Learning low-frequency temporal patterns for quantitative trading

Joel Da Costa
Department of Statistical Sciences
University of Cape Town
Cape Town, South Africa
joeldacosta@gmail.com
https://orcid.org/0000-0001-7821-6635

Tim Gebbie
Department of Statistical Sciences
University of Cape Town
Cape Town, South Africa
tim.gebbie@uct.ac.za
https://orcid.org/0000-0002-4061-2621

Abstract—We consider the viability of a modularised mechanistic online machine learning framework to learn signals in low-frequency financial time series data. The framework is proved on daily sampled closing time-series data from JSE equity markets. The input patterns are vectors of pre-processed sequences of daily, weekly and monthly or quarterly sampled feature changes. The data processing is split into a batch processed step where features are learnt using a stacked autoencoder via unsupervised learning, and then both batch and online supervised learning are carried out using these learnt features, with the output being a point prediction of measured time-series feature fluctuations. Weight initializations are implemented with restricted Boltzmann machine pre-training, and variance based initializations. Historical simulations are then run using an online feedforward neural network initialised with the weights from the batch training and validation step. The validity of results are considered under a rigorous assessment of backtest overfitting using both combinatorially symmetrical cross validation and probabilistic robust non-parametric validation techniques. Results are used to develop a view on the phenomenology of financial markets and the value of complex historical data-analysis for trading under the unstable adaptive dynamics that characterise financial markets.

Index Terms—online learning, feature selection, pattern prediction, backtest overfitting

I. INTRODUCTION

A. Technical Analysis

Technical analysis is a financial analytical practice that makes use of past price data in order to identify market states and forecast future price movements based on past movements. The techniques typically rely on past market data (price and volume), rather than company assessments using fundamental analysis. We explore the idea that technical analysis has merit in exposing market inefficiencies when they are signified by repeated feature time-series patterns [1], [2].

Financial markets have been shown to be complex and adaptive systems where the effects of interaction between participants can be highly non-linear [3], but they may also have combinations of top-down and bottom-up sources of information and interaction that mix in vast numbers of interactions mediated by numerous flocks of heterogeneous strategic agents that constitute modern financial markets [4]. Complex and dynamic systems such as these may often exist at multiple ‘order-disorder borders’ - and they will then generate certain non-random patterns and internal organisation on different averaging scales. Two key price generation processes have emerged: the low-frequency domain (the result of sequences of closing auctions generating prices), and the high-frequency intra-day domain driven by order-flow itself. Here we consider low-frequency daily sampled data that is the result of the price discovery from closing auctions.

Even at low-frequency, identifying patterns and structure is simultaneously reasonable and notoriously difficult. While it is often clear in hindsight that patterns exist, the amount of noise and non-linearity in the system can make prediction challenging. Fittingly, neural networks are a popular choices for modelling within financial markets because of their ability to perform well as universal approximators [5].

Practical approaches to money management within the realities of adapting and changing market systems increasingly favour online methods, in particular [6] explored the application of online learning models in this space in the South African market to show that direct (but simplistic) online pattern-learning is able to identify and potentially exploit trading opportunities on the JSE through the assessment of Open High Low Close (OHLC) data. This was extended by [7] to more directly explore the use of online learning applied to optimizing parameters for traditional technical trading indicators as applied to maximising wealth trading zero-cost portfolio strategies.

The work presented here fits into the growing body of work which considers mechanistic and brute-force approaches of applying machine learning models to financial market data. The complexity, non-linearity, noise and stability of financial markets are highlighted through both the successes and challenges found in training these models. These difficult dynamics, and their notable difference when compared to other popular areas of ML research - which are often around Independently and Identically Distributed (IID) datasets - present fundamental problems to be explored; both in terms of prediction efficacy as well as validation. We present a framework using batch offline and online learning on JSE closing data, feature extraction and robust non-parametric validation techniques.
B. Backtesting and Model Validation

Financial academic literature is currently facing a problem in terms of validation and verification of results. Trading strategy profitability has typically been proven using historical simulations, or “backtests”. However, the recent advances in technology and algorithms available to construct these strategies have resulted in researchers being able to test increasing numbers of variations of factors. This has made it increasingly difficult to control for spurious results. The problem is so extensive that some meta-research papers suggest that most published research findings are false [8].

