Deep Learning for Event Betting

Srinivasan Sathiamurthy **Rebecca Derham**

1 Introduction

In this research project, we explore mid-frequency event betting on currencies using deep learning techniques. Specifically, we train a model that has good predictive power over the returns of our chosen securities, and use this predictive power to create a trading strategy. Mid-frequency trading refers to making at most one trade per day, and event betting refers to trading on high-confidence beliefs in an attempt to ensure that we always make positive returns, rather than only maximizing the expected value of profits.

We hope that this process will yield several results useful to portfolio managers as well as to financial researchers. First, portfolio managers who trade on a daily scale are sensitive to large losses, as their clients then would want to take their money out the next day. In addition, it is harder to make positive returns consistently when only trading once per day as opposed to when trading at a higher frequency. As a result, a methodology for generating reliable mid-frequency strategies would prove highly valuable.

In addition, deep learning methods will be able to capture more relationships between features without having to hand craft them as in traditional financial models. While the value of this has already been seen in many other fields such as computer vision, deep learning has not been as thoroughly explored in mid-frequency trading, mainly due to lack of data. However, we will overcome this limitation in three main ways. First, we will group data from several individual securities into one dataset which can be used to train a single model, rather than limiting ourselves to training on only one security's data. Second, we will discretize our outcome variable by predicting whether the returns will be positive, negative, or neutral, as either positive or negative returns are profitable while neutral ones are not; a trading strategy is determined by whether a potential trade will be profitable while the magnitude of the returns are less important, and so this simplification will reduce overfitting while preserving the practicality of our results. Finally, we will bet on only high-confidence beliefs, which will further reduce overfitting and thus provide an advantage over traditional regression-based models. Our extension of our baseline model focuses on avoiding the overfitting aspect in particular, both in terms of retaining model accuracy past the training dataset, and also also maintaining pnl of our resulting trading strategies.

2 Background

Our baseline model and our final models use the same framework of data, and build off of similar financial intuition. The major differences are the graphical structures we encoded in the neural network architectures, and the other forms of regularization we experimented with.

To summarize the set up of our baseline model, we chose to trade G10 currencies, as currencies in general are highly correlated with global economic events and the G10 currencies are less subject to idiosyncrasies than those of emerging markets. We use the past four daily prices as well as the first three moments of each asset's historical price information as features, as well as a variety of macroeconomic indicators. We decided to use data from January 2008 to December 2020 as training data, and from January 2021 to September 2023 as validation data. This gave us approximately 4000 training datapoints and 500 validation datapoints per currency.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

We then quantiled the returns conditioned on holding for five days at each time step into three evenly distributed categories: below the 33% quantile, which corresponds to the lowest third of returns, between 33% and 66% quantiles, which corresponds to medium returns, and above 66% quantile, which corresponds to highly positive returns. As our baseline model, we fit a simple 2 layer classification MLP architecture with the hidden layers having 30 and 10 nodes respectively and a cross entropy loss function to predict these classes based on the other features in the dataset (described in Section 3). This resulted a training accuracy of 61.6%, and a validation accuracy of 49.12%; the accuracy curves for this model shown in Figure 1. We also considered the confusion matrices for the training and validation data (Figure 2).

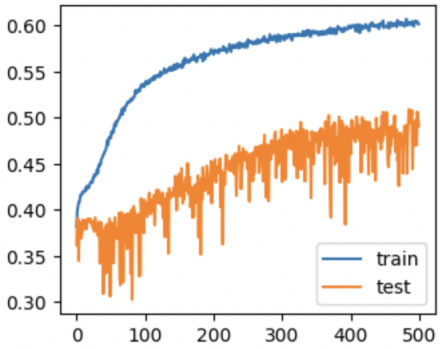

Figure 1: Accuracies for Proposed Model

0.6681	0.2303	0.1016
0.2435	0.5003	0.2561
0.0825	0.2304	0.6870

Training

0.8262	0.1555	0.0183
0.3274	0.4440	0.2286
0.1727	0.2946	0.5327

Testing

Rows correspond to truth, columns to predictions

Figure 2: Confusion Matrices for Proposed Model

Although this initial model's test accuracy is only around 50% (compared to the random-guessing accuracy of 33%), the confusion matrix yielded promising results; the model is able to predict downturns of the currency values with 82.6% accuracy, and upturns with a 53% accuracy in the testing dataset, which is what is most important when it comes to building a trading strategy to be used on time frames after that of the training data.

