NUS National University of Singapore

ISS INSTITUTE OF SYSTEMS SCIENCE

Intellitrader

Masters of Intelligent Systems

Capstone Project

Ajay Vikram Singh(A0020986B)

Rana Bhattacharjee(A0195178N)

Abstract

Context: Financial markets asset pricing has been a subject of immense research for being able to predict has immense financial rewards. Financial Assets include Foreign exchange (FX), bonds, equities etc.

Financial assets prices are influenced by various influencing factors covering economic, political, social, and regulatory dynamic. Financial assets trading with a higher temporal prediction accuracy leads to financially superior trade outcomes.

Aims: We aim to predict FX for next day trading session – i.e. Predict today (T0), tomorrow's /next trading day/rolling 24 hrs FX rate (T0+1). We try to predict USD/Euro fiat currencies as they constitute the most traded and dynamic currency pair.

The objective is to create a prediction system that helps market participant – i.e. a FX broker. Focus is on prediction accuracy Vs specific broker platforms. Specific broker platforms will be identified later, based on the quality of their APIs, features offered and trade execution.

The goal of the project is to achieve high enough next day prediction accuracy for a successful trading position over a period of time.

Methods: We collected historical FX transaction data over several preceding years. We defined and engineered features from the raw datasets that heuristically synthesize and summarize information. We set the transaction data (/last week /last fortnight /last month /last quarter) sets. We researched a variety of statistical and Deep Neural Networks based models. We didn't use text data as part of model building for commercial reasons. The pipelines were duly built.

We trained multiple deep learning models and received varying prediction accuracy.

Overall, we built the complete pipelines for data ingestion, data cleaning, feature engineering, ongoing training, so that a commercial trader can be served daily feeds for next day's trading session.

Results:

Conclusions:

Model has successfully learnt predicting next day FX. While we will back-tested and took care to avoid overfitting, the model performance underperforms our expectations for being used as a decision support system for a trading desk.
Our Prediction model lacks incorporation of market news and market sentiments. Surveying various literatures as well as applying our own experience, we see that incorporation of same should improve the prediction accuracy substantially.

We aim to improve the model further by exploiting two areas:

a) Embed trading community sentiment and context News besides trading data.
b) Experiment further novel information extraction networks

Key words:

Trading – USD/Eur – FX prediction – Traded data (Ticker data) - Sentiment analysis – Feature Extraction – Statistical forecasting - Neural networks – Prediction accuracy

Introduction

Trading in the FX markets (as well as other markets e.g. Equity, Fixed Income) follows two broad schools of thoughts and consequent analysis to support Trading (Short, long, Spot buy, Spot sell) decisions:

1. Fundamental analysis (macro-economics, asset micro-economics i.e. it's financial health and asset growth estimates)
2. Technical analysis (historical market charts and technical patterns)

The quest is always on to develop a robust system-driven models that outperform the indices (average indicator of asset class market). Decision Automation is key as it reduces the emotional aspects in trading (e.g. confirmation bias), thus making model more robust.

Technical analysis has been a step in the direction, however this is primarily determined from historical market prices, with a basic assumption that the price contains all the information available including human psychology of market participants leading that price in a given temporal instant. However, today's technical analysis models have a piecemeal approach of factoring various contributors to price prediction including incomplete capture of human sentiments and behaviour. As a result, most of technical models are NOT able to extract "signal" in an inherently low "signal/noise" ratio temporal problem space.

Importance and Applications

Financial and technical analysis have been used to good advantage by traders and investor alike. The key elements that both rely on is years of human learning encoded in rule books, rule engines as well as using human ingenuity [traders, investors, money managers] to make profits [Buy – Sell].

Ability to synthesize data, learn from same and predict asset pricing with reasonable accuracy can provide a competitive advantage to the investor or trader.

Application of our model is with trade desks for institutional or individual foreign exchange traders. The application is in three (3) forms:

a) Closed loop Auto-trading system – relying fully on the prediction engine and letting it decide buy / sell trades based on existing positions
b) Assisting existing trading systems – augmenting existing rule engines in place to assist trade buy/sell decision
c) Providing a single pane of glass – providing informative view

We see that the logical approach for our application will be to start with c) and arrive at b). Achieving a) will mean a wilding profitable corporation in its own right and shall require significant monetary, time, and resource roadmap.

Surveyed Approaches

Survey of existing works

We surveyed existing works and literature in FX trading as well as equity trading. The purpose was to evaluate best practices in data driven decision making. Various books have been written on successful data driven trading [1] [2]

We spread our analysis in following dimensions:

a) Qualitative data and Quantitative data: Here we looked at what type of news qualifies as relevant to trading [3], features of importance in quantitative data etc
b) Traditional statistical approaches & technical analysis– Financial ratios, statistically significant deviations etc.
c) Algorithmic approaches – quantitative finance, algorithmic trading techniques applied
d) Machine learning technical techniques – The traditional approach has been in using Machine Learning(ML) models and lately several Deep Learning(DL) models have appeared that have significantly outperformed the ML approaches. e.g. [4] discusses Neural networks application to forecasting and trading. There are different kinds of DL models: Deep Multi-layer Perceptron(DMLP), RNN, LSTM, CNN, Restricted Boltzmann Machines(RBMs), DBN, Autoencoder(AR) and Deep Reinforcement learning [5].

Traditional FX trading

Historically Technical Analysis(TA) has been used for FX time series prediction. Unlike fundamental analysis that values a security based on business results such as sales, earnings etc, TA looks as demand and supply and its impact on price and volume. This can be used on any historical financial data. A survey undertaken among forex dealers in Hong Kong in 1995 [6] revealed that time horizons play a part in choosing the method of analysis. At shorter horizons, there is higher reliance on TA compared to fundamental analysis and this gets revered in longer timeframes

There are three principles about the behaviour of technical analysts [7]. Firstly, the asset price history incorporates all the relevant information which makes the use of asset fundamentals valueless. The second is that asset prices move in trends, implying predictability and also profitability. Lastly asset price patterns repeat themselves.

The FX market differs from equity markets in some aspects [8]. The total turnover in global FX market is much greater than the turnover of the largest stock exchanges. Due to the existence of professional managers, the impact of individual investors may be neglected without loss of generality [9]. The FX market has a much higher share of short term inter dealer trading [10]. Also there is less confidence among traders in models of fair value in FX markets compared to equity market [11].

Technical Analysis usage goes back to at least 1700 [12] and although modern technical analysis was developed in the context of the stock market, foreign currency traders have been employing these strategies for trading over the years. TA can be divided into broad methods-charting and mechanical. Charting is the older method which is subjective and requires analyst to be able to find and interpret patterns. Mechanical rules require the trader to apply rules based on mathematical functions of past and current data. Some of the popular mathematical rules used are filter, moving average, linearly weighted moving average, exponentially weighted moving average, moving average convergence-divergence, moving average oscillator, stochastic oscillator, channel breakout, relative strength indicator and Bollinger bands. While many more rules can be created, considering too many rules can reduce test power [13]. At the same time, choosing too few rules [14] can cause biases in statistical inference due to data mining. A universe of 7650 trading rules [15] applied to six currency pairs over a 20 yr. period showed that while a large number of outperforming rules are profitable over short periods based on the Sharpe ratio, they are not consistently profitable and so overall results are more consistent with the adaptive markets hypothesis.

The methods of TA attempt to identify trends and reversals of trends [16]. To identify trends, analysis must find peaks and troughs in the price series. A peak is the highest value of the exchange rate within a specified period of time (a local maximum) while a trough is the lowest value the price has taken during the same period(a local minimum). As can be seen in figure 1, a trendline can be drawn visually or automatically by connecting two local troughs in the data. As more troughs touch the trendline without violating it, the technician places more confidence in the validity. The angle of the trendline indicates the speed of the trend with steeper lines indicating faster appreciation.