The standard way of implementing backtests is to split the data into two portions: an In Sample (IS) portion which is used to train the model, and an Out of Sample (OOS) portion which is used to test the model and validate results. If vast numbers of different model configurations are tested, then it is only a matter of time before false positives occurs with high performance both IS and OOS (i.e. overfitting) [9], [10]. The nature of financial data makes it difficult to resolve these issues effectively. There is a low signal-to-noise ratio in a dynamic and adaptive system with only one true data sequence. Traditional hypothesis testing frameworks (e.g. Neyman-Pearson) are not sufficient in this context making more sophisticated techniques necessary.

The problem of overfitting is not novel. However, in a machine learning context, frameworks are often not suited to trading schedules with a random frequency structure. They do not account for overfitting outside of the output parameters nor take into consideration the number of trials attempted. The common ‘hold-out’ strategy is where a certain portion of the dataset is reserved for testing true OOS performance. Numerous problems have been pointed out with this approach. The data is often used regardless, and awareness of the movements in the data may influence strategy and test design [11]. For small samples, a hold-out strategy may be too short to be conclusive [12]. Even for large samples, it results in the most recent data (which is arguably the most pertinent) not being used for model selection [9], [13]. We present a novel application of existing sophisticated validation methods (see Section [11]) to a machine learning framework.

II. FRAMEWORK IMPLEMENTATION

A. Full Framework Process

The framework implementation brings several ideas together: i.) SAE based feature selection, ii.) Deep Learning with pre-training and weight initialization, and iii.) Online Learning and Backtest Overfitting Validation. The learning part of the framework consists of two phases: I. batch learning and II. online learning.

I. Batch learning phase: IS data is used to train Stacked Autoencoder (SAE) networks which in turn are used to perform feature reduction for Feedforward Neural Networks (FNNs) learning the price fluctuation predictions in the IS data. Both are trained using Stochastic Gradient Descent (SGD).

II. Online learning phase: The batch trained FFN networks are used to predict price fluctuations on OOS data through Online Gradient Descent (OGD). These online predictions are then sequentially used to simulate trading in a Money Management System (MMS), which in turn generates simulated returns. Finally, the MMS returns are used as input for the Probability of Backtest Overfitting (PBO) and Deflated Sharpe Ratio (DSR) techniques in order to validate the legitimacy of the framework.

The process has two key principles: First, implementing a generalised version of a system which could offer exploration of more complex techniques. Second, ensuring an effective modularisation of steps such that the process can be reconfigured accordingly while maintaining its integrity. In doing so, a separable system is created which brings together all key concepts. We aim to delivery the simplest implementation of a complex framework such that the effects of individual components can be properly assessed and developed. The full process flow can be seen in Figure [7] in the Appendix.

B. Data Processing

Datasets are transformed into log feature differences and aggregated to include the changes over rolling window periods. The log feature fluctuations are processed for each asset’s closing price at each time point $i$. Log fluctuations take compounding into account in a systematic way and are symmetric in terms of gains and losses; the log transformation also provides an ergodic time series. The rolling window summations are calculated for the past for input data e.g. (1, 5, 20) days, and in the future for predicted output - 5 days in this paper. These are calculated as summations of the log differences $\Delta p_i$ such that for $d$ days at timepoint $t$:

$$p_{d,t}^- = \sum_{i=t-d}^{t} \Delta p_i, \quad p_{t,d}^+ = \sum_{i=t+1}^{t+d} \Delta p_i.$$  

(1)

The aggregations are scaled using a modified Normalization, where the min and max values are determined by the Training portion of the dataset, but applied to both the Training and Prediction portions. This emulates a production implementation where future data is unknown. The log-differenced, aggregated and scaled data is then used as the input for the neural network models. Predicted outputs have the scaling and log differencing reversed in order to reconstruct the actual price point predictions for performance assessment.

C. Network Weight Initialization

The problem of vanishing and exploding gradients has been one of the primary barriers to deep learning with neural networks. The approach of greedy layer wise pre-training for SAEs was suggested by [14], which allowed much deeper layered networks to be trained than previously possible [15]. Once the SAE is trained, a final output layer is added and the entire network can then be fine-tuned through back-propagation without suffering such performance degradation from vanishing or exploding gradients, enabling training of both SAE and FNN networks [16], [17].
In Section V we see that batch training on historical data has a limited benefit, which gives primacy to weight initialization techniques for machine learning of financial time series. Initial results found that RBM pre-training for sigmoid SAE networks (as described by [17]) had detrimental effects on network performance. This suggests that the IID assumptions and the different loss functions result in the financial time series data used being pathological for RBM pre-training, and is discussed further in [18]. It has also been shown that pre-training may largely act as a prior which may not be necessary if large enough datasets are available [19], [20]. In the context of financial time series this prior can explain the poor performance. For these reasons we focus on variance based weight initialisations developed by [21] and [22]. These have simpler implementations, faster computation and enables initialization for non-probabilistic activation functions, such as ReLU.

