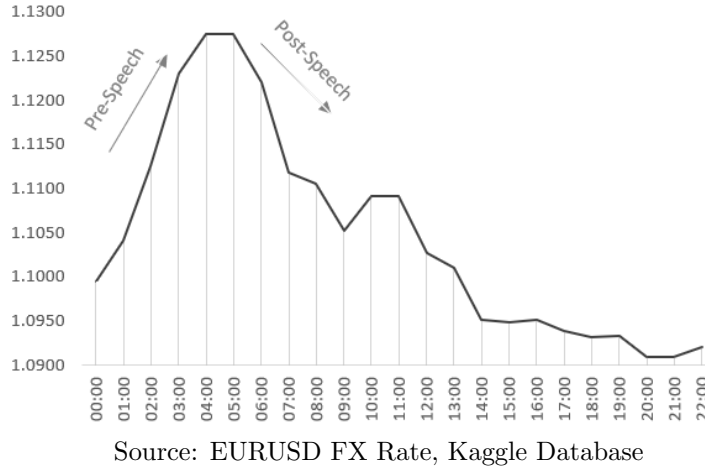
However, post his initial winning speech (which was much less controversial and business friendly) markets fully reversed any losses and ended 5% higher (figure 2). Thus even models that had predicted a Trump win (and subsequent market crash) and positioned as such based on the historical correlations in the lead up to election day would have incurred severe losses. Currency markets were also impacted, with EURUSD rising nearly 3% as the election result was announced, before subsequently falling by an even larger amount after Trump’s conciliatory speech (figure 3).

Correlations and patterns in markets can be extremely fickle, complex and dynamic in nature [11]. Why is this so one may ask, well part of the issue lies in the efficiency of informational flow across markets, barriers to entry are much smaller than ten or twenty years ago and there has likely been a compression in the informational edge that smart money investors (such as hedge funds) previously have had [12]. The market price itself is a reflection of all these moving parts and thus even if a robust pattern has been found, it is likely to be arbitrated away as more market actors step in to trade away any mis-pricing [13].

One issue that is particularly pertinent in the financial domain is the issue of data dimensionality and the underlying size of each individual data set, even if there were fifty years of daily financial data, this is less than 20,000 data observations², thus while in other fields of research there may be millions of data points for categorisation and training, this is not the case for finance. This means that one must think carefully before adding features to the model. There are hundreds of potential predictors of a financial asset (in fact Chatzis et al. used 180 predictors when trying to forecast financial market crisis events [14]), and this means that due to the

²Assuming a 252 trading day calendar

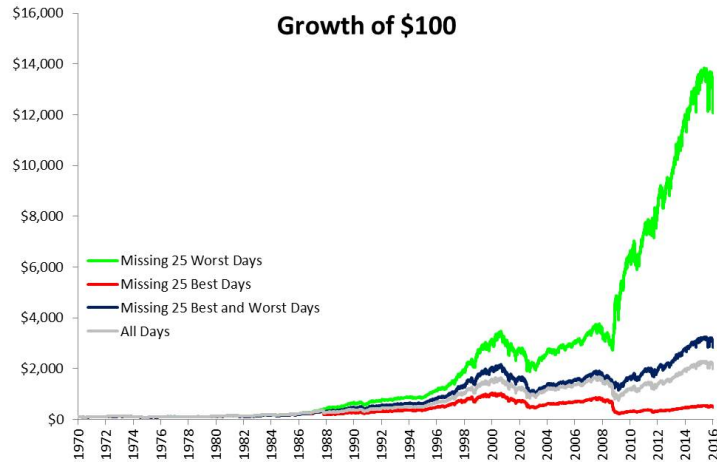
Figure 3: EURUSD Price Action During 2016 US Presidential Election Results



number of possible variations, the probability of finding a strategy by luck remains high. Arnott et al. provide an example of a trading strategy which would have passed even the most stringent evaluation methods in both training, validation and testing phases which actually turned out to have been created using a ranking of stocks based on alphabetical ordering [8].

4.1 Heteroskedasticity in Financial Markets

Figure 4: Growth of \$100 Invested Since 1975



Complex systems that continuously evolve make applying ML techniques in a way that results in a robust model very difficult [8]. Financial markets have been shown to exhibit heteroskedasticity in their returns [16] and as such, creating a robust systematic trading strategy relies on more than just a simple data input/signal output type of architecture. In trading

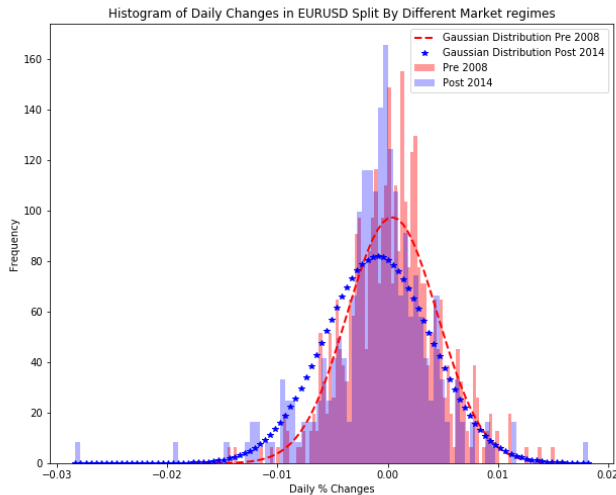
systems, classification of regimes and subsequent buy or sell signals must also be compared vs the expected value of the that trade. A model with an 80% classification success rate will under perform if it mis-classifies the the 20% of samples which result in large moves in the asset.

The corollary of that is by only targeting classification of outliers you reduce your sample set and thereby reduce the statistical validity of the results. There is rarely a free lunch in financial markets and as such there will be no free lunch when using machine learning methods to predict financial times series.