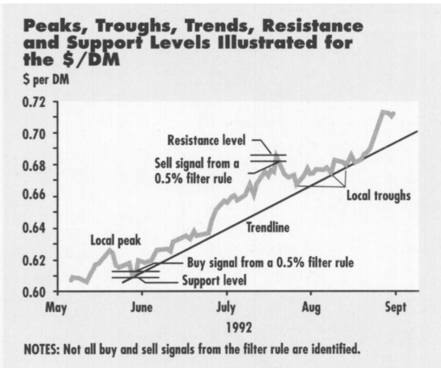


Figure 1 Peaks, Troughs, Trends, Resistance and support levels illustrated for the $/DM

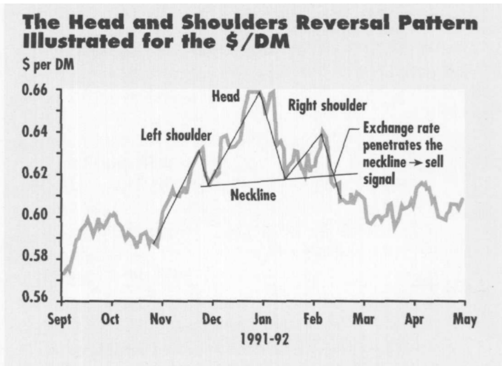


Figure 2 The Head and Shoulders pattern illustrated for $/DM

After the trendline has been established, the technician trades with the trend, buying the foreign currency if the uptrend is signalled and selling if it is a downtrend.

Spotting the reversal is equally important as detecting the trend and peaks and troughs are important too. Local peaks are called resistance levels and local troughs are called support levels. Several patterns are used to identify a shift in trend from one direction to another and an example of one of the best known reversal patterns is the “head and shoulders” pattern as shown in figure 2. This pattern is a reversal following an uptrend characterized by three local peaks with the middle peak being the largest of the three. The line between the trough and the shoulders is known as the neckline and when the exchange rate penetrates the head and the shoulders, the technician confirms a reversal of the trend. Other similar reversal patterns are the V(single peak), double top (two similar peaks) and triple top (three similar peaks). The reversal patterns of a downtrend are the mirrors of reversal patterns of uptrend.

Another variety of mechanical trading rules is the “moving average” class. This is the average closing price of the exchange rate over a given number of previous trading periods. The length of the average “window” reflects long term or short term trends. Figure 3 shows an example of the 5 and 20 day moving averages with the exchange rate itself. A typical moving average rule prescribes a buy(sell) signal when the short moving average crosses the longer moving average from below(above), i.e. when the exchange rate is rising(falling) fast.

A final type of trading rule is the class of oscillators which are useful in non-trending markets when the exchange rate is not trending up or down strongly. A simple type of oscillator index, as shown in figure 4 is given by the difference of two moving averages: the 5-day moving average minus the 20-day moving average. Oscillator rules suggest buying(selling) the foreign currency when the oscillator index takes an extremely low(high) value. The oscillator index, as a difference between moving averages will also generate buy/sell signals from moving average rule, when the index crosses zero.

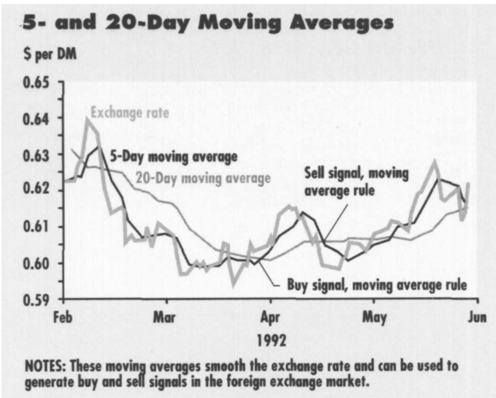


Figure 3 5 and 20 day moving average

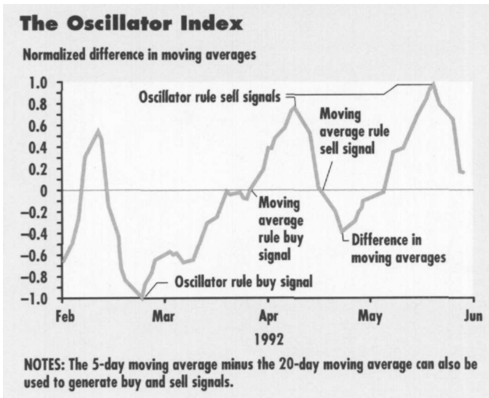


Figure 4 The Oscillator Index

A basic problem in evaluating technical trading strategies is that rules requiring judgement and skill are impossible to quantify and therefore unsuitable for testing. An objective rule does not depend on individual skill or judgment and should be commonly used to reduce the problem of drawing false conclusions from data mining. We can have filter rules that give signals when exchange rates rise x percent over recent minimums or we can have moving average rules that gives signals when short moving averages are higher than long moving averages.

AI at FX forecasting

Quantitative exploration

Over the last two decades, with the increase of time-series data availability [17], there have been hundreds of techniques used for time series prediction. Deep Learning is a type of ANN that consists of multiple processing layers and enables high level abstraction to model data. The advantage is that the good features are extracted automatically without human intervention(as is required in traditional technical analysis above). DL models have been used in multiple areas such as image, speech, video, audio, natural language processing, sentiment analysis, language translation etc. [18]

Prior to DL approaches, there have been ML approaches for prediction based on soft computing approaches [19], usage of data mining techniques [20], and ML models such as ANNs, evolutionary computations, Genetic programming and Agent based models [21].

Deep Multi-Layer Perceptron(DMLP)

(DMLP) was one of the first developed ANNs. Its difference from shallow nets is that it contains more layers, at least three- input, hidden and output. The number of neurons and the number of layers are the hyper parameters. In general, each neuron has input (x), weight(w) and bias (b) terms. The first equation [22] shows the output of a single neuron in the Neural Network(NN). Also each neuron has a non-linear activation function which produces a cumulative output of all the preceding neurons. The most common activation functions are sigmoid [23], hyperbolic tangent [24], Rectified Linear Unit (ReLU) [25], leaky-ReLU [26], swish [27] and softmax [28].

$$y_i = \sigma\left(\sum_i W_i x_i + b_i\right)$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$R(z) = \max(0, z)$$

$$R(z) = 1(x < 0)(\alpha x) + 1(x >= 0)(x)$$

$$f(x) = x\sigma(\beta x)$$

$$\mathrm{softmax}(z_i) = \frac{\exp z_i}{\sum_j \exp z_j}$$

Using DMLP, problems of classification and regression can be solved by modelling the input data, however there are problems of storage capacity if input parameters are increased, because of the fully connected nature of the network. To overcome this issue, different types of DNN methods such as CNNs have been proposed [5].

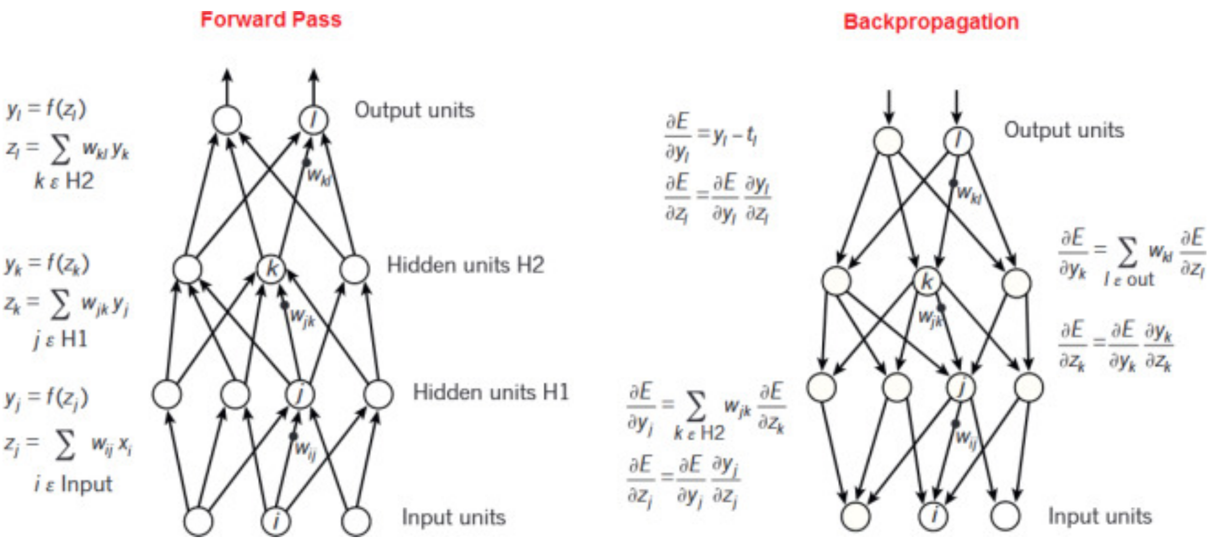


Figure 5 Deep Multi-Layer Neural Network forward pass and backpropagation

The DMLP learning stage is implemented through back propagation, wherein the error in neurons in output layer is propagated to the preceding layers. Optimisation algorithms such as SGD, AdaGRAD< ADAM and RMSProp [29] [30] are used to find the optimum parameters.