Concretely, we use a modified He initialization: “He-Adjusted”. This initialization uses a mean of the input and output layers to scale the weight variance. For networks with constant layer sizes, the initialization is the same as He [22]. For SAE networks, where layer size changes by definition, the He-Adjusted initialization results in better sized weights. For $n$, the number of nodes in a layer, we initialize using:

$$w_{ij} \sim U(-r, r), \text{ with } r = \sqrt{\frac{12}{(n_i + n_j)}}. \quad (2)$$

D. Unsupervised Learning: SAE Training

The benefit of the modularised system is emphasised here, as the SAE training will not suffer from limitations due to backtesting considerations: any amount of configurations can be tested for feature extraction without concern. The best chosen SAE networks (based on a minimum Mean Squared Error (MSE) score) are used to reprocess both the Training and Prediction datasets such that the input is encoded, and the output is as before. These encoded datasets can then be used for the subsequent steps in the framework. We did not implement a step to update the SAE, but results detailed in Section V suggest that this would be an important inclusion in a production system.

E. Supervised Learning: Prediction Network Training

Once the predictive network is trained on IS data using SGD, the OGD process is run through the encoded Prediction dataset in order to generate the predictions for the asset prices that the model produces - thus emulating what would have occurred in a live environment.

F. Money Management Strategy

The MMS follows an arithmetic long strategy of buying any asset for which the predicted price is above the current price, and selling the stock at the prediction horizon regardless. Trading costs were included at 10% capital costs per annum for borrowing to purchase, and 0.45% for transaction costs as per [6], without taking liquidity effects into account. The naive approach is taken purposefully so as not to bias the perspective of the system as a whole by the effects of an impactful trading strategy. It is important, in the interest of effective optimization, that the pattern prediction of the system is not tightly coupled with making it profitable. Thus, the modularity of the system is continued with a separation between the prediction signal and the MMS implementation.

G. Validation

Validation is implemented with Combinatorially Symmetric Cross-Validation (CSCV) [9] that uses the IS and OOS returns from the MMS, which in turn uses the prices from the prediction network; which is a somewhat novel application. Conceptually, the whole system comes into place here, as the results from the CSCV process are now indicative of not only backtest overfitting in the trading strategy, but also in the prediction network and without having to consider the impact of the many configuration tests for feature extraction. A modified version of ONC was run, with reduced cluster exploration. As noted in [18], this did not appear to affect results.

III. ASSESSMENT METHODOLOGIES

A. Probability of Backtest Overfitting

CSCV was developed by [9] as a robust approach to assessing backtest overfitting. Their research defines backtest overfitting as having occurred when the strategy selection which maximizes IS performance systematically underperforms the median OOS performance in comparison to the remaining configurations. They use this definition to develop a framework which measures the probability of such an event occurring, where the sample space is the combined pairs of IS and OOS trading performance measures. The PBO is then established as the likelihood of a configuration underperforming the median OOS while outperforming IS.

The CSCV methodology is generic, model-free and non-parametric, allowing it to arguably be used in any model case. By recombining the slices of available data, both the training and testing sets are of equal size (advantageous for comparing performance statistics). The symmetry of the set combinations in CSCV ensure that performance degradation is only as a result of overfitting, and not arbitrary differences in data sets. There is no requirement of a hold-out set, which removes potential credibility issues regarding whether the holdout set was treated appropriately or not. The logit distribution developed through the assessment offers a useful view on the robustness of the strategies used and the nature of the PBO score.

The PBO can be estimated using the CSCV method results, which provides an estimate rate at which the best IS strategies underperform the median of OOS trials. [9] extend this to show that with models overfitting to backtest data noise, there comes a point where seeking increased IS performance is detrimental to the goal of improving OOS performance.