For example figure 4, shows that if an investor were to miss out on the best 25 days of the Standard and Poors 500 companies equity index (S&P 500) over the last fifty years, the reduction in returns would have been very severe [15]. Given that 25 trading days represents only 0.005% of the total trading days , returns are significantly impacted. This shows the importance of being correctly positioned for outlier events and also the fact that one model can be severely impacted by one off or black swan type events [17]. Previous research has centered on price based data possibly as it is easily available and also because it is one dataset that is likely to have higher frequency data available ([18], [19], [20]).

In fact, even when analysing the daily changes in EURUSD , we can see that the distribution of future returns can shift quickly, and the regime a model has learned in can end up shifting so that the model's future predictions under perform.

Figure 5: Distribution of Currency Returns Shifts According to Different Market Regimes



Source: EURUSD FX Rate, Kaggle Database

For example, figure 5 splits the daily distribution of EURUSD percentage returns into two different regimes, one in 2008 vs a second regime in 2014, each market regime results in a shift in the distribution of the currency and thus when training a model, one must be cognizant of the impact the wider market regime can have on a models performance.

5 Building Blocks of Currency Trading Strategies

One area which seems under researched (from the prior background research performed), is to use lower frequency economic data to assist in a model's prediction accuracy. While price based metrics offer the easiest route to higher frequency data, they also lack the ability to capture the fundamental economic factors that can drive currency direction [21].

5.1 Drivers of Currency Markets

Firstly, before we analyze feature selection methods, it is also important to try to understand what consists of a good trading signal, whether discretionary or systematic in its generation. Below we showcase factors that can impact currency direction.

5.1.1 Economic Fundamentals

One of the most important factors that sets the broad direction of a currency pair is the relative economic performance between two countries. Dahlquist et al. show that measuring relative economic momentum between different countries has the ability to forecast currency direction over medium term horizons [22]. That paper showed statistically significant results that economic fundamentals have an important role in relative capital flows and subsequent currency direction.

5.1.2 Price Momentum

Price momentum here refers to the historical trend in the currency, research does suggest there exists exploitable momentum within the currency space [23]. The reasoning for the existence of momentum in currencies is likely due to decision delays amongst different investor segments who build up positions in currencies over time. Auto-correlation in currency returns may also exist as information is priced slowly over time (such as economic data) and thus as more economic data is published, this adds further confirmation to a trading signal and thus conviction and trade size increases.

5.1.3 Investor Positioning

Currency positioning relates to the magnitude of investor holdings of a currency. Positioning and currency flows can have informational value that impacts a model's performance [24]. For instance, if the finalised model triggered a buy recommendation on a currency, but if the wider market had already built up a sizable long position in that currency, then the likelihood of further buyers coming in to drive the price higher is diminished.

5.1.4 Currency Valuation

Uncovering the true fundamental fair value of a currency has been the focus of international academic research for many years [25], and it remains an elusive and rather ambiguous task as currency drivers can undergo structural shifts. Purchasing Power Parity currency levels, which rely on the Bassala Samuelson effect and the law of one price [26] are one method of trying to anchor the value of a currency to one fundamental price level. While currencies can deviate

from fair value, there is some evidence that developed market³ currencies tend to mean revert in the long term [27].

5.2 A Multi Factor Model For Currency Prediction

While the above drivers of currency markets are not exhaustive, it can provide a frame work for constructing a systematic trading strategy. Currency drivers can shift over the short and medium term and while this can all be partially captured by price action, it is also prudent to try to understand the dynamics of other potenial drivers as discussed in the previous section. Therein lies the main problem when combining machine learning and finance, that other areas such as the biological or physical sciences do not encounter, is that at the core of each price, are human or human created trading algorithms which have collectively decided on a price level at which to transact.

Every price of a trade in financial markets consist of the dreams, fears and greed of human emotion. Not only this, but market investors are influenced by past patterns and incorporate new information (including newly published research) into future decisions, thus altering the environment in which the historical pattern has been learned. Bartram at al. provide evidence that anomaly (excess) profits published in research papers tend to decay substantially in the period after publication, which suggests any new findings are arbitraged away⁴ quickly as the new information is incorporated into the market [13].

While early academic research espoused the efficient market hypothesis, there are certain times where the market is not fully rational [28]. Thus opportunities do exist, but finding repeatable opportunities requires more than just analysing price patterns in data. Given the randomness of markets, even poor strategies can perform well for long periods of times due to sheer luck. This creates new challenges for practitioners of machine learning in finance, in fact, new research has also started to focus on protocols and best practices around data mining in finance [8].

5.3 The Myth of Machine Learning in Finance?

Given the inherent challenges faced when developing systematic trading strategies, one may wonder if creating a strategy based on machine learning methods is a fruitless endeavor.

Due to the unique issues related to applying machine learning in the financial domain, this paper proposes to test various models with the goal of finding a robust methodology for extracting features which also have the ability to generalise into the future. Once the results and performance of each model has been obtained, it is then vital to compare and contrast how each model performs both in the training and validation phases.

To ensure we can test the selected model fully on new data, the last one year of test data will be truncated from the data until all validation and testing phases have been completed, this allows us to see how the finalised model would perform on fully unseen data even after initial testing and calibration phases across all the models selected.

³Developed Market Currencies consist of USD, EUR, JPY, CHF, GBP, AUD, NZD, NOK , SEK, CAD

⁴Of course a more sinister reason could be model overfitting on historical data

6 Machine Learning Model Overview

Below we provide a brief overview of various models we will test in our analysis and that have been utilised in previous research.

6.1 Artificial Neural Networks

Artificial Neural Networks (ANN) were conceptually first written about in the early 1940s [29] when neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper on the theoretical functioning of neurons. Donal Webb then expanded on this in his paper "The Organization of Behavior" while the first computerised neural networks were developed by IBM in models called "ADALINE" and "MADALINE" [29]. At their core, ANNs are a series of large matrix multiplications where weights and biases, which represent the strength of connections between nodes, are optimized to provide the best approximation to the desired output.