Recurrent Neural Network (RNN)

RNNs are another type of DL network used for time series data. Unlike FNNs, RNNs use internal memory to process incoming inputs. Each RNN unit takes the current and previous input data at the same time. They process the input sequences one by one at any given time during the operation. The units in the hidden layer hold information about the history of input in state vector. When the output of the units in the hidden layer is divided into discrete time steps, an RNN is converted into a DMLP.

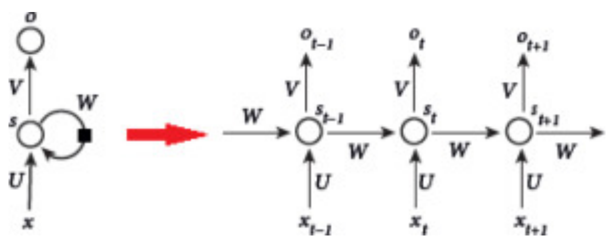


Figure 6 RNN cell through time

The difficulty of training an RNN is the backward dependence over time. As learning period increases, the RNNs become very complex. To solve this problem, LSTMs with different structures of ANNs have been developed [5]. The following equations (8) and (9) show the simpler RNN formulations and the total error which is the sum of errors from each time iteration.

$$h_t = Wf(h_{t-1}) + W^{(hx)} x_{[t]}$$

$$y_t = W^{(S)} f(h_t)$$

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W}$$

Long Short Term Memory(LSTM)

LSTM is a type of RNN where the network remembers both short term and long term values. They consist of LSTM units which combine to form an LSTM layer. An LSTM unit is composed of cells, each with an input gate, output gate and a forget gate, which regulate the information flow. With these features, each cell remembers the values over arbitrary time intervals. The following equations show the forward pass of an LSTM unit [31].

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$

$$c_t = f_t * c_{t-1} + i_t * \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$

$$h_t = o_t * \sigma_h(c_t)$$

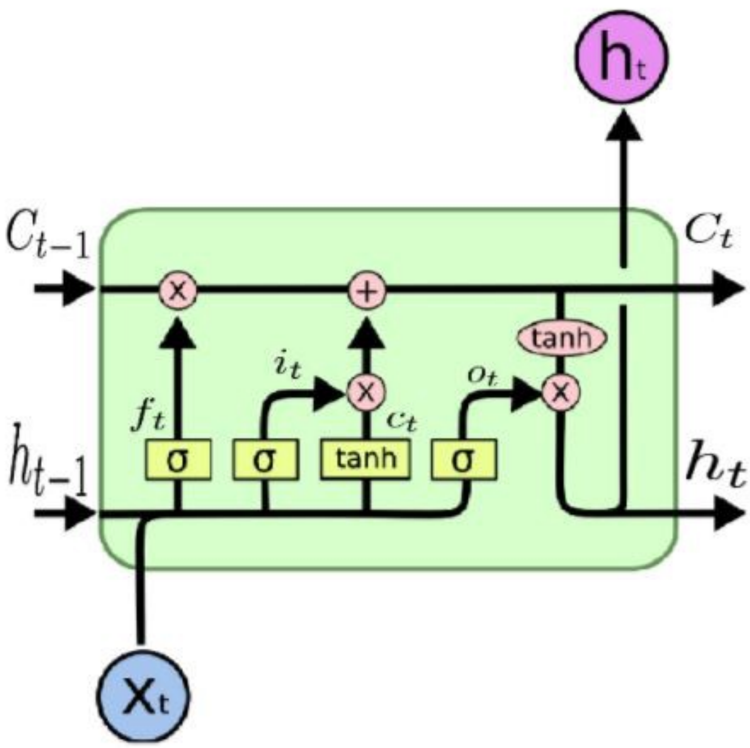


Figure 7 Long Short Term Memory Unit

Here x_t is the input vector to the LSTM unit, f_t is the forget gate's activation vector, i_t is the input gate's activation vector, c_t is the cell state vector, σ_g is the sigmoid function, σ_c, σ_h are the hyperbolic tangent function, * is the elementwise(Hadamard) product, W, U are the weight matrices to be learnt, b are the bias weight vectors to be learnt.

Convolution Neural Network(CNN)

The CNN is a type of DNN that consists of convolution layers based on convolution operation. This is mostly used in image processing problems. The advantage of CNN is the number of parameters compared to vanilla DL models such as DMLP. Filtering with the kernel window results in fewer parameters. The following equations shows the process of convolution followed by the NN architecture with a Softmax for classification at the end of the network.

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)\, w(t-a)$$

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)\, K(i-m, j-n).$$

$$z_i = \sum_j W_{i,j}\, x_j + b_i.$$

$$y = \operatorname{softmax}(z)$$

$$\operatorname{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

Restricted Boltzmann Machine(RBM)

RBM is a productive stochastic ANN that learns a probability distribution on the input set [32]. These are mostly used for unsupervised learning and used for dimension reduction, classification, feature learning and collaborative filtering [33]. The benefit is ability to find hidden patterns in unsupervised fashion but the training process is difficult.

The RBM is a am undirected two-layer graphical model with a visible and hidden layer.

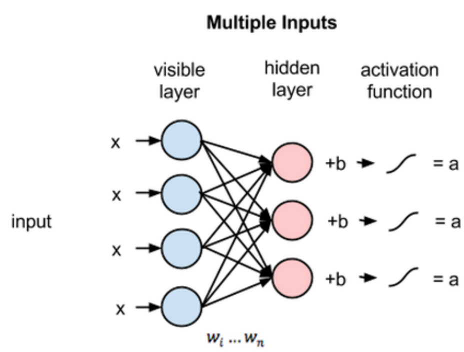


Figure 8 RBM layers

Each cell is a computation unit that makes stochastic decisions of whether the nerve node will transmit the input, once it processes the input. The inputs multiplied by the weights and added to the biases are passed through the activation function and in reconstruction stage, the results from the outputs re-enter the network as input before exiting the visible layer as output. The goal is to reduce the difference between the previous input and the values after the process.

The following equations show the probability semantics for an RBM using its energy function where P denotes the probability semantics for an RBM, Z denotes the partition function, E denotes the energy function, h denotes the hidden units and v denotes the visible units. a is the bias weights for the visible units, b is the bias weights for the hidden units, W is the matrix weight of the connection between the visible and hidden units, T is the transpose of the matrix.

$$P(v,h) = \frac{1}{Z} \exp(-E(v,h))$$

$$Z = \sum_v \sum_h \exp(-E(v,h))$$

$$E(v,h) = -a^T v - b^T h - v^T W h$$

The learning is performed multiple times [32] and training is implemented by minimising the negative log-likelihood of the model and the data. The Contrastive Divergence(CD) algorithm is used as the stochastic approximation algorithm, which replaces the model expectation using an estimation using Gibbs sampling with a limited number of iterations. The hyper parameters(momentum, learning rate, weight cost, batch size, number of epochs, number of layers, size of units, loss function) are searched with MS, GS, RS and Bayesian methods.