B. Deflated Sharpe Ratio

The Sharpe Ratio ($SR$) is based on the assumption that the returns used are the result of a single trial, as is the case
for most standard performance measures. In consideration of the issues laid out in [18], it then becomes a misrepresentative performance measure. [23] developed the Probabilistic Sharpe Ratio (PSR) which estimates the likelihood that an observed best estimated \( \hat{SR} \) exceeds a provided benchmark \( SR^* \) (which might be expected from variance in the trials). It is worth emphasising the distinction in investment strategies between a Family Wise Error Rate (FWER), which is the probability that one or more false positives occur, and a False Discovery Rate (FDR), which is the ratio of false positives to predicted positives. Investment strategy generations will tend to rely on the single best approach produced, and so must consider FWER. [24] further developed the False Strategy Theorem (FST) with this in mind, allowing the assessment of whether a presented strategy is a false positive or not.

The DSR calculates the likelihood that the true \( SR \) is positive under consideration of numerous trials being tested \([24]\). The DSR can be estimated using the PSR methodology as \( PSR[SR^*] \) where the benchmark Sharpe ratio, \( SR^* \), is calculated using the False Strategy Theorem. The calculation of \( SR^* \) requires both the variance of trial \( SR \) values and the number of independent trials, which are not typically considered and where determining independence is challenging. [25] aim to resolve this with the Optimal Number Clusters (ONC) algorithm, a modified K-means methodology of clustering strategies and trial results. This clustering allows an estimation then of both the variance and number of trials, which in turn allows the DSR to be calculated. With this as a confidence level, one can accept or reject the notion that the observed \( \hat{SR} \) is positive.

IV. EXPERIMENT PROCESS

A. Data & Software

Datasets were constructed using JSE closing price relative data for 2003-2018 [26], with a 60:40 split on the Training Prediction subsets. The closing price dataset consisted of 10 assets from the JSE Top 40: AGL, BIL, IMP, FSR, SBK, REM, INP, SNH, MTN, DDT (coming from a variety of sectors). More source information, data snapshots and price charts are available in [18].

The software libraries, written in Julia, were produced for all the training, experimentation and recording of results. These are discussed extensively in [18], and made available online [27].

B. Parameter Space Exploration

The parameter space is explored using a phased grid search approach. For each stage, the relevant parameters are each specified as a set of values, and all sets are then used to generate the full combinatorial space, such that each possible combination of the specified parameters is tested.

1) Stage 1: The data configuration (i.e. data windows, prediction point, scaling, data split points) as well as the SAE configuration (network size, learning rates, learning optimization parameters, SGD epochs) are set in Stage 1 and used to train the SAE networks.

2) Stage 2: The preferred SAEs are chosen from Stage 1, and determine the data configuration used for Stage 2. These are then used to encode the datasets, which will be used for FFN training. The FFN SGD and OGD parameters are set in this stage (network size, learning rates, SGD epochs etc.), and will be combined combinatorially with the SAEs that were chosen for testing as well.

V. FINDINGS

A. Value of Historical Data and Training

We expected that the IS batch training using SGD for the predictive network would improve OOS P&L performance. Theoretically, the training on historical data might prime the network for predicting future data. However, we found that the effects of IS training had limited benefit. We ran experimental trials to test the hypothesis that the amount of historical IS data available is of limited benefit. We found the P&L results validate this idea, as seen in Figures [1] and [2]. We saw that extensive training on past data may be akin to pre-training network weights at best, and counterproductive in overfitting to dynamics that no longer exist at worst. This highlights the complexity and dynamic nature of financial time series, where past relations and behaviours are not necessarily indicative of present state. It follows that the primary determinants of OOS P&L are those present in the OGD (OOS) learning phase: the OGD learning rate, the data horizon aggregations, and the SAE feature selection.

This fits well with research showing that online algorithms typically perform as fast as batch algorithms during the ‘search’ phase of parameter optimization, but that ‘final’ phase convergence tended to fluctuate around the optima due to the noise present in single sample gradients [28], [29]. [30] showed that it is actually more practical to consider the convergence towards the parameters of the optima, rather than the optima itself (as defined by the cost function) - the difference between the learning speed and optimization speed, respectively. Online learning methods are thus well suited to financial market modelling using neural networks. They allow effective and efficient incremental updates as more recent (and relevant) data becomes available. Further, the increased learning speed over optima convergence makes them a fitting choice when data is non-IID and constantly changing.