The popularity of ANN methods have grown as enhanced computational power led to the lowering of barriers to entry in the field. One of the more popular networks used in practice is the mutlilayer perceptron, initially introduced by Frank Rosenblatt in 1958 [30]. A perceptron is the simplest form of a layered network and consists of a single neuron with an adjustable weight. The simplest form of the perceptron can divide the hyperplane into a 2-dimensional space which is then used to classify the given inputs. The hyper plane is defined by the linearly separable function below.

$$\sum_{i=1}^n x_i w_i - \theta = 0 \quad (1)$$

Multi Layered Perceptrons (MLP) can be defined as

$$NeuralNet_j^l = \sum_{i=1}^{N_l} w_{ij}^{l-1} y_i^{l-1} \quad (2)$$

where $NeuralNet_j^l$ is the l th layer of the j th neuron which gives us the weighted sum of its inputs. The weights of the connections to the next layer $l = 1$ are used to calculate the weighted sum of inputs for the subsequent layer in the network.

$$Y_j^l = ActivationFunction(NeuralNet_j^l) \quad (3)$$

This weighted sum of inputs is then passed through to an activation function which maps the output of the hidden layer to a space defined by the activation function, in this step we can introduce non linearity into the model itself by using a non linear activation function such as the sigmoid function described below.

$$y = \frac{1}{1 + e^{-x}} \quad (4)$$

This maps any series of values onto a plane of $[-1,1]$ and the output depends on the values produced by the node which feeds the activation function.

While there are a vast array of model training techniques , one method to train an ANN is to use backpropagation, where updates to the network's weights are fed backwards through

each layer. Each model has an error function which is used to grade performance, which could be the squared difference between the models output and the desired output (mean squared error).

The universal approximation theorem [31] of the MLP model in neural networks states that ANNs can approximate any function and thus in theory, should be able to identify any patterns within the data that another method would also identify.

The temptation to introduce multiple layers with thousands of connections in order to find a solution is ever present, however (and especially in the realm of finance) this increases the risk of over-fitting or selecting a model based on sheer luck, which fails when tested on live unseen data.

6.1.1 Background Research

ANNs have been used extensively in academic financial journals, Czekalski et al. made use of feedforward multilayer perceptrons to predict the direction of the EURUSD currency rate [32], while Wang et al. applied probabilistic neural networks and Least Square Support Vector Machine models to assist in the prediction of Chinese equities [20]. Literature on using ANNs to forecast currencies has a twenty year history as even at turn of the millennium Joarder et al. tested three separate ANN based forecasting models using standard backpropagation, scaled conjugate gradient and backpropagation with Bayesian regularisation to predict movements in various Australian dollar currency crosses [33]. Gunduz et al. tested a variety of ANN models and Support Vector Machines (SVM) to try to predict the price of equities traded on the Istanbul Stock Exchange, they uncovered model classification accuracy rates consistently around 70% on test data [34].

The majority of research cited make use of various technical indicators which are derived directly from the price series, for example, Gunduz et al. make use of ten price based technical indicators to uncover patterns within Turkish equities [34]. While on average most prior research suggested above 70% classification rates were attainable on average. In contrast to this, there has also been some research to suggest that ANNs often show inconsistent performance on noisy data ([35], [36], [37]), which highlights the challenges in applying complex models as the risk of overfitting the data remains an ever present threat.

6.2 Support Vector Machines

One model which can be used to uncover repeatable patterns in financial data are Support Vector Machines (SVM) and Support Vector Regressions (SVR). SVMs were initially introduced by Cortes and Vapnik [38] which led to the development of non-linear models which provide binary classifications of target vectors while SVRs output real-valued predictions. In similar fashion to an ANN, the SVM tries to classify an n-dimensional feature vector according to where that data lies on an n-dimensional space.

The model makes use of the kernel trick [39] to transform the original data into a higher dimensional space such that we can then uncover a separable hyperplane. The function updates the parameters of the kernel function such that it can more accurately separate and classify the data points. The error correction model in the case of a SVM or SVC (Support Vector Classifier) is a maximisation problem where the model tries to maximise the distance between all the separating hyper planes (maximal margin classifier) it has constructed. The solution to the optimisation problem can take the form of the following

$$\text{Maximise}_{\beta_0, \beta_1, \dots, \epsilon_1 \dots M} M_{\text{margin}} \quad (5)$$

Subject to

$$\sum_{j=1}^p \beta_j^2 = 1 \quad (6)$$

$$y_i(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_j x_{ij}) \geq M(1 - \epsilon_i) \quad (7)$$

such that $\epsilon_i \geq 0$ and $\sum_{i=1}^N \epsilon_i \leq C$. Where M is the width of the margin to be maximised, β_i denotes the weight of the coefficients and x_i is the i^{th} observation ($i = 1, 2, \dots, N$) of the dataset, ϵ_i is a variable whose value sets the degree to which individual observations are allowed on the wrong side of the margin or hyperplane. C is a non negative tuning parameter which sets out the total error allowed for observations to be on the wrong side of the margin [40].

6.2.1 Background Research

Kara et al. utilised SVMs to discover patterns in Turkish equities achieving classification success rates of around 60% when tested on hold out data [41]. Both polynomial and radial basis SVMs were trained on a host of price based technical data with the radial SVM outperforming the polynomial SVM. The polynomial SVM also had a much higher variability in classification success rates and thus the radial basis may be best suited to predicting financial time series. The results of the radial SVM were significant with a test statistic of 3 and p-value of 1%.

Similarly, Patel et al. tested a selection of four different machine learning methods (ANN, SVM, Random Forest and Naive Bayes) using ten technical indicators [42]. The predictor space was also transformed into trend deterministic states, which map the original continuous values into discrete values based on the trend of the continuous valued indicator (downward trend = -1, upward trend = 1). Results suggested that model performance is improved when trend deterministic data is used instead of continuous data. They postulate that the models perform better on the trend deterministic data as this transformation inputs the underlying trend of the indicator which then allows the model to learn the relationship between the underlying predictor trends and the actual output trend.