Deep Belief Networks DBN)

A DBN is a type of ANN consisting of a stack of RBM networks. It is a probabilistic generative model that consists of latent variables. In a DBN there is no link between the units in each layer. When the DBN is trained in an unsupervised manner, it can learn to reconstruct the input set in a probabilistic manner. Then the layers in the network begin to detect discriminating features in the input. The process has two steps: stacked RBM learning

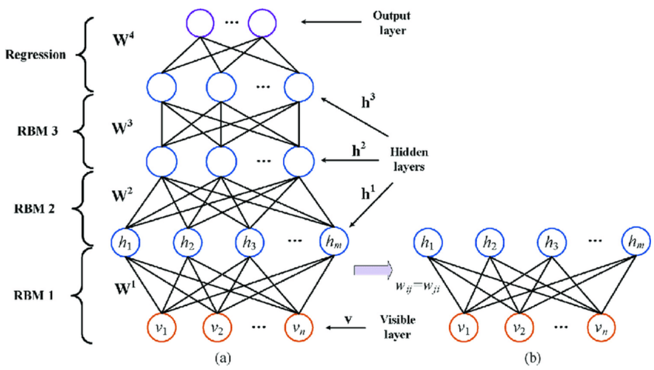


Figure 9 Deep Belief Network

and backpropagation learning. In stacked RBM learning, an iterative CD algorithm is used [33] and in backpropagation learning, optimization algorithms are used to train the network [34]. The hyperparameters are similar to those of RBM and are searched similar to other deep networks.

Autoencoders(AEs)

AEs are ANNs used as unsupervised learning models. They remap the inputs such that the inputs are more representative for classification. The most notable features are dimensionality reduction and feature learning. One drawback is that focussing on minimizing the loss of data relationship in the encoding of AEs causes the loss of significant data relationships.

In general, AEs have two components: encoder and decoder. The input $x \in [0,1]^d$ is converted through a function $f(x)$ (W_1 denotes a weight matrix, b_1 denotes a bias vector, σ_1 element-wise sigmoid activation of the encoder). Output h is the encoded part of the AEs. The inverse of function $f(x)$ called function $g(h)$ produces the reconstruction of output r (W_2 denotes a weight matrix, b_2 denotes bias vector and σ_2 is an element-wise sigmoid activation function of the decoder. The following equations show the simple AE process and the calculation of the MSE.

$$h = f(x) = \sigma_1 (W_1 x + b_1)$$

$$r = g(h) = \sigma_2 (W_2 h + b_2)$$

$$L(x, r) = \|x - r\|^2$$

Qualitative Exploration

Market participants react to textual, video news from various quarters – social, business, regulatory, economic [macro fundamentals], political, and geographic/environmental. [3] describes and evaluates qualitative information standard fundamental and non-fundamental news.

Qualitative news assimilation takes the form of identifying key concepts. [35] takes the PCA [principal component analysis] on tokenized news pieces to find causation of price movements.

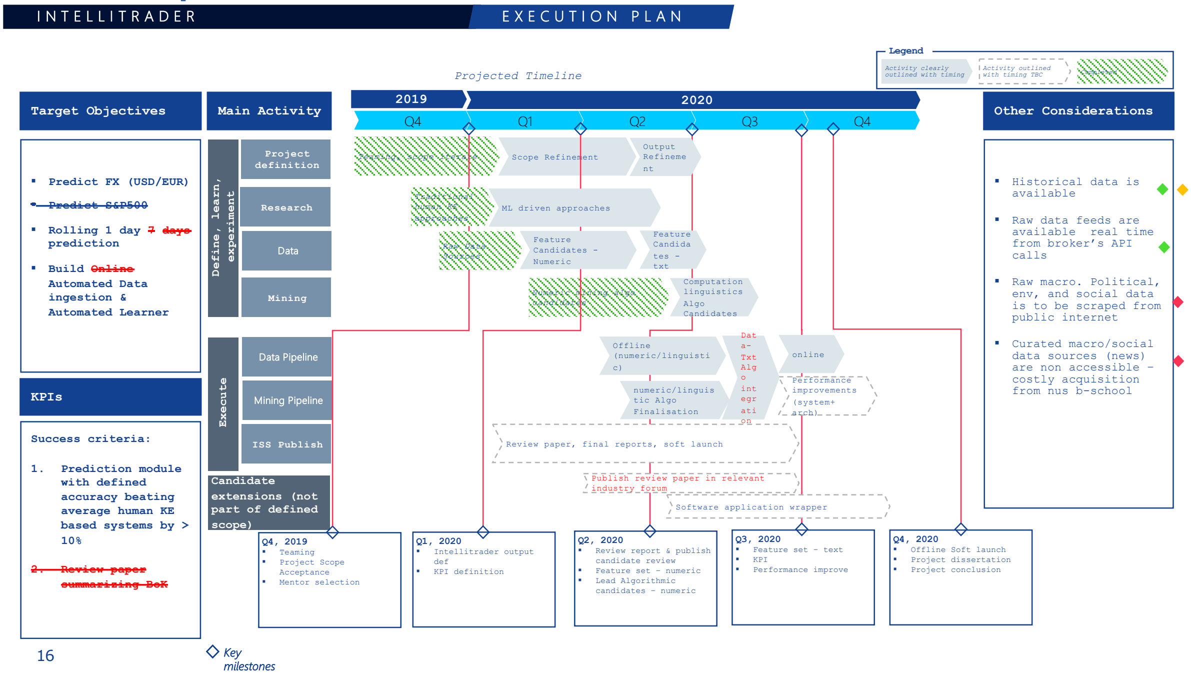
Latest qualitative information based approaches explore connecting sentiment analysis in a news segment/preceding news segments and its impact on the future asset price movement. [36]explores trusting the social media for financial analysis.

Different news carry different weightage and impact to the asset price and trading decisions. [37]explores assigning weights to different news.

Approach

Our approach is summarized in the project plan below:

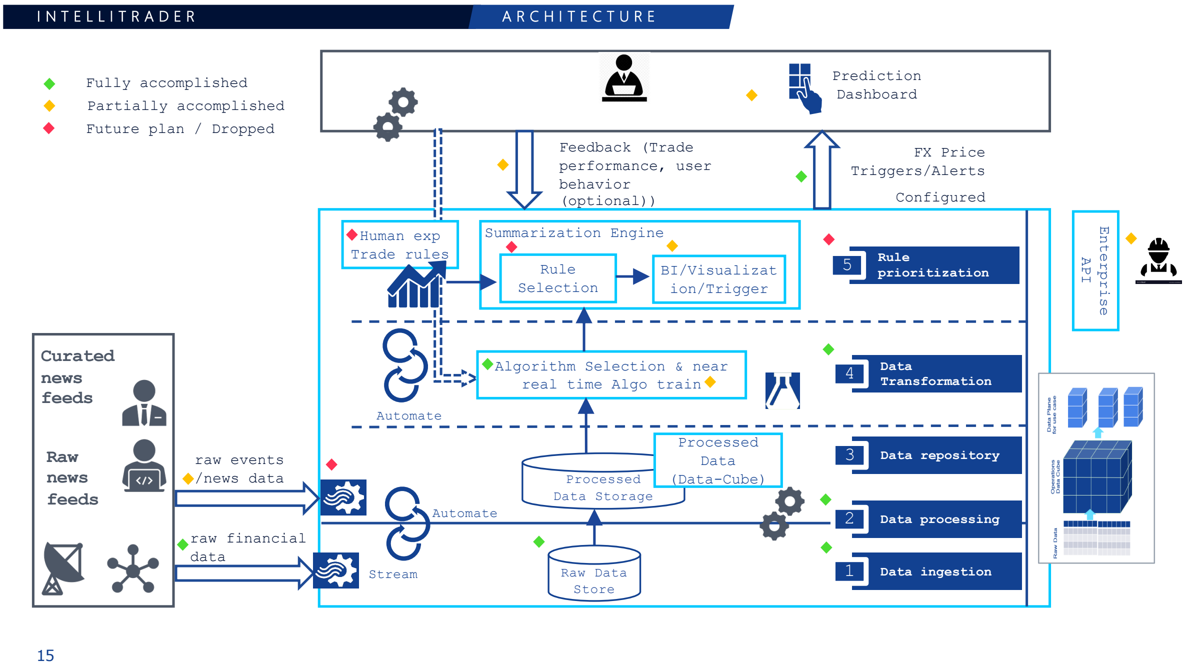
Execution plan



Some parts of the proposed plan had to be curtailed for reasons of financial viability.