3 Related Work

To first address our data selection, it is well known that availability of information has a large effect on markets. When important indicators are released, investors with vastly different requirements, interpretations of the data, and opinions about the market immediately begin trading (Basdekidou, 2017). Ultimately, this results in overreactions to market information, which leads to the existence of profitable strategies trading at this medium frequency (Park, 2017). As such, we use publicly available domestic macroeconomic indicators as part of our feature set.

Next, we must transform this information into actual predictions of future asset prices. Researchers have achieved success with various deep learning techniques across multiple time horizons. The main reason for deep learning's success is that it learns suitable representations directly from raw data, as opposed to conventional methods whose features are engineered manually and combined with domain expertise (Kolm, Turiel, and Westray 2021). However, a lack of data, especially at the mid-frequency level, pervades this space, which has led to mixed results (Li et al. 2010). In our case, since the scope of our research considers trading on a daily scale, we are limited to considering events that occur roughly once per trading day, which is about 252 data points per security per year. Macroeconomic events tend to occur less frequently than this, and don't necessarily yield useable information at every occurrence. It then follows that draw-downs pose a much higher risk in medium frequency trading than in high frequency trading, since our events do not occur often enough for us to have the luxury of relying on a positive expected value of our strategy's returns to offset bad draw-downs in the short to medium term. To circumvent this problem effectively, many researchers have discretized the trading process by labeling each time stamp as likely to increase, decrease, or remain relatively stable over some fixed period, and attempting to accurately predict these three classes (Passalis et al. 2020).

The most common models used are combinations of CNNs, LSTMs, and MLPs. Zhang et al, 2018, explored the architecture of a CNN followed by an LSTM for better memory retention and obtained good results. However, Briola et al, 2020, discovered that a large enough MLP performs just as well due to MLPs being universal function approximators. However, one gap we found in our research is

that most of these existing approaches use high frequency data in the minute to nanosecond range (Huang et al. 2011), while our investment time horizons are much longer than that.

For our model, we are most interested in the work of Passalis et al., since it incorporates the probability classification structure we wish to ultimately trade off of, and also considers various look-back periods for each feature.

4 Methods

The data we used and the train/test split were the same as our baseline model. Our main focus for the project's extension beyond the baseline was experimenting with graphical structures to better encode our financial intuition and domain knowledge, as well as other forms of regularization.

First, we attempted a transfer learning scheme to solve two problems simultaneously: the lack of enough train/validation data, and the possibility of our model overfitting to black swan events. Our first implementation of this involved stacking the training and validation data for every currency, and normalizing by the values for each feature across all currencies in the training set. When we did so, we initially achieved much better results than our preliminary model, which normalized features per currency and aggregated them to form our train/validation data sets. However, we realized that this was due to the prices of Swedish, Norwegian and Japanese currencies having a significantly different magnitude than the other G10 currencies in our dataset. As a result, they were essentially acting as a regularizer for our model by preventing it from overfitting on the other currencies, since the features for these smaller currencies amounted to noise compared to the others. In light of these findings, we decided to focus only on trading AUD, CAD, GBP, EUR, and CHF against USD, and normalize features per currency. Instead of allowing the other three currencies to act as regularizers, we wish to directly encode regularization in our model structure for better model interpretability.

The next part of our baseline extension focused on achieving proper regularization via experimenting with the graphical structure of our neural network. We note here that we trained another "baseline" model as a fully-connected neural network with similar structure to our graphical models, as it would serve as a better comparison than our initially proposed MLP. We therefore refer to this new model as our "baseline" for the rest of the paper.