B. Primary Determinants of P&L

Input data was scaled to 3 different configurations to assess the effects of shorter and longer data horizons, using SAE MSE and predictive OOS P&L to assess performance. The configurations tested (in trading day window periods) were: 1.) horizon-tuple 1 with [1, 5, 20], 2.) horizon-tuple 2. with [5, 20, 60], and 3.) horizon-tuple 3. with [10, 20, 60]. The SAE networks were only trained on IS data, and not updated afterwards. As noted below, an effective SAE feature selection in this process is an optimization that may be limited to a certain time period and may not generalise well OOS. We also found that lower variance, in the shorter horizon aggregations,
Fig. 1. These results show OOS P&L grouped by the number of epochs in the SGD IS training phase. Here 100 Epochs offered the best overall performance, and further training to 500 or 1000 epochs degraded performance due to the network overfitting on the IS data. The results show that the benefit of historical data is limited - having networks become better at learning return relationships from 10 years in the past did not lead to increased OOS P&L for more current data. The small difference in the upper half of observations between 10 and 100 Epochs further emphasises this point.

Fig. 2. To further explore the effect of IS training on historical data, configurations were run with a percentage of the usual training data excluded, with the P&L results grouped above. The exclusion of up to 80% of the IS training data resulted in only a 2.2% drop in median OOS P&L for those networks. The training in these instances was not adjusted to increase the number of epochs according to the size of IS data, and so the configurations with more data excluded were also in essence trained less.

resulted in easier replication; while longer horizons are more difficult (as indicated through MSE scores). The reproduction differences are discussed in [18].

We observed strong interactions between the SAE feature sizes, FNN OGD learning rates and data horizon aggregation configurations. The performance differences seen further emphasise the unstable nature of financial systems. Generally, the FNN OGD learning rate had the largest impact on OOS performance, and demonstrates the benefits in being able to adapt quickly to new information, as seen in Figure 3. As the SAE feature size decreased from 25 to 10, SAEs learnt longer term features (as they were increasingly unable to represent short term fluctuations). For FNN networks with larger learning rates which could otherwise adapt quickly, the increased focus on long term features caused P&L performance degradation. For FNN networks with smaller learning rates, poorly able to adapt quickly to new information either way, there was a benefit from SAE features with an increasing representation of long term trends. The relationship is emphasised dramatically in the 10-feature SAEs, to the point that lower learning rates can be more effective in generating OOS P&L. The P&L performance suggests that the 10-feature SAEs overfit to long term IS features, and became pathological for short term adaptation OOS.

The most noteworthy results were the 5-feature SAEs, where performance was often on par or better than the higher feature sizes, or no SAE at all, as seen in Figure 3. It is possible that the small encoding layer acts as a form of regularization, forcing the SAE to learn more consistently generalisable features. The performance of the 5-feature networks, with an 83% reduction in input data, is clear evidence of the efficacy and potential of feature selection in financial times series.

The effect of data horizon aggregations is as expected: short term horizons (i.e. [1,5,10]) outperformed in configurations with more SAE features and higher learning rates; longer term horizons (i.e. [10, 20, 60]) outperformed in low learning rate and low feature configurations. The differentiation between these groups is seen more robustly in Section V-F where data horizon aggregations are determined to be the primary clustering attribute for trade correlations. Strategies focusing on short term predictive strategies (aggregations of [1, 5, 10]) had higher variance in returns than the longer data horizon strategies, though also the highest highest P&L and Sharpe ratios. This again shows the benefits in focusing on recent cross-sectional data in financial markets. The differentiation between the groups is discussed more in [18].

C. Money Management Strategy Results

The benchmark is an upper bound on performance, representing MMS returns based on perfect knowledge of future prices. The benchmark full return rate is 2.4% with trading costs, over a period of 1555 trading days. So while the strategies’ proximity to the benchmark do represent a framework success, they are not necessarily representative of a feasible market solution. Ultimately, this enforces the notion that the MMS implementation is of exceeding importance in a live trading process, and predictive accuracy is only able to achieve so much.

Figure 4 shows the distributions of OOS P&L with trading costs being accounted for. There were a significant number of configurations within 20%-30% of the benchmark. The trials with 0 P&L are networks which suffered from either exploding or vanishing gradients, and were not able to make sufficient predictions.