System architecture:

System architecture approach is summarized as follows:



Our focus is on creating the core concept: that is a high accuracy prediction algorithm. Other periphery components, i.e. Trader GUI and web API interface will be established in commercial live system.

Datasets

Quantitative datasets

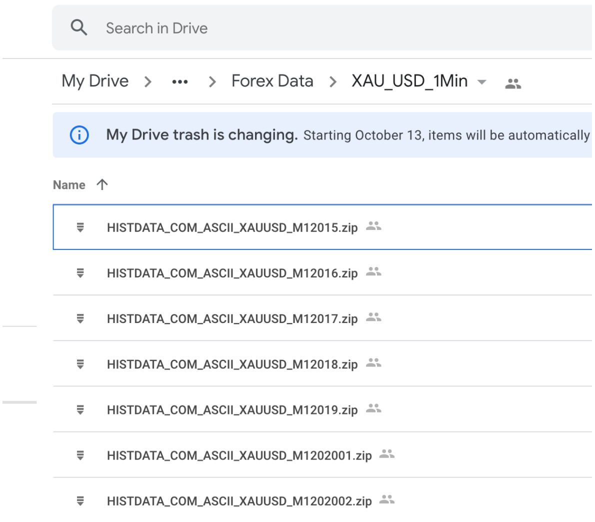
Data type: Ticker data, i.e. the data of the executed trades in a given day. Raw Quantitative data consists of daily exchange rate information:

1. Timestamp: Date and time of trade execution
2. Bid: Buyer bid
3. Ask: Seller ask

Duration: Last decade (2010-2019) of trades executed

Source: www.histdata.com, a free historical ticker data website.

Format: .csv format and available for various currency pairs e.g. EUR/GBP/JPY/USD in various currency pairs. We chose the USD/EUR FX rate as it's the highest trading volume pair.



Qualitative datasets

Data type: Digital Written English information about government policies, news, and investor, trader, public opinion.

We classified Data sources into two classes:

a) Authentic data sources: e.g. government websites, market regulator websites and well known /acclaimed news houses (Wall street journal)
b) Non-Authentic data sources: e.g. web forums, investor group chats etc

Duration: Matching Quantitative data duration, 2010-2019 to support time window aligned training. [e.g. 5 days preceding window (t0-4) to t0, to predict next day (t0+1)]

Source: Multiple. Following were the finalized data sources:

a) Raw: Keyword based, twitter handle based Twitter financial feed API. https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference
b) Partial processes: Selected economic, market news: Dow Jones news feed https://developer.dowjones.com/site/docs/newswires_apis/dow_jones_top_stories_api/index.gsp
c) Partial processed: Webhose: https://webhose.io/products/news-feeds/
d) Partial processed: Ravenpack: https://www.ravenpack.com/finance
e) Financial databases: WRDS, Datastream/Eikon

Format: Raw Text corpus (.html), processed text corpus (.txt/.csv) with defined metadata dictionary

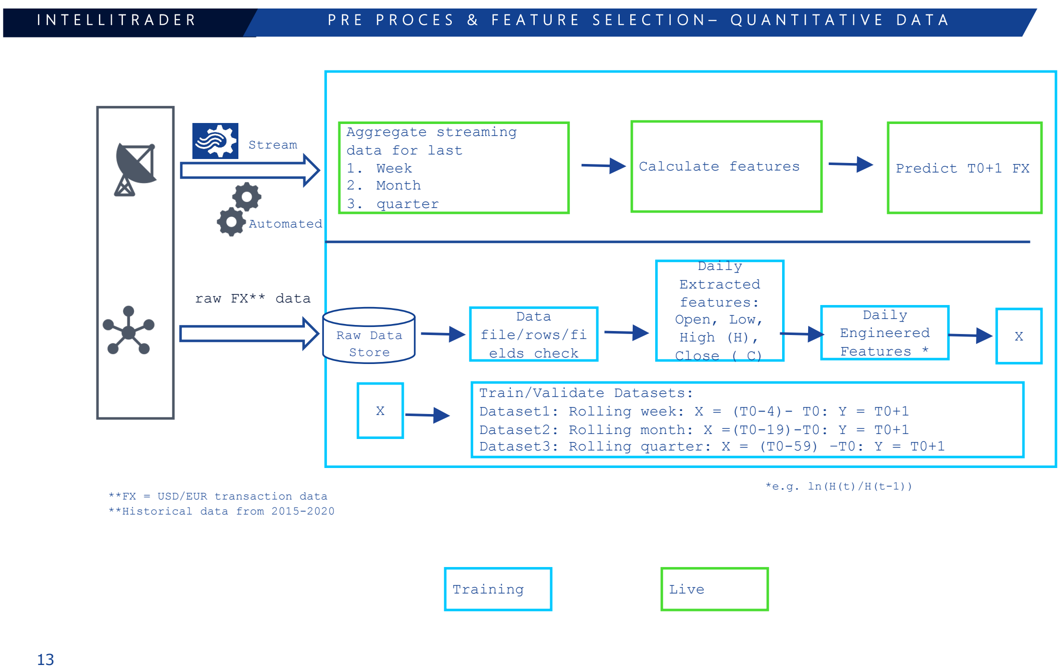
Pre-processing

Quantitative data

Salient points: Trading tick data for EUR/USD from 2010-2019

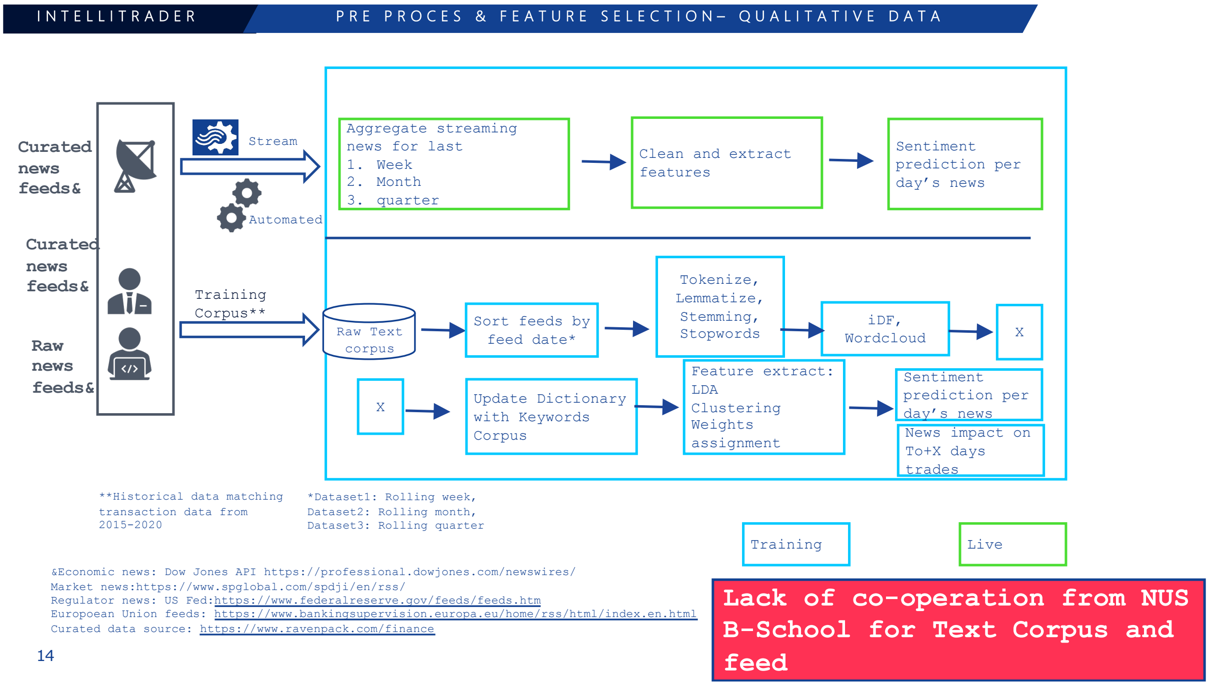
Database size: 4.5GB

Number of rows:149 million rows



Qualitative data

Qualitative Pre-processing pipeline:



Features of interest

Quantitative information

Base features:

1. B_F1: Day's Open price
2. B_F2: Day's Lowest price
3. B_F3: Day's Highest price
4. B_F4: Day's Close price

Engineered features:

1. EF_1 (daily average): = (B_F3(Tx)+B_F3(Tx))/2: x = xth day
2. EF_2 (log change of average): ln(EF_1(Tx)/EF_1(Tx-1)) x = xth day, ln = natural log
3. EF_3 (percentage change of average) : (EF_1(Tx)/EF_1(Tx-1)-1) x= xth day
4. EF_4 (log change of open to previous close) : ln((B_F1(Tx)/B_F4(Tx-1)) x= xth day

Qualitative information *aka* Trading opinions

Market participants perception influences trade dynamics. Perception is built on various information sources being ingested by participants over a period of time. More recent news is more relevant to trade decisions then farther /earlier information. According to …, news of last 1 week contributes to maximum

Objective is to extract fundamental information from relevant news and use the same for analysis and prediction of FX.

Information sources include policies, news, and investor/trader/public opinion as expressed in written digital natural language sources.

For project purpose, we classified Data sources into two classes:

c) Authentic data sources: e.g. government websites, market regulator websites and well known /acclaimed news houses (Wall street journal)
d) Non-Authentic data sources: e.g. web forums, investor group chats etc

Economics New:

Global economics news. Data source: Dow Jones API: https://professional.dowjones.com/newswires/

Regulator News

Central bank and government finance news RSS feeds.

1. US Fed: https://www.federalreserve.gov/feeds/feeds.htm
2. European Union feeds (European central bank): https://www.bankingsupervision.europa.eu/home/rss/html/index.en.html

Market & Trade News

Sources covering market participant sentiment news, Market change news [38]

Channels of news acquisition:

1. News articles: Yahoo finance.
2. Financial market aggregator news: https://www.marketwatch.com/rss/
3. Web forums: Yahoo message boards, Reuters
4. Market sentiment curated data: Thomson Reuters MarketPsych Indices [2,000 premium news and 800 financial social media sources]
5. Analysis: Wall street journal/ S&P dow jones: https://www.spglobal.com/spdji/en/rss/

Pre-processing & Analysis:

1. Filter noisy posts
2. Stop words removal
3. Bag of words, Named entity recognition
4. word segmentation
5. Topic identification
6. Sentiment scores: +ive/-ve. Relationship between participants in same polarity
7. Factor analysis: dimensionality reduction and latent correlations and relationships

8. Segmentation techniques: SVM

Challenges

Options explored

We explored multiple RSS streams as well as public API based extractions. While we had limited free datasets afforded from relevant feeds, we were able to generate needed pre-processing as defined in the qualitative pre-processing steps.

We also

Pricing information

We encountered following challenges on text data acquisition:

1. Prohibitive Pricing of the financial feeds: for us to have any reasonable training corpus, we assessed that we need approximately 5,000-15,000 USD or Singapore dollars equivalents
2. We explored with NUS databases, and took help from our mentor, Jen Hong. But unfortunately NUS business school could share database keys with only 3,000 Dollars.
3. Lastly, the corpus was not matching the historical timeline 2010-2019

Hence, while much of our research suggested that combining quantitative and qualitative information sets will lead to a better prediction accuracy [39], in access of 10% prediction accuracy by including text, we decided to not proceed with qualitative analysis being included in the final model trainings and outcomes.

Future roadmap

We will like NUS ISS help for free access to NUS business school database so that we can build the Text Analytics based pipeline into the solution.

Feature extraction Algorithm

MLP

Key architectural constructs:

- Input (T0-x days): 5, 10, 20, 60
- Total Number of nodes: 900 nodes
- Activation function: ReLU
- Optimizer: Adam
- Total Number of layers: 4
- Loss function: Mean squared error for the predicted vs actual feature.

LSTM (Basic)

Key architectural constructs:

- Input (T0-x days): 5, 10, 20, 60
- LSTM units : 50/layer
- Layers: 2
- Optimizer: Adam
- Loss function: Mean squared error for the predicted vs actual feature.

Vanilla Encoder Decoder

Key architectural constructs:

- Input (T0-x days): 5, 10, 20, 60
- Encoder Layers: 2
- Dimension: 16
- Hidden layer: 128
- Activation function: Sigmoid
- Optimizer: Gradient Descent
- Decoder Layers: 2
- Loss function: Mean squared error for the predicted vs actual feature

LSTM with attention

- Input (T0-x days): 5, 10, 20, 60
- Layer size : 128
- Number of layers: 1
- Optimizer: Adam
- Loss function: Mean squared error for the predicted vs actual feature

GRU

- Input (T0-x days): 5, 10, 20, 60
- Layer size : 128
- Number of layers: 1
- Optimizer: Adam
- Loss function: Mean squared error for the predicted vs actual feature

Multi-headed attention

- Input (T0-x days): 5, 10, 20, 60
- Number of heads: 8
- Number of layers: 1
- Attention windows: 6
- Optimizer: Adam
- Loss function: Mean squared error for the predicted vs actual feature

SeriesNet

- Input (T0-x days): 5, 10, 20, 60

- Number of CNN blocks: 7
- Number of filters(conv1D): 32
- Filter length = 2
- Dilation=1
- Optimizer: Adam
- Loss function: Mean squared error for the predicted vs actual feature

Loss Functions considered

We finalized on a Mean squared error function.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where Y is the feature under optimization using a give Deep Neural Network architecture.

Validation criteria

The datasets were reasonably balanced across all observation classes. Observation classes were:

1) Last 5 trading days (rolling week) based datasets
2) Last 10 trading days (rolling fortnight) based datasets
3) Last 20 trading days (rolling month) based datasets
4) Last 60 trading days (rolling quarter) based datasets

We finalized on a randomized 80|20 distribution, with 80% used for training various Deep Neural network architectures.

Execution

Features sets:

Base features:

1. B_F1: Day's Open price
2. B_F2: Day's Lowest price
3. B_F3: Day's Highest price
4. B_F4: Day's Close price

Engineered features:

5. EF_1 (daily average): = (B_F3(Tx)+B_F3(Tx))/2: x = xth day
6. EF_2 (log change of average): ln(EF_1(Tx)/EF_1(Tx-1)) x = xth day, ln = natural log
7. EF_3 (percentage change of average) : (EF_1(Tx)/EF_1(Tx-1)-1) x= xth day
8. EF_4 (log change of open to previous close) : ln((B_F1(Tx)/B_F4(Tx-1)) x= xth day

Data sets:

1) Last 5 trading days (rolling week) based datasets
2) Last 10 trading days (rolling fortnight) based datasets
3) Last 20 trading days (rolling month) based datasets
4) Last 60 trading days (rolling quarter) based datasets

Experiments were run for all finalized Deep neural network architectures to minimize the loss function for each feature against each data sets.

In total, **224 experiments**, comprising of 32 experiments per model, and total of 7 models.

Experiment setups, datasets and observed results are explained in the table in next session

Model Comparison

Loss function results are catalogued below across 224 experiments:

Best results on a given feature are catalogued with grey shade, and best outcomes in a given architecture class are highlighted in bold.