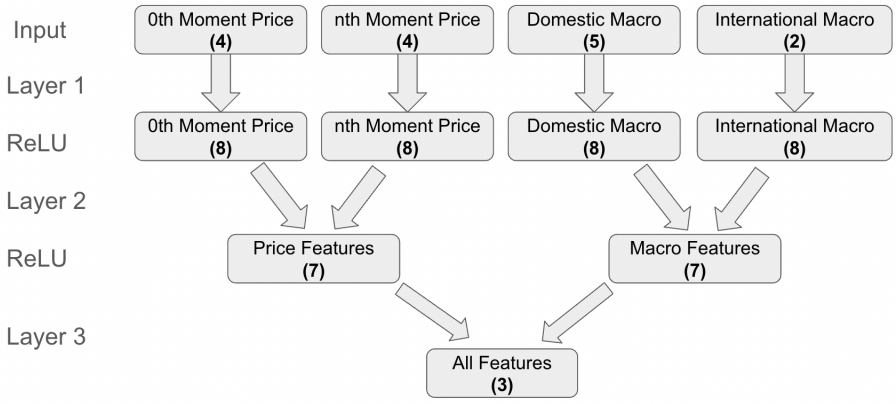

Figure 3: Graphical Structure for Model 1

For our first model ("Model 1"), we focused on controlling the relationships for the model to capture such that it would be more interpretable. We did so by first aggregating the zeroth moment price features (prices lagged by 1, 2, 3, and 4 days), the $n > 1$th moment price features (5 and 10 day moving average, 14-day moving volatility, and 14-day moving skew), US domestic macroeconomic features (VIX, 10/2 treasury yield spread, Case-Schiller index, unemployment rate, and average hours worked in the manufacturing sector), and US international macroeconomic factors (CPI and trade balance). Each of these groups of features were fed through a unique first layer to obtain a new set of features. We then fed these outputs into a second layer which combined all the price features and all the macro economic features. These ultimately combined into the last layer which outputted

the probabilities of the three quantile classes. A diagram of the graphical structure with more node information is shown in Figure 3. Note that the numbers in parentheses correspond to the number of individual nodes within each group, and each grey arrow corresponds to a fully connected layer between the two sets of nodes.

Our second model ("Model 2") has less regularization, and instead directly aggregates all the price information together, and all the macro information together separately in the first layer, and then combines them in the second layer before predicting class probabilities in the third layer. This ultimately still has the same number of nodes per layer, but there are more edges that were not present in Model 1 to control for more specific relationships. A diagram of this model's graphical structure is shown in Figure 4.

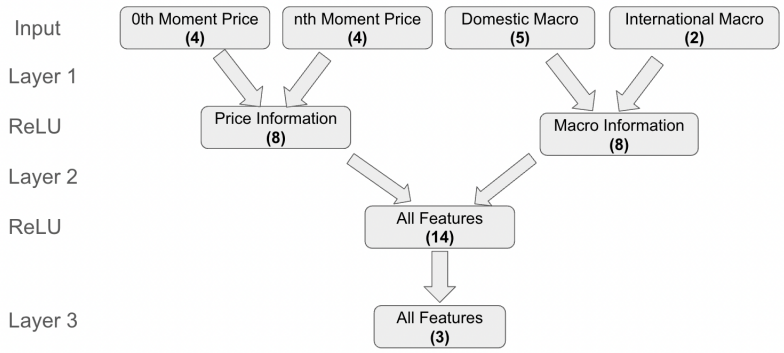

Figure 4: Graphical Structure for Model 2

In our third model ("Model 3"), we opted to explore other forms of regularization to better determine the strength of our more restrictive yet intuitive graphical structure. To do so, we started with the same structure of Model 2 and applied dropout with probability 0.2 on the first and second linear layers. A diagram of this model's graphical structure is shown in Figure 5.

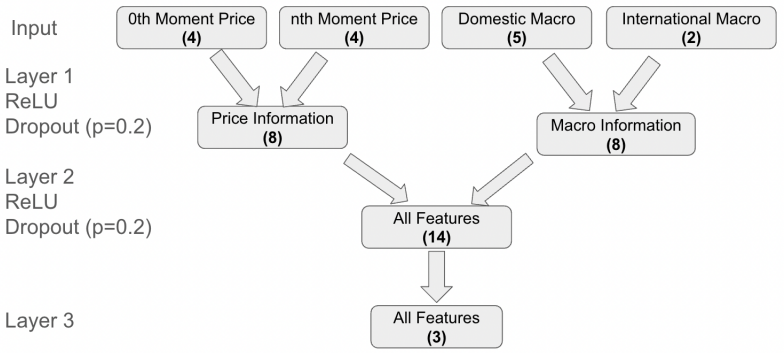

Figure 5: Graphical Structure for Model 3

In summary, we trained three graphical models. The most general is Model 2 with a structure which is slightly more restrictive than a fully connected network. Model 1 adds regularization by using a more complicated graphical structure which limits learnable relationships, and Model 3 adds regularization via dropout.