1Input data was 10 assets with 3 horizon aggregations each, resulting in an input size of 30 at each timestep.
OOS P&L By Feature Size and OGD Learning Rate

Fig. 3. This figure shows that the lower learning rates (0.005, 0.01) performed best with strategies using long term trend pricing. The 10 feature encoding appeared to optimise specifically for this perspective. The optimisation caused outperformance at the lower learning rates and detrimental performance at higher learning rates (which perform best with short term fluctuation strategies). The 15 to 25 horizon encodings showed a better association to the short term strategies. Here higher encodings and higher learning rates offer the best performance. The 5 feature encoding offered consistent performance across learning rates and shows the learning of generalisable features.

MMS OOS P&L Distributions, with Costs Applied

Fig. 4. The distributions of all OOS P&L values, with the benchmark P&L indicated in orange, show an encouraging view of the results. There is a significant negative skew, with a proportionally small number of strategies resulting in negative returns, even with capital and trading costs applied. There were a large proportion of strategies near the OOS upper bound, which is within 28% of the benchmark.

D. Probability of Backtest Overfitting

1) Applying PBO in Mechanistic Machine Learning: While the methodology is a model free approach to assessing overfitting, the application in a machine learning context is novel and has dynamics worth considering. The use of offline batch learning parameters, online learning parameters and adaptive network weights make the concept of model parameters less distinct. If a model performs well OOS due to effective learning, this can be due to the model’s strength rather than overfitting.

It is noted that the logit metric, which the CSCV method relies on, has its basis in an ordinal ranking; indicating whether the best strategy in the IS set is higher than the median in the OOS set. This means that poor performing configurations can artificially bolster an ordinal position past the median point and so bias PBO results. An honest, wide exploration of the parameter solution space in a mechanistic machine learning framework is likely to result in “poor” configurations being tested (as visible in the ‘0’ P&L configurations in Figure 4). As a result, the methodology shifts the onus onto the researcher in both handling and reporting these dynamics responsibly.

Further to this, the parameter space search methodology (Section IV-B) may also result in a lower likelihood of PBO due to the way of combining parameters across IS and OOS stages. By way of example, any configuration which performs well IS will have all possible OOS parameters tested in combination with it. While some of these combinations may result in poor performance, there will always be a combination of the best IS and best OOS parameter choices. This makes it unlikely that the best configurations will be past the median point for the logit calculation, resulting in a systematically low PBO regardless of how many configurations are attempted.

Lastly, the CSCV algorithm requires a parameter choice of how many windows to split the data into. While not inherently problematic, this choice can have a significant impact on results which is not visible in the reported PBO value. We discuss this further in [18].

2) PBO Results: We ran the CSCV algorithm on the majority of the configurations tested, which resulted in a final PBO value of 1.7%. A subset of networks were excluded on accounts of ‘null’ predictions, resulting in a sample size of 21653 (out of a total of 22248 configurations). The CSCV algorithm was run with a split value of 16. There were 15 years of data, making 16 a reasonable choice as the split parameter (which needs to be even). Ideally, the splits would represent shorter periods, but the exponential increase in computational time made this impractical. The full logit distribution can be seen below in Figure 5.

We found interesting dynamics around the calculation of PBO, and the configurations contributing to the figure. The configuration process went through 2 primary phases: an extremely broad combinatorial grid search, consisting of 20736 configurations; and a second much narrower search of 1512 configurations. Assessing only the configurations from the second phase resulted in a PBO score of 6.3%, which was significantly higher than the overall PBO score. The effect here highlights important aspects of the PBO calculation. The PBO score was much higher for the configurations which were picked more specifically after having already seen a large number of results, which is correctly indicative of increased likelihood to overfit. However, the PBO score is not monotonically increasing with N, as one would expect. This is counterintuitive and is in line with the concerns regarding the effects of increasing configuration sample size.
Logit Distribution for All Configurations

Fig. 5. The CSCV logit distribution for the 21,653 configurations run, with a calculated PBO of 1.7%. The strong negative skew is indicative of IS and OOS strategy returns being closely matched in rank and results in a low PBO score. This is a favourable assessment for the efficacy of the full framework presented here and shows that training was able to occur without much risk of backtest overfitting.

E. Optimal Number of Clusters

The ONC algorithm produced three clusters: one consisting mostly of the negative Sharpe ratio configurations, and two containing the remaining configurations partitioned by their data horizon aggregations configurations. The distributions for all three clusters’ Sharpe ratios can be see below in Figure 6.