Overall best outcomes are in bold and red.

Model Architecture	Historical binding	Prediction Structure	O	L	H	C	Avg = (H+L)/2	Log (Avg t/Avg (t-1))	% (Avgt-Avg t-1)/Avg t-1	log (Open T/Close T-1)
MLP	weekly history	Rolling 5 days, predict 6th day	5.21E-05	8.17E-05	1.87E-05	2.83E-05	6.57E-05	7.52E-06	7.07E-06	**1.70E-07**
	fortnight history	Rolling 10 days, predict 11th day	2.40E-05	2.53E-05	4.91E-05	4.53E-05	1.66E-05	8.33E-06	7.21E-06	1.18E-07
	monthly history	Rolling 20 days, predict 21 st day	7.39E-05	2.62E-05	1.90E-05	3.05E-05	4.67E-05	7.89E-06	8.99E-06	2.83E-07
	quarterly history	Rolling 60 days, predict 61st day	9.00E-05	8.32E-05	9.30E-05	2.44E-05	3.78E-05	8.71E-06	8.54E-06	5.83E-07
Model Architecture	**Historical binding**	**Prediction Structure**	**O**	**L**	**H**	**C**	**Avg = (H+L)/2**	**Log (Avg t/Avg (t-1))**	**% (Avgt-Avg t-1)/Avg t-1**	**log (Open T/Close T-1)**
Basic LSTM	weekly history	Rolling 5 days, predict 6th day	4.74E-05	7.35E-05	0.000190192	0.000545594	8.70E-05	6.31E-06	6.44E-06	1.48E-07
	fortnight history	Rolling 10 days, predict 11th day	0.000434509	6.05E-05	0.000134756	0.000258425	5.42E-05	6.27E-06	6.77E-06	**1.18E-07**
	monthly history	Rolling 20 days, predict 21 st day	4.73E-05	0.00052808	0.00033091	0.000138907	0.000240363	6.33E-06	6.27E-06	1.44E-07
	quarterly history	Rolling 60 days, predict 61st day	7.63E-05	0.00015406	4.42E-05	4.75E-05	5.03E-05	5.87E-06	6.26E-06	3.13E-07
Model Architecture	**Historical binding**	**Prediction Structure**	**O**	**L**	**H**	**C**	**Avg = (H+L)/2**	**Log (Avg t/Avg (t-1))**	**% (Avgt-Avg t-1)/Avg t-1**	**log (Open T/Close T-1)**
Vanilla Encoder	weekly history	Rolling 5 days, predict 6th day	0.026326132	0.025684406	0.026985978	0.026339711	0.02926946	1.69E-05	1.69E-05	**9.94E-07**
	fortnight history	Rolling 10 days, predict 11th day	0.030169326	0.029438923	0.030987456	0.03014728	0.030235418	1.72E-05	1.72E-05	1.05E-06
	monthly history	Rolling 20 days, predict 21 st day	0.016089937	0.015826169	0.016322725	0.016070185	0.018642766	1.69E-05	1.69E-05	9.50E-07
	quarterly history	Rolling 60 days, predict 61st day	0.013033613	0.012858878	0.013171817	0.013013151	0.014389659	1.68E-05	1.68E-05	9.23E-07
Model Architecture	**Historical binding**	**Prediction Structure**	**O**	**L**	**H**	**C**	**Avg = (H+L)/2**	**Log (Avg t/Avg (t-1))**	**% (Avgt-Avg t-1)/Avg t-1**	**log (Open T/Close T-1)**
Attention Only	weekly history	Rolling 5 days, predict 6th day	4.05E-05	4.42E-05	3.85E-05	3.84E-05	7.10E-05	4.54E-06	4.54E-06	**2.62E-08**
	fortnight history	Rolling 10 days, predict 11th day	4.37E-05	4.28E-05	4.89E-05	4.03E-05	9.29E-05	4.60E-06	4.67E-06	6.72E-08
	monthly history	Rolling 20 days, predict 21 st day	4.59E-05	5.35E-05	5.12E-05	5.04E-05	6.78E-05	4.11E-06	4.22E-06	4.79E-08
	quarterly history	Rolling 60 days, predict 61st day	2.29E-05	2.59E-05	1.82E-05	1.76E-05	0.000165832	4.47E-06	4.25E-06	5.26E-07
Model Architecture	**Historical binding**	**Prediction Structure**	**O**	**L**	**H**	**C**	**Avg = (H+L)/2**	**Log (Avg t/Avg (t-1))**	**% (Avgt-Avg t-1)/Avg t-1**	**log (Open T/Close T-1)**
GRU	weekly history	Rolling 5 days, predict 6th day	0.000691137	0.000636682	0.000679951	0.000672595	0.002484081	3.80E-05	3.78E-05	1.12E-05
	fortnight history	Rolling 10 days, predict 11th day	0.000740155	0.000666208	0.000663834	0.000578271	0.00107039	7.17E-06	7.17E-06	7.75E-08
	monthly history	Rolling 20 days, predict 21 st day	0.039409402	0.040959624	0.041048716	0.042278141	0.000296937	3.56E-06	3.56E-06	**7.23E-08**
	quarterly history	Rolling 60 days, predict 61st day	0.010739709	0.009002187	0.010408199	0.01371527	0.049763267	4.08E-05	2.71E-05	6.89E-06
Model Architecture	**Historical binding**	**Prediction Structure**	**O**	**L**	**H**	**C**	**Avg = (H+L)/2**	**Log (Avg t/Avg (t-1))**	**% (Avgt-Avg t-1)/Avg t-1**	**log (Open T/Close T-1)**
Multihead attention	weekly history	Rolling 5 days, predict 6th day	0.000199587	0.000214429	0.0001956	0.000191448	0.000145672	1.13E-05	1.12E-05	**4.34E-06**
	fortnight history	Rolling 10 days, predict 11th day	0.000147357	0.000153291	0.000154146	0.000138056	0.000174132	1.19E-05	1.12E-05	4.40E-06
	monthly history	Rolling 20 days, predict 21 st day	0.001678234	0.001559112	0.001794763	0.001660155	0.00023638	1.12E-05	1.10E-05	4.43E-06
	quarterly history	Rolling 60 days, predict 61st day	0.001071243	0.001431132	0.001293113	0.001399098	0.003921759	1.12E-05	1.15E-05	5.19E-06
Model Architecture	**Historical binding**	**Prediction Structure**	**O**	**L**	**H**	**C**	**Avg = (H+L)/2**	**Log (Avg t/Avg (t-1))**	**% (Avgt-Avg t-1)/Avg t-1**	**log (Open T/Close T-1)**
LSTM-Attention	weekly history	Rolling 5 days, predict 6th day	1011.743013	12.19428764	12.17228818	0.017428734	6.77E-05	3.25E-06	3.25E-06	5.08E-08
	fortnight history	Rolling 10 days, predict 11th day	8.56E-05	4.76E-06	4.75E-06	3.39E-08	0.179209878	0.000484145	0.000466226	1.15E-05
	monthly history	Rolling 20 days, predict 21 st day	0.050973917	2.13E-05	2.13E-05	2.42E-07	7.63E-05	3.57E-06	3.58E-06	**3.07E-08**
	quarterly history	Rolling 60 days, predict 61st day	0.000237528	3.49E-06	3.52E-06	3.31E-08	0.001147298	3.47E-06	3.46E-06	4.82E-08
Model Architecture	**Historical binding**	**Prediction Structure**	**O**	**L**	**H**	**C**	**Avg = (H+L)/2**	**Log (Avg t/Avg (t-1))**	**% (Avgt-Avg t-1)/Avg t-1**	**log (Open T/Close T-1)**
SeriesNet	weekly history	Rolling 5 days, predict 6th day	0.000886937	0.000777911	0.0007986	0.000750802	0.000791109	7.86E-06	7.89E-06	**1.54E-08**
	fortnight history	Rolling 10 days, predict 11th day	0.00178351	0.001642802	0.001818069	0.00168125	0.001737036	4.96E-06	4.97E-06	1.58E-08
	monthly history	Rolling 20 days, predict 21 st day	0.002750359	0.002615764	0.002815297	0.002675225	0.002711615	4.23E-06	4.24E-06	3.78E-08
	quarterly history	Rolling 60 days, predict 81st day	0.00852576	0.00815849	0.008873621	0.00847593	0.008458718	3.33E-06	4.48E-06	2.35E-08
	consolidated output series of 50 future periods		0.007541682	0.008191677	0.008630198	0.00834457	0.008220323	3.42E-06	3.43E-06	**2.60E-08**

Outcomes commentary

We observe:

a) In general ln(Open T/Close T-1) engineered performs consistently better
b) Attention Only and LSTM-Attention architectures perform better than other architectures.
c) While we achieve best outcomes for last rolling 5 days based data aggregation, it's closely followed by last rolling month based outcomes, we observe that in general last rolling 5 days have better price prediction ability and stability across models.