SVM classification rates using this approach (for both radial and polynomial kernels) approached nearly 90% on average. Huang et al. also tested the performance of an SVM model (radial kernel) to predict Nikkei futures prices⁵ and achieved classification rates of 73% on test data. The authors also found that combining model forecasts increased out-of-sample test results [18]. This is an architecture that will be tested in the project as well. Overall, prior research has found significant results for using SVMs as a method to predict future time series such as equities ([43], [19], [35]) and currency movements [44].

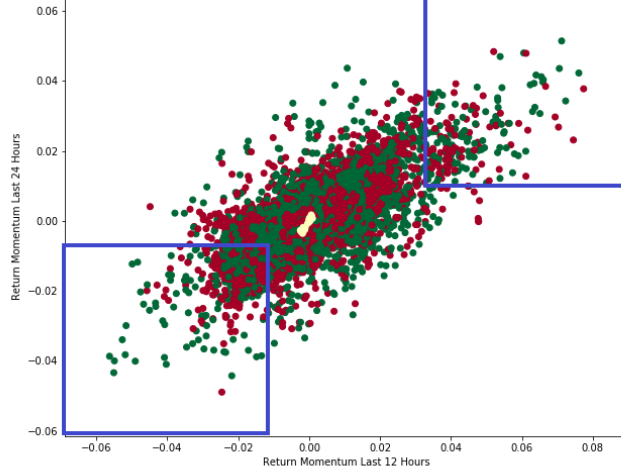
6.3 Ensemble Methods: Random Forests

Ensemble methods are a technique in machine learning which uses the outputs of multiple models to form a consensus around the prediction, while there are many different techniques one can

⁵Nikkei 225 is one of the main Japanese Stock Market Indices

use, this paper proposes to explore whether ensemble methods can be used to reinforce the accuracy of the model on unseen data. One approach is to use tree based methods [45], which involve segmenting the predictor space into a number of regions. Figure 6 looks at the prior 12 hour (x-axis) and 24 hour (y-axis) return of EURUSD with the subsequent 24 hour ahead move in the currency colour coded from red (negative return) to green (positive return). Even in such

Figure 6: Tree Based Methods Segment the Predictor Space into Numerous Regions



Source: EURUSD FX Rate, Kaggle Database

a simple split of the predictor space, we see some evidence that more subsequent positive days occur in the bottom left of the chart (when prior 12 and 24 hour moves in the currency have been negative). Tree based methods use this type of inference to make predictions on future data points. The goal of the decision tree is to find multiple regions which then minimise the squared error of predictions

$$\sum_{j=1}^J \sum_{i=1}^{R_j} (y_i - y_{R_j}^*)^2 \quad (8)$$

where $y_{R_j}^*$ is the mean response for the training observation within the j^{th} region. The methods tested in the project will revolve around Random Forests (introduced by Leo Breiman [46]) which averages the prediction across hundreds of single decision trees.

6.3.1 Background Research

Research has shown that Random Forests can generalise better than single decision trees as they maintain high variance while reducing the bias of the model [47]. Random forests use bootstrapping to take multiple random samples from the training data and a random subset of feature vectors to create a diverse array of decision trees which aid model generalisation. This method helps to create decision trees which are uncorrelated to each other and thus have a

lower probability of overfitting the data. Patel et al. showed that it was possible to achieve classification rates of over 80% for various stock price predictions when using the random forest algorithm and input features which consisted of technical price based indicators [42]. The research used a random selection of three feature vectors (out of a total of ten) to create each single decision tree in the random forest model and tested the results from 20 to 200 trees using a majority vote for the final classification.

Kumar et al. performed extensive tests to compare the performance of support vector machines to random forests and found that on average support vector machines outperformed random forests when predicting the direction of the Indian stock exchange [36]. Three predictors were considered for each node while the number of trees for each model ranged from 200 to 1500. There was no significant reduction in the error rate for the out-of-sample data based on the number of trees. The RF model achieved classification rates of over 67% on out of sample data. Chatzis et al. found that RF methods outperformed SVMs and logistic regression techniques when predicting stock market crisis events, they also note that RF models are relatively robust to overfitting due to each forest only being exposed to a subset of the available feature vectors [14]. The allure of using a decision trees (DT) ensemble method is that those models on an individual basis have low bias but high variance. However combining these models in an ensemble fashion can help to reduce the probability of overfitting, while improving model generalisation on unseen new data.

7 Proposed Methodology

The following section dives into the methodology to achieve the main project aim and objectives that were outlined in section 2. One area of research where there looks to be a gap in the research is to use a combination of models to handle different aspects of the trade signal generation process. This has the potential to perform well, as one model alone is unlikely to generalise into the future, especially when applied to the the domain of trading (where structural breaks can occur).

Neural Networks, by the nature of their construction, need a large training set to understand the patterns within the data, thus this type of model could assist in the creation of higher frequency signals while a separate model can capture the market regime.

However, as we have mentioned previously, patterns in financial data remain exceedingly complex and the performance of one model could be due to a particular market regime. For example, regimes of strong investor risk appetite tend to be an environment conducive to the positive performance of higher yielding currencies regardless of underlying country specific fundamentals [48].

Thus for a model to be successful, it must be capable of capturing changes in signal dynamics across different market regimes. This can be achieved by feeding features which also allow us to capture risk appetite, overall economic performance or mean reverting price patterns.

As pointed out by Mclean et al. [49], financial model performance can differ vastly prior to publication than after it⁶, whether due to over fitting or post publication anomaly decay [13], this means we must be extra vigilante when assessing model performance.