After training these models, we sought to extend the research more deeply into the financial domain by backtesting a simple strategy using these models on our in-sample data (2007/01/01 to 2020/12/31) and out-sample data (2021/01/01 to 2022/12/20) for each currency pair of interest (AUD/USD, CAD/USD, GBP/USD, CHF/USD, and EUR/USD). To create each of these strategies, we generated a prediction from the model at every time stamp assuming we were given the features for that day, and by comparing the model's predicted probabilities of quantile classes to a threshold value (determined later), we either longed or shorted the currency pair between the end of the current day and the end

of the next day. We calculated the annualized returns which would be achieved by following this strategy, as well as the annualized information ratio, which is the annualized returns divided by the annualized volatility of the strategy (an information ratio of 1 is mediocre, a value of 2 is good, and a value of 3 or higher is excellent). After some testing, we found that a threshold probability of 0.4 for both longing and shorting worked best across all models and currencies.

5 Results

We first examine the results of training our models, and then discuss the trading strategies we created using their predictions.

5.1 Models

We trained a total of four models: our baseline model, which was a fully connected neural network, followed by three different iterations of graphical models with various structures. The training and testing accuracies of each are shown in Figure 6. First, looking at the training accuracy overall, although there was a lot of variance present in our training/test accuracy curves, some consistent trends emerged, such as higher regularization leading to lower training accuracy scores. This was evident since Model 2 had no dropout as opposed to Model 3, and it had nonlinear relationships between the two types of price and macro features that Model 1 did not. Additionally, dropout seemed to be a stronger regularizer, as the training accuracy for Model 3 (58.9%) was lower than that of Model 1 (64.0%). However, despite both Model 1 and Model 2 capturing fewer nonlinear relationships than the baseline model, they had similar training accuracies as the baseline model, which suggests that the additional nonlinear relationships the baseline model captured were not relevant, and it is indeed fruitful to encode one's financial domain knowledge into the neural network architecture in a graphical manner.

Looking at the testing accuracy, Model 1 and Model 2 exceeded that of the baseline's accuracy by more than 10%, while Model 3's testing accuracy was just 1% below that of the baseline. Additionally, Models 1, 2, and 3 all had testing accuracy curves that showed continued improvement for the later epochs, and would have likely increased past their current thresholds if the number of epochs increased, which was not a phenomenon present in the testing accuracy curve of the baseline model. This finding further supports the need of regularization for our problem, and shows the efficacy of constraining the graphical structure of the neural network.

Additionally, comparing the testing confusion matrix (Figure 7) and focusing on the classification rates for the first (downward price movement) and third quantile (upward price movement) ranges, Models 1, 2, and 3 all have better classification profiles than that of the baseline model. However, their performance with respect to classifying these two categories decays with the amount of regularization among these three, with Model 2 performing significantly better than Model 1, which in turn is slightly better than Model 3.

5.2 Trading Strategies

For the second part of our analysis, we compare and contrast the performances of the strategies associated with each model, pnl plot-wise as well as their associated metrics, insample and outsample. See Figure 8 for a sample of the in-sample and out-of-sample returns of our strategies; each row corresponds to a different model (baseline, and then models 1, 2, and 3). While we created a strategy for each of the 10 currencies we fit the models on, we only include plots of the EUR strategy due to space constraints.

One immediate observation is that the information ratios of every model's associated strategy decayed from insample to outsample significantly for each of the five currencies considered. This suggests that all of our models are overfitting significantly, at least on the strategy side. In particular, our baseline model overfits significantly more than any of the other three models, with its in-sample information ratios 5-6 times bigger than that of the other models for each currency. However, it also performs better out of sample than Models 1 and 2 for most currencies, and is only bested by Model 3. Additionally, it is also worth noting that Model 3's strategies in general bet on fewer events than any of the other strategies (including the baseline), which raises the question of whether the increased information ratio for Model 3's out of sample strategies might just be due to a statistical anomaly.