If we consider the two primary clusters, we see that Cluster One contained all the trials with horizon aggregations of [5, 20, 60] and [10, 20, 60], and Cluster Two contained all trials with horizon aggregations of [1, 5, 10]. The nature of the experimentation process, with the combinatorial parameter space exploration (as detailed in Section IV-B), is such that other parameters were mostly evenly split across the 2 clusters (e.g. OGD learning rate, network sizes, initializations and so on).

The clusters here indicate that the networks adapted to at least two different general strategies for predicting prices: one which was more influenced by the long term fluctuations, and the second was more influenced by the short term fluctuations. The results presented in Section IV-B are then indicative of the networks ability to execute these overarching strategies effectively.

The best Sharpe ratio value (with trading costs applied) was 0.64 and part of Cluster Two, with the [1, 5, 10] price fluctuation horizon aggregations. The distributions seen in Figure 6 indicate that at a general level, Cluster One has more consistent performance, Cluster Two on the other hand has higher variance, with more strategies at both the lower and higher range of Sharpe ratios. The lack of further clusterings was probed manually to find that the variance from further subclustering lead to a worse cost function score.

F. DSR and PSR Results

Using the clusters produced by the ONC Algorithm (Section IV-B), the DSR could be determined. The aggregate cluster time series returns were calculated and annualized to allow their variance estimates to be used to calculate $SR^*$ (the maximum expected observed Sharpe ratio due to variance under the null hypothesis of $H_0 : SR = 0$). Using $SR^*$ as the benchmark, the PSR calculation ($\hat{PSR}(SR^*)$) can then be used to determine if the observed $\hat{SR}$ is a false positive or not. This gives us the DSR as a confidence for observing a positive best $SR$. The benchmark $SR^*$ calculated was 0.211245, and the best $\hat{SR}$ observed was 0.642632, leading to a $\hat{PSR}(SR^*)$ of 1.0, thus indicating that the trials certainly contain a strategy which has a positive SR rate. This seems a reasonable conclusion, considering the SR distributions in Figure 6.

VI. Conclusions

Mechanistic machine learning approaches to financial market data hold some promise for enhancing the performance of low-frequency quantitative trading. To investigate this potential we provide a novel framework that we show to be effective in both training, and in validation. The framework is configurable and based on decoupled modules and uses several well understood techniques: deep-learning neural networks for stock price fluctuation prediction, stacked autoencoders for the purpose of feature selection, both CSCV & PBO to assess the returns from MMS, and the likelihood that backtest overfitting has taken place, and DSR in order to assess the likelihood of a positive Sharpe ratio having been observed.

While machine learning models are expected to excel in big data environments, in financial markets there is in fact a lack of data with relevant information and signal, both for training and more so for validation. We show that IS training on historical data had a limited benefit. This is not a surprising empirical insight and was confirmed by the negligible impact of increasing training time, as well as the small impact of large reductions in the training data sizes. Increased performance for IS data was not significantly linked to OOS performance. This emphasises the idea that the changing dynamics of financial
markets over time need careful attention. Learning optimizations for IS training, such as regularization and learning rate schedules, were shown to have IS benefits, but little impact on OOS performance [18]. This gives weight to the importance of online learning methods for financial applications, and in turn highlights the importance of network initialization. The results showed better performance for the He-adjusted initialization and poor performance in RBM based pre-training.

The primary determinants of OOS P&L were shown to be those which affect the online learning data and the model’s ability to adapt to this. We found SAE encoding layer sizes influenced the nature of features learnt, with smaller encodings generally leading to learning of longer term features. This relationship continued from layer sizes 25 to 10, with increasing effect. The results from the 10-feature SAEs suggest the SAEs were overfit to long term IS features and pathological for OOS adaptation. The relationship changed at 5-feature SAEs, which learnt far more generalisable features. The 5-feature SAEs had very competitive performance and show that feature selection in financial time series is both possible and beneficial despite the complexity present. Predictive strategies focusing on long term changes were present in configurations with longer data horizons, less features and lower learning rates (slow adaptation). Short term strategies presented with shorter data horizons, more features and larger learning rates (quick adaptation). The data horizon was the primary separating attribute in the ONC clusters, emphasising these groupings. The short term strategies had higher variance, but also the highest returns. This again highlights both the increased value in recent information in financial markets, as well as the difficulty in using it due to the amount of noise present.