The Python programming language (Python 3.6, [50]) will be used as the main programming application to achieve the objectives, with the python packages of Numpy [51], Scipy [52], Sci-Kit Learn [53], Matplotlib [54] and Pandas [55] all providing very useful data manipulation functions

⁶Research suggested a 26% fall in performance out-of-sample and 58% decline post publication

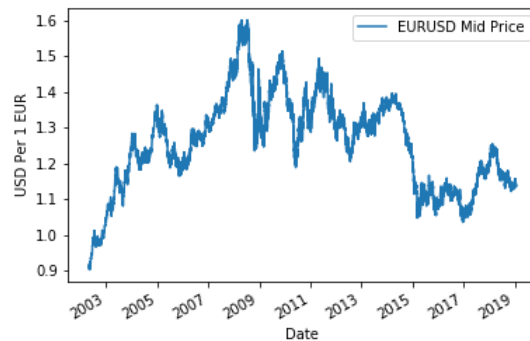
to help store, clean and process data. For more complex ANN learning, the machine learning package Keras [56] will be used in the training and testing of ANN models.

7.0.1 Objective 1: Data Capture, Cleaning and Standardisation

The initial step will be to source intraday hourly price data for a selection of currencies⁷ and related economic and investor flow data. Figure 7 shows the performance of EURUSD over the last sixteen years formed of hourly snapshots⁸.

There is evidence of the existence of momentum in currencies, and using higher frequency data can allow us to explore different time frames which have the most consistent momentum [57]. Economic data can be obtained from a variety of sources, with the Federal Reserve Economic

Figure 7: EURUSD Price Action



Source: EURUSD FX Rate, Kaggle Database

Database (FRED)⁹, IMF and OECD all providing vast treasures of low frequency economic data to create feature vectors. Relative economic growth (as measured by momentum across indicators such as consumer spending, GDP growth, industrial production and business confidence) has been shown to assist in the prediction of currency movements [22].

The data will be stored and manipulated using Pandas dataframes while Sci-Kit Learn has some very useful functions such as StandardScaler which can standardise data using various different techniques (z-scoring, min-max etc).

Standardising the data remains an important initial step that has been used across a wide variety of prior research. Subtracting the mean and dividing by the standard deviation (z-scoring) [58], min-max standardisation¹⁰ ([34], [36]) and other techniques such as mapping functions to scale feature vectors onto a pre-defined scale ([41], [20], [42]) all help to achieve the overall goal of scaling, which is to ensure that larger or more volatile features do not overwhelm features of smaller magnitude and lower volatility.

Any errors or missing data will need to be rectified and as such the data will be tested for

⁷Analysis will involve developed market currencies of EUR, USD, JPY, GBP, SEK, NOK, AUD, NZD, CAD, CHF

⁸circa 90,000 observations

⁹Successfully used in [14] to assist with forecasting stock market crises

¹⁰Map observations onto a scale of [0,1] with 0 being the minimum value of the observation and 1 denoting maximum

outliers which are impossible (i.e. negative currency values) or do not fall within appropriate bands (i.e. 100% percent jumps in currency values between hourly time frames). Cleaning of the data will be broken down into outlier detection (to capture any erroneous inputs) and cross checking the data vs known events in time (such as Brexit, election dates) to see if the observed outliers match the expected time frames. This provides some clarity on the time snapshot of the data as well. The time series will be transformed to a GMT date timescale so that there are no mis-matched values and to ensure the signal lags have been appropriately adhered to. Sci-Kit Learn also provides functions to handle unbalanced datasets and anomaly detection which will be used to assist in cleaning the raw data [59].

7.0.2 Objective 2: Feature Extraction

When working with financial time series, the feature selection process should be driven by economic intuition as this can lessen the probability of over-fitting [8].

As patterns and correlations in prices can change quite quickly, it is important not to overfit the model on the training data. A transformation of the data is also needed, with some findings suggesting that performing principal component analysis to create lower dimensional feature vectors is a good technique to use when constructing predictive models [60]. The majority of previous research used price-based technical indicators ([41], [20], [42], [34]) such as moving average crossovers, rate of price change, average true range (ATR) and exponential moving averages to try to predict future currency direction. While technical indicators can be useful to forecast currencies, it is also important to measure non price based information such as economic momentum and valuation. The proposed feature set is outlined in table 1.

Table 1: Model Features

Indicator	Description	Type
Short Term Momentum	5,10, 21, 55 Period Moving Average	Technical
Long Term Momentum	100, 155, 200, 255 Period Moving Average	Technical
Mean Reversion	Standard Deviation from 55, 100, 200 Period Moving Average	Technical
Volatility	55, 100, 200 Period Rolling Volatility	Technical
Relative Economic Momentum	1 Year Economic Momentum	Economic
Global Economic Momentum	1 Year Global Economic Momentum	Economic
Purchasing Power Parity	Deviation From Fair Value	Fundamental Valuation

7.0.3 Objective 3: Model Training/Fine Tuning and Architecture Selection

The model training stage allows the learning algorithm to understand and capture the relationship between the features and the classification target vector (the target vector will be the future direction of the asset price across various time periods, both regression and classification models will be tested). Prior research has generally used binary classification of the future asset price move when developing models([61], [34], [14]).

Model training will also allow us to tune parameters to help understand which combination of models (ANN, RNN, PNN ,SVM, RF) and underlying architecture (hidden layers, number of decision trees) are best suited to capturing the nuances in the data. This stage of the process will also allow experimentation with various models to see the differences in performance. Various Python packages such as Sci-Kit Learn and Keras will be used in order to create and test the classification success rates of each type of model. Fischer et al. made use of Keras (on top of

Google’s Tensorflow) to implement Recurrent Neural Networks (RNN) [58] while using Sci-Kit Learn for logistic regression, SVM and Random Forest methods.

Testing how combinations of models perform will show how this stacked type of model architecture can assist in model generalisation and boost overall predictive performance ([14], [62]).

7.0.4 Objective 4: Model Validation

In order to validate the model, the selected model and its structure will need to be tested in a similar setting to real world trading. This requires the construction of a backtesting engine which can simulate each model’s performance taking into account transaction costs and the funding costs associated with trading currencies. This will provide a much more accurate analysis of the selected trading strategy performance¹¹. While classification success rates will be important in gauging model accuracy, high classification rates may not lead to positive strategy performance if large currency moves are mis-classified.