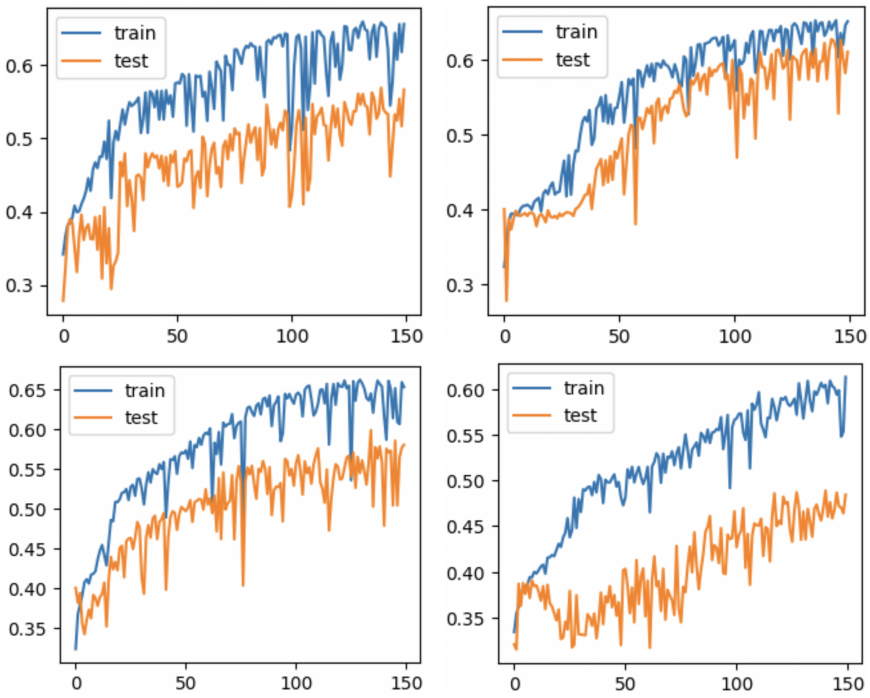

Figure 6: Accuracy per Epoch: Baseline, Model 1, Model 2, Model 3 (top left to bottom right)

0.7654	0.1872	0.0474
0.3304	0.4574	0.2122
0.0677	0.4086	0.5238

Baseline Model

0.7293	0.2313	0.0395
0.2234	0.5448	0.2318
0.0205	0.2496	0.7300

Model 1

0.8885	0.1945	0.0070
0.3048	0.4767	0.2186
0.0227	0.2540	0.7234

Model 2

0.6713	0.2897	0.0391
0.2818	0.4463	0.2719
0.0085	0.2128	0.7787

Model 3

Figure 7: Confusion Matrices on Testing Data
Rows correspond to true values, columns to predicted