Finally, we observe that instability/inconsistency in results outcome when performed for different time intervals, leading to need for model robustness for any commercial usage.

Conclusion

We took a formidable project to predict asset pricing given the possible financial benefits unlocking. The field, while there are some public information on explorations, is quite restrictive in quality knowledge sharing. This again is attributed to corporate and private interests to “solve” the problem.

At the same time, we didn’t anticipate that we have to stop the qualitative branch of our project plan.

Having said that we are happy with the variety of explorations we tried, running 100’s of experiments and observing reasonable good accuracies.

The outcomes give us confidence to take the project ahead, enrich the information inclusion in the project as well as data transformations, and novel DNN/ML/AI architecture explorations.

Acknowledgements

We will like to thank Prof Jen Hong for his guidance in project setup discussions, and scope definition.

Bibliography

[1] s. Peterson, The Quants: How a New Breed of Math Whizzes Conquered Wall Street and Nearly Destroyed It.

[2] G. Zuckerman, The Man Who Solved the Market: How Jim Simons Launched the Quant Revolution.

[3] F. P. Kathryn M.E. Dominguez, “What defines ‘news’ in foreign exchange markets -”.

[4] C. D. J. L. S. Georgios Sermpinis, “forecasting and trading the EUR/USD exchange rate with stochastic Neural Network combination and time-varying leverage”.

[5] Y. B. &. G. H. Y. Lecun, “Deep Learning,” *Nature,* vol. 521, pp. 436-444, 2015.

[6] L. Y.-H. &. D. Mole, “The use of fundamental and tenchnical analysis by foreign exchange dealers: Hong Kong evidence,” *Journal of International Money and Finance,* vol. 17, no. 3, pp. 535-545, 1988.

[7] J. A. Murphy, “Futures Fund Performance: A test of effectiveness of technical analysis,” *Journal of Futures Markets,* pp. 175-185, 1986.

[8] L. M. &. M. Taylor, “The obstinate passion of foreign exchange proffesionals: Technical Analysis,” *Journal of Economic Literature,* no. 45, pp. 936-972, 2007.

[9] M. S. &. M. Taylor, “Under the microscope: The structure of the foreign exchange market,” *International Journal of Finance and Economics,* vol. 11, pp. 81-95, 2006.

[10] R. Lyons, “New Perspectives on FX markets: Order-flow analysis.,” *International Finance,* vol. 4, pp. 303-320, 2001.

[11] S. Taylor, “Trading Futures using a channel rule: A study of the predictive power of technical analysis with currency examples,” *Journal of Futures Markets,* vol. 14, pp. 215-235, 1994.

[12] P. W. &. R. D. C. Neely, “Forecasting the equity risk premium: The role of technical indicators,” *Management Science,* vol. 60, pp. 1772-1791, 2014.

[13] P. Hansen, “A test for superior predictive ability,” Working Paper - Brown University, 2003.

[14] M. Q. &. Y. Wu, “Technical trading-rule profitability, data snooping and reality check: Evidence from the foreign exchange market,” *Journal of Money, Credit and Banking,* vol. 38, pp. 2135-2158, 2006.

[15] S. S. &. J. C. Nima Zarrabi, “FX technical trading rules can be profitable sometimes!,” *International Review of Financial Analysis,* vol. 49, pp. 113-127, 2017.

[16] C. J. Neely, “Technical analysis in the foreign exchange market: A layman's guide,” *Review-Federal Reserve Bank of St. Louis,* vol. 79, no. 5, pp. 23-38, 1997.

[17] R. G. E. K. &. G. B. DF Silva, “Speeding up similarity search under dynamic time warping by pruning unpromising alignments,” *Data Mining and Knowledge Discovery,* vol. 32, no. 4, pp. 988-1016, 2018.

[18] J. Schmidhuber, “Deep Learning in Neural Networks : An overview,” *Neural Networks,* vol. 61, pp. 85-117, 2015.

[19] B. F. &. R. R. A. A Aliev Rafik, “Soft computing and its applications in business and economics,” *Studies in Fuzziness and Soft computing,* 2004.

[20] B. K. &. E. Vityaev, Data Mining in Finance : advances in relational and hybrid methods, 2006.

[21] A. B. &. M. O'Neill, Natural Computing in Computational Finance : Introduction, Berlin: Springer, 2009.

[22] B. Y. &. A. C. Ian Goodfellow, Deep Learning, http://www.deeplearningbook.org.

[23] G. Cybenko, “Approximations by superpositions of a sigmoidal function,” *Math Control Signals Systems,* vol. 2, no. 4, pp. 303-314, 1989.

[24] B. L. K. &. S. C. Kwasny, “Why tanh: choosing a sigmoidal function,” *IJCNN International Conference on Neural Networks,* vol. 4, pp. 578-581, 1992.

[25] V. N. &. G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” *Proceedings of the 27th International Conference on Machine Learning,* vol. 10, pp. 807-814, 2010.

[26] A. Y. H. &. A. Y. N. Andre L Maas, “Rectifier nonlinearities improve neural network acoustic models,” *Proceedings on International Conference on Machine Learning,* vol. 30, p. 3, 2013.

[27] Z. B. &. L. V. Q. Prajit Ramachandran, “Searching for activation functions,” no. https://arxiv.org/abs/1710.05941.

[28] B. Z. &. Q. V. L. Prajit Ramachandran, “Searching for activation functions,” https://arxiv.org/abs/1710.05941, 2017.

[29] H. R. &. S. Monro, “A stochastic approximation method,” *Annals of Mathematical Statistics,* vol. 22, no. 3, pp. 400-407, 1951.

[30] E. H. &. Y. S. J Duchi, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research,* vol. 12, pp. 2121-2159, 2011.

[31] S. H. &. J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[32] L. Z. Y. R. &. P. S. Qiu Xueheng, “Ensemble deep learning for regression and time forecasting,” in *IEEE Symposium on Computational Intelligence in Ensemble Learning*, 2014.

[33] A. G. P. &. R. A. K. Rafael Hrasko, “Time Series Prediction Using Restricted Boltzmann Machines and Backpropagation,” *Procedia Computer Science,* vol. 55, pp. 990-999, 2015.

[34] P. T. &. P. Wang, “Failure diagnosis using deep belief learning based health state classification,” *Reliability Engineering and System Safety,* vol. 115, pp. 124-135, 2013.

[35] H. Z. a. T. Y. C. N. U. o. S. I. o. S. S. S. Zhao Liu, “An NLP-PCA Based Trading Strategy On Chinese Stock Market”.

[36] E. Tham, “ Trusting the Social Media. 31st Australasian Finance and Banking Conference 2018”.

[37] M. S. R. a. M. S. R. Sheikh Shaugat Abdullah, “Analysis of Stock Market using Text Mining and Natural Language Processing”.

[38] B. a. W. M. Gebka, “Causality between trading volume and returns: Evidence from quantile regressions. International Review of Economics & Finance, 27, pp.144-159, 2013.”.