7.0.5 Objective 5: Strategy Backtesting on Withheld Data

While back testing on training and validation data can assist us in finding appropriate parameters for each model, it is important to also test the finalised model on fully withheld data as this allows us to understand how the model may perform in the live trading phase. Further analysis of model validation techniques are discussed in detail in the next section.

7.1 Performance Evaluation

Analysing and outlining model performance will be one of the key objectives to be satisfied in the final stages of the project. In light of the findings from ([8], [6]), which highlight in detail the pitfalls inherent in systematic trading, the below outlines an approach to reviewing and evaluating model performance to standards generally set in financial institutions.

7.1.1 Results Across All Model Iterations

Keeping track of the number of models being tested is very important as it can help define the statistical likelihood of finding a strategy that is a true positive. Even given 20 randomly selected strategies, there is a non-negligible probability that one strategy will beat the threshold needed for a statistical significance [8]. Thus, the number of trials run must be considered so we can deflate the statistical significance of the strategy performance as a function of the number of trials [63].

7.1.2 Cross Validation

Splitting the dataset into various training and test sets will be very important in being able to assess model performance. The most common cross validation techniques are K-fold cross validation and Leave-One-Out cross validation. This project will use a training set, validation set and finally a test dataset which will be held out until the final model has been selected. This will provide a useful gauge of out of sample performance¹².

¹¹The vast majority of research papers referenced in this proposal did not perform this type of testing

¹²In depth study of cross validation techniques outlined in [64]

7.1.3 Problem of Dimensionality

In finance, the lack of sufficient data (price) vs the number of possible predictors does lead to the issue of dimensionality, where we can add further features but not necessarily add further data points to be tested. Thus the model proposed here will remain cognizant of the number of features being used and will try to reduce the probability of finding false positive trading strategies. The dimensionality of the current data set will involve one target vector (i.e. future price move) and 17-20 predictor columns as outlined in 1.

7.1.4 Out-of-Sample Testing on Unseen Data

As stated by Arnott et al [8], selection of model inputs is inherently biased (especially when done by domain experts) as the experts use heuristics to choose what they find the most economically intuitive as predictors, thus even out of sample test results are not fully out of sample. To override some of these concerns, this paper proposes to omit one years worth of data from any analysis until the model and related architecture has been fully completed and the selected model or model architecture has been finalised. Then the chosen model will be blind tested on the final year of data that has been held back to give the best approximation of how the chosen model might have performed.

7.1.5 Simplicity Trumps Complexity

The goal of this project is to try to create a model capable of generalising into the future, therefore it is important to choose an architecture which remains the simplest in terms of construction and feature sets. All models should be compared against a simple linear model as a benchmark, to test whether the value added by the increased complexity and reduced interpretability is substantial enough to overcome the drawbacks.

7.2 Assessing Model Performance

7.2.1 Mean Squared Error

The mean squared error (MSE) is used to measure the prediction accuracy of the model. It looks at the average distance of the predicted variable from the actual value.

$$MSE = \frac{1}{N} \sum_{i=1}^n (y_i - F(x_i))^2 \quad (9)$$

where y_i is the actual value and $F(x_i)$ is the value predicted by the function when the i_{th} observation is passed to the function. Normally it is best practice to choose the model which has the lowest test MSE.

7.2.2 Precision and Recall

Another method of model validation is to look at the precision and recall of the model [42], whose values can be calculated from a confusion matrix, which shows the number of correct and incorrect classifications per class. Table 2 shows the confusion matrix for a two class classification problem.

Table 2: Confusion Matrix

Actual (row) Predicted (col)	Buy	Sell
Buy	True Positive	False Negative
Sell	False Positive	True Negative

The confusion matrix allows us to calculate the precision and recall of the model, the formulas for which are shown below.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (10)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegatives} \quad (11)$$

The precision tells us how well the model has performed in terms of the number of positive signals (correct buy signals as a percentage of all buy signals) that were actually triggered. Recall depicts how accurate the model has been as a function of the total number of positives in the data set itself. This is an important metric to analyse when predicting a financial time series as certain periods can be heavily biased towards one class or another, for example in the case of a financial asset which has a strong trend will by design have a higher occurrence of one class vs another.

7.2.3 Deflated Sharpe Ratio

Given the inherent issues of overfitting in finance ([8], [4]), it is also important to analyse the overall Sharpe Ratio [65] which measures the return of a trading model per unit of risk (or volatility of returns).

High Sharpe ratios can be obtained by a genuinely good prediction model or through pure overfitting/selection bias, thus it is important to deflate the Sharpe Ratio of every strategy tested to take into account the number of model tests as well as the skew of each strategy's distribution [66]. The Deflated Sharpe Ratio is a test statistic which can outline the probability of having found a true positive. It is defined below as

$$\widehat{DSR} = \left[\frac{\widehat{SR} - \widehat{SR}_0 \sqrt{T-1}}{\sqrt{1 - \gamma_3 \widehat{SR} + \frac{\gamma_4 - 1}{4} * \widehat{SR}^2}} \right] \quad (12)$$

where $\widehat{SR}_0 = \sqrt{V[\widehat{SR}_n] \left((1 - \gamma) Z^{-1} \left[1 - \frac{1}{N} \right] + \gamma Z - 1 \left[1 - \frac{1}{N} \exp(-1) \right] \right)}$ and \widehat{SR} is the estimated Sharpe Ratio to be tested, T is the sample length, γ_3 is the skewness of the returns distribution and γ_4 is the kurtosis of the selected strategy, $V[\widehat{SR}_n]$ is the variance across the number of strategy Sharpe Ratios tested, N is the number of independent trials of strategies run (i.e. the number of models tested) and Z is the cumulative function of the Gaussian distribution¹³.

¹³Further information and theoretical proofs of the Deflated Sharpe is available from [66]

8 Proposed Work Plan

The next page will outline the necessary steps to undertake the work needed. In total, this will require just over 300 hours of work, from the end of April until the deadline date in September 2019.