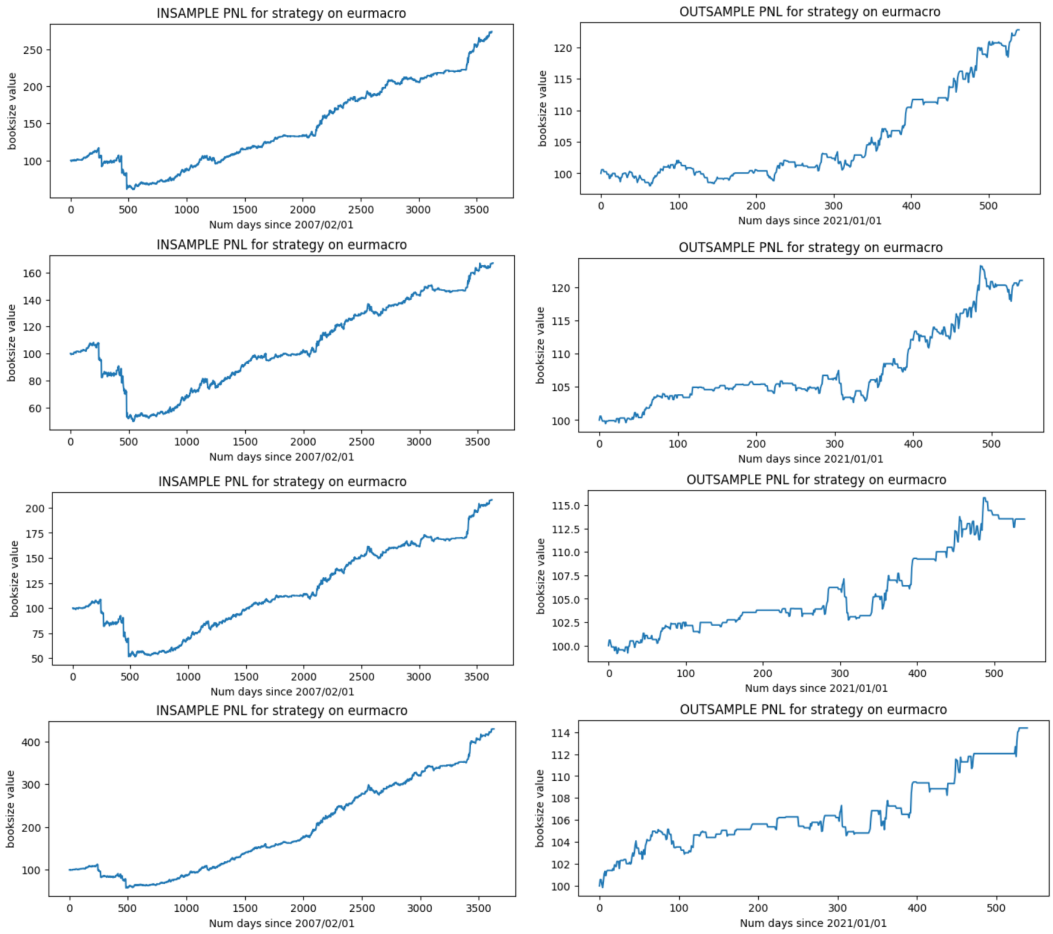

Figure 8: Sample of trading strategies:
Rows correspond to baseline and then Models 1, 2, and 3

6 Discussion & Analysis

6.1 Analysis

In summary, we determined that it was fruitful to constrain the graphical structure using financial domain knowledge, as all three of our models have better classification accuracy than the baseline model. In addition, we found it useful to incorporate other forms of regularization, in particular dropout, since ultimately all of our models still appear to overfit significantly in terms of the strategy side.

Looking at the pnl plots of our strategies, one could also bring issue to the fact that the in-sample regimes and out-sample regimes differ vastly. For instance, the FED interest rates were near zero throughout most of the in-sample period, while the FED aggressively hiked interest rates during our out of sample period. It's well known that the FED interest rates have ripple effects throughout the world economy, and therefore the model needs to capture entirely different relationships to predict the directionality of currency movements in the out of sample period than in the in-sample period. The macro data variables themselves are also subject to this issue, since their values are also impacted by this regime shift, and thus the model needs to know how to deal with such features, that likely have a much different magnitude than their values in sample.

Despite this phenomenon, it seems that our model is able to predict the directionality of currency movements with nontrivial expected value, which suggests that either the half of our model that incorporates pricing information is stronger than the macro feature weights, or that the out of sample

macro features still contribute some input to our model's predictive power. It remains to be shown if one of these two hypothesis bests the other.

6.2 Limitations & Next Steps

One of the major limitations we perceive in comparing the performance of strategies is determining a consistent and quantitative criterion for determine whether one event betting strategy is statistically better than another similar strategy. This involves taking several factors into account, such as the number of events each strategy bets on, how accurate each strategy classifies the events, and how consistent the events are over time. Our model attempts to take care of accuracy, but we still need to find measures to track the other two metrics, much more, optimize for them.

Additionally, we still need to resolve the fundamental issue of ovefitting for insample at the strategy level using just our models. Since we didn't tune our strategies in sample, the overfitting can be entirely attributed to that of model performance. As a result, it suffices to find a sufficiently regularized model that captures the right edge relationships that has consistent performance across our insample and outsample regimes. The limitations to this that we faced was a lack of higher frequency data, as well as not having a diverse enough range of features with which we could explore more advanced graphical structures like that of Model 1.

References

[1] Park, Seoungbyung (2017) *Factor Based Statistical Arbitrage in the U.S. Equity Market with a Model Breakdown Detection Process*, pp. 419. Master's Theses (2009-)

[2] Passalis, N., Tefas, A., Kannaiainen, J., Gabbouj, M. & Iosifidis, A. (2020) *Temporal Logistic Neural Bag-of-Features for Financial Time Series Forecasting Leveraging Limit Order Book Data*, pp. 183–189. Elsevier

[3] Kolm, P. & Westray, N. (2021) *Information Content Of Cross- Sectional Multilevel Order Flow Imbalance*

[4] Yuhong, L. & Ma, W. (2010) *Applications Of Artificial Neural Networks In Financial Economics: A Survey*, pp. 211-214 Vol 1. 2010 International Symposium On Computational Intelligence And Design. IEEE

[5] Briola, A., Turiel, J. & Aste, T. (2020) *Deep Learning Modeling of Limit Order Book: A Comparative Perspective* arXiv

[6] Huang et al. (2011) *LOBSTER: Limit Order Book Reconstruction System* SSRN

[7] Zhang et al. (2018) *BDLOB: Bayesian Deep Convolutional Neural Networks For Limit Order Books* arXiv

[8] Basdekidou, V. (2017) *Nonfarm Employment Report Trading with Binary Options and Temporal Functionalities*, pp. 15–24. Annales Universitatis Apulensis Series Oeconomica