Table 3: Proposed Work Plan

Objective	Task	Description	Start Date	Completion Date
1	Data Retrieval	Source data from various public and institutional databases	April 20 th	April 26 th
	Data Cleaning	Screen data for outliers	April 27 th	May 5 th
	Truncation	Split data into appropriate training and testing sets	May 5 th	May 6 th
	Standardisation	Transform and center data	May 6 th	May 7 th
2	Choose Target	Choose variable to be classified	May 8 th	May 9 th
	Feature Selection	Select intuitive features for the model	June 14 th	June 30 th
	Model Preprocessing	Choose the features appropriate for each model	July 1 st	July 3 rd
3	Architecture Selection	Analyse various model combinations	July 12 th	July 28 th
	Model Training	Run various training iterations	July 12 th	July 28 th
	Model Tuning	Run various model architectures	July 12 th	July 28 th
4	Model Validation	Cross validation across models	July 28 th	August 10 th
	Backtesting	Run trading signals through backtester engine	August 12 th	August 20 th
	Bench-marking	Compare final model to linear/trivial classifier	August 21 st	August 25 th
5	Blind Testing	Run finalised model on blind data	August 26 th	August 26 th
	Performance Evaluation	Analyse performance statistics	August 27 th	September 3 rd
	Final Project Report	Collate results and findings	August 27 th	September 15 th

8.1 Time Allocation and Potential Risks

Table 3 provides a guideline towards the time allocated to each task, the chronological order of the tasks may change as the project develops, it may be necessary to go back and add more features or implement the model architecture differently. Some risks to the timeline of events are listed below.

8.1.1 Poor Feature Selection - *high risk*

By the stage of model training, it may become evident that the original features selected have little predictive power, to solve this, new data sources could be utilised or innovative transformations made on the existing data so that new avenues of the relationship between the predictor and dependent variables are uncovered.

8.1.2 Under Allocation of Resources - *moderate risk*

It may become evident over the course of the project that too little computing resources have been allocated to train each model, this may require the use of cloud based GPUs ¹⁴ to speed up the learning process and the production of meaningful results.

8.1.3 Data/Procedural Errors - *moderate risk*

Performance of the results will be skewed by any errors in the data selection or model training process, one pitfall to avoid is lookahead bias. This means we must use data that would only have been available at the initial time of classification. Thorough data checks and attention to detail should help to reduce this risk.

8.1.4 Data Consistency - *moderate risk*

Going forward, there could be data access issues, that the data does not exist in a consistent format or that the database where the data was previously available is removed from public use. The solution to this would be to search for new databases with similar information or to continue ahead with data already retrieved.

¹⁴Available from Google Cloud or Amazon Web Services

References

- [1] S. Bookman, “15 Supercomputers Less Powerful Than Your Phone.” <https://www.theclever.com/15-huge-supercomputers-that-were-less-powerful-than-your-smartphone/>, 2015. [Online; accessed 02-April-2019].
- [2] C. A. MacK, “Fifty years of Moore’s law,” in *IEEE Transactions on Semiconductor Manufacturing*, 2011.
- [3] R. Rebonato, “Asset Management: A Systematic Approach to Factor Investing,” *Quantitative Finance*, 2017.
- [4] M. Lopez de Prado, “The 10 Reasons Most Machine Learning Funds Fail,” *SSRN*, 2018.
- [5] B. Lebaron, “Chaos and Nonlinear Forecastability in Economics and Finance,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 1994.
- [6] D. H. Bailey, J. Borwein, M. Lopez de Prado, and Q. J. Zhu, “Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance,” *SSRN*, 2013.
- [7] I. Portugal, P. Alencar, and D. Cowan, “The use of machine learning algorithms in recommender systems: A systematic review,” 2018.
- [8] R. D. Arnott, C. R. Harvey, and H. Markowitz, “A Backtesting Protocol in the Era of Machine Learning,” *SSRN Electronic Journal*, 2018.
- [9] D. Silver and D. Hassabis, “AlphaGo: Mastering the ancient game of Go with Machine Learning,” 2016.
- [10] D. Silver, “The story of AlphaGo so far.” <https://deepmind.com/research/alphago/>, 2016. [Online; accessed 14-April-2019].
- [11] S. Camargo, S. M. Queirós, and C. Anteneodo, “Bridging stylized facts in finance and data non-stationarities,” *European Physical Journal B*, 2013.
- [12] N. Kaissar, “Hedge Funds have a Performance Problem.” <https://www.bloomberg.com/opinion/articles/2016-03-24/hedge-funds-have-a-performance-problem>, 2016. [Online; accessed 04-April-2019].
- [13] S. M. Bartram, L. Djuranovik, and A. Garratt, “Currency Anomalies,” *SSRN Electronic Journal*, 2019.
- [14] S. P. Chatzis, V. Siakoulis, A. Petropoulos, E. Stavroulakis, and N. Vlachogiannakis, “Forecasting stock market crisis events using deep and statistical machine learning techniques,” *Expert Systems with Applications*, 2018.
- [15] M. Batwick, “How missing out on 25 days in the stock market over 45 years costs you dearly.” <https://www.marketwatch.com/story/how-missing-out-on-25-days-in-the-stock-market-over-45-years-costs-you-dearly-2016-01-25>, 2016. [Online; accessed 31-March-2019].

- [16] A. Corhay and A. T. Rad, "Conditional heteroskedasticity adjusted market model and an event study," *Quarterly Review of Economics and Finance*, 1996.
- [17] N. N. Taleb, "Black Swans and the domains of statistics," *American Statistician*, 2007.
- [18] W. Huang, Y. Nakamori, and S. Y. Wang, "Forecasting stock market movement direction with support vector machine," *Computers and Operations Research*, 2005.
- [19] S. Shen, H. Jiang, and T. Zhang, "Stock market forecasting using machine learning algorithms," 2012.
- [20] S. Wang and W. Shang, "Forecasting direction of China security index 300 movement with least squares support vector machine," in *Procedia Computer Science*, 2014.
- [21] B. Yoann, "The Truth Nobody Wants to Tell You About AI for Trading." <https://hackernoon.com/https-medium-com-supernova-the-truth-nobody-wants-to-tell-you-about-ai-for-trading-5d29a297ee93>, 2019. [Online; accessed 31-March-2019].
- [22] M. Dahlquist and H. Hasseltoft, "Economic Momentum and Currency Returns." 2015.
- [23] T. J. Moskowitz, Y. H. Ooi, and L. H. Pedersen, "Time series momentum," *Journal of Financial Economics*, 2012.
- [24] L. Menkhoff, L. Sarno, M. Schmeling, and A. Schrimpf, "Currency momentum strategies," *Journal of Financial Economics*, 2012.
- [25] K. Rogoff, "The Purchasing Power Parity Puzzle," *Journal of Economic Literature*, 1996.
- [26] F. Hassan, "The price of development: The Penn-Balassa-Samuelson effect revisited," *Journal of International Economics*, 2016.
- [27] M. Ca' Zorzi, J. Muck, and M. Rubaszek, "Real Exchange Rate Forecasting and PPP: This Time the Random Walk Loses," *Open Economies Review*, 2016.
- [28] B. Döme, "Efficiency and Return Predictability." 2008.
- [29] B. Widrow and M. A. Lehr, "30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation," *Proceedings of the IEEE*, 1990.
- [30] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, 1958.
- [31] V. Kůrková, "Kolmogorov's theorem and multilayer neural networks," *Neural Networks*, 1992.
- [32] P. Czekalski, M. Niezabitowski, and R. Styblinski, "ANN for FOREX forecasting and trading," in *Proceedings - 2015 20th International Conference on Control Systems and Computer Science, CSCS 2015*, 2015.
- [33] K. Joarder and R. A. Sarker, "Forecasting of currency exchange rates using ANN: A case study," in *Proceedings of 2003 International Conference on Neural Networks and Signal Processing, ICNNSP'03*, 2003.

- [34] H. Gunduz, Y. Yaslan, and Z. Cataltepe, "Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations," *Knowledge-Based Systems*, vol. 137, pp. 138–148, 2017.
- [35] K.-j. Kim, "Predicting Stock Price Direction using Support Vector Machines," *Neurocomputing*, 2003.
- [36] M. Kumar and M. Thenmozhi, "Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest." 2006.
- [37] K. J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Systems with Applications*, 2000.
- [38] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, 1995.
- [39] D. Wilimitis, "The Kernel Trick in Support Vector Classification." <https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f>, 2018. [Online; accessed 07-April-2019].
- [40] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. 2013.
- [41] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange," *Expert Systems with Applications*, 2011.
- [42] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques," *Expert Systems with Applications*, 2015.
- [43] B. Gui, X. Wei, Q. Shen, J. Qi, and L. Guo, "Financial time series forecasting using support vector machine," in *Proceedings - 2014 10th International Conference on Computational Intelligence and Security, CIS 2014*, 2015.
- [44] J. Kamruzzaman, R. Sarker, and I. Ahmad, "SVM based models for predicting foreign currency exchange rates," 2004.
- [45] V. Podgorelec and M. Zorman, "Decision Tree Learning," in *Encyclopedia of Complexity and Systems Science*, 2015.
- [46] L. Breiman, "Random Forrests," *Machine learning*, 2001.
- [47] R. Genuer, "Variance reduction in purely random forests," *Journal of Nonparametric Statistics*, 2012.
- [48] C. Christiansen, A. Ranaldo, and P. Söderlind, "The time-varying systematic risk of carry trade strategies," *Journal of Financial and Quantitative Analysis*, 2011.
- [49] R. D. Mclean and J. Pontiff, "Does Academic Research Destroy Stock Return Predictability?," *Journal of Finance*, 2016.

- [50] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 1976.
- [51] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, “The NumPy array: A structure for efficient numerical computation,” *Computing in Science and Engineering*, 2011.
- [52] D. R. Tobergte and S. Curtis, “Learning SciPy for Numerical and Scientific Computing Second Edition,” *Journal of Chemical Information and Modeling*, 2013.
- [53] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. 2017.
- [54] M. A. Wood, *Python and Matplotlib Essentials for Scientists and Engineers*. 2015.
- [55] M. I. Reiff, “Pandas,” *Journal of Developmental and Behavioral Pediatrics*, 2002.
- [56] F. Chollet, “Keras: The Python Deep Learning library,” 2015.
- [57] I. Filippou and M. P. Taylor, “Common Macro Factors and Currency Premia,” *Journal of Financial and Quantitative Analysis*, 2017.
- [58] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *European Journal of Operational Research*, 2018.
- [59] B. Schölkopf, “Sci-Kit Learn: Novelty and Outlier Detection.” https://scikit-learn.org/stable/modules/outlier_detection.html, 2019. [Online; accessed 06-April-2019].
- [60] F. Song, Z. Guo, and D. Mei, “Feature selection using principal component analysis,” in *Proceedings - 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, ICSEM 2010*, 2010.
- [61] G. Abreu, R. Neves, and N. Horta, “Currency exchange prediction using machine learning, genetic algorithms and technical analysis,” 2018.
- [62] J. Stock and M. Watson, “Forecasting with Many Predictors,” *Handbook of economic forecasting*, 2010.
- [63] D. H. Bailey and M. Lopez de Prado, “The Sharpe Ratio Efficient Frontier,” *SSRN*, 2011.
- [64] C. Bergmeir and J. M. Benítez, “On the use of cross-validation for time series predictor evaluation,” *Information Sciences*, 2012.
- [65] W. F. Sharpe, “The Sharpe Ratio,” *The Journal of Portfolio Management*, 2009.
- [66] D. H. Bailey and M. Lopez de Prado, “The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting and Non-Normality,” *SSRN*, 2014.