The most challenging aspect of a mechanistic approach to learning is avoiding backtest overfitting (see Section I-B). Probing and validating of the generalisation error was done using the PBO methodology in conjunction with DSR [25]. The results discussed in Section V show a low likelihood of the models having overfit. The CSCV and PBO methodologies suggest that they are able to add-value to our novel implementation of machine learning models, while providing a robust assessment of results. The results were further validated using the ONC and DSR algorithms to detect positive Sharpe ratios.

The phenomenological view of financial markets based on our experimental results suggest a very limited benefit to training on long term historical financial time series. A cross sectional view of the data has far more weight in delivering OOS returns; this is noteworthy in the context of neural networks. Based on our simulation work we speculate that money management strategies can be more important determinants of OOS profitability relative to signal generation and should also be learnt.

ACKNOWLEDGMENT
The authors would like to thank Sebnem Er, Patrick Chang and Turgay Celik for valuable comments on the project.

REFERENCES
[1] J. Murphy, “Technical analysis of financial markets,” 01 1999.
[2] J. H. Andrew W. Lo, Heretics of Finance. Bloomberg Press, 2009.
[3] B. Arthur, “Complexity in economics and financial markets,” Complexity, vol. 1, pp. 20–25, 10 1995.
[4] D. Wilcox and T. Gebbie, “Hierarchical causality in financial economics,” Available at SSRN: https://ssrn.com/abstract=2544327, 2014.
[5] K. Hornik, “Multilayer feed-forward networks are universal approximators,” Neural Networks, vol. 2, pp. 359–366, 1989.
[6] F. Loonat and T. Gebbie, “Learning zero-cost portfolio selection with pattern matching,” PLOS ONE, vol. 13, 05 2016.
[7] N. Murphy and T. Gebbie, “Learning the population dynamics of technical trading strategies,” arXiv:1903.02228 [q-fin.CP]. 03 2019.
[8] J. Ioannidis, “Why most published research findings are false,” CHANCE, vol. 32, pp. 4–13, 01 2019.
[9] D. Bailey, J. Borwein, M. Lpez de Prado, and Q. J. Zhu, “The probability of backtest overfitting,” Journal of Computational Finance, vol. 20, pp. 39–69, 4 2017.
[10] R. Mclean and J. Pontiff, “Does academic research destroy stock return predictability?,” The Journal of Finance, vol. 71, 05 2013.
[11] F. Schortheide and K. Wolpin, “On the use of holdout samples for model selection,” American Economic Review, vol. 102, 05 2012.
[12] S. M. Weiss and C. A. Kulikowski, Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991.
[13] D. Hawkins, “The problem of overfitting,” Journal of chemical information and computer sciences, vol. 44, pp. 1–12, 05 2004.
[14] G. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” Neural computation, vol. 18, pp. 1527–54, 08 2006.
[15] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” Adv. Neural Inf. Process. Syst., vol. 19, pp. 153–160, 01 2007.
[16] M. Ranzato, C. Poultney, S. Chopra, and Y. Lecun, “Efficient learning of sparse representations with an energy-based model,” 01 2006.
[17] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science (New York, N.Y.), vol. 313, pp. 504–7, 08 2006.
[18] J. Da Costa, “Online non-linear prediction of financial time series patterns,” Master’s thesis, University of Cape Town, Cape Town, ZA, Dec 2020.
[19] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, pp. 1798–1828, 08 2013.
[20] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, “Why does unsupervised pre-training help deep learning?,” Journal of Machine Learning Research, vol. 11, pp. 625–660, 02 2010.
[21] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” Journal of Machine Learning Research - Proceedings Track, vol. 9, pp. 249–256, 01 2010.
[22] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science (New York, N.Y.), vol. 313, pp. 504–7, 08 2006.
[23] J. Da Costa, “Online non-linear prediction of financial time series patterns,” Master’s thesis, University of Cape Town, Cape Town, ZA, Dec 2020.
[24] D. Bailey and M. Lopez de Prado, “The sharpe ratio efficient frontier,” The Journal of Risk, vol. 15, pp. 3–44, 12 2012.
[25] D. Bailey and M. Lopez de Prado, “The deflated sharpe ratio: Correcting for selection bias, backtest overfitting, and non-normality,” The Journal of Portfolio Management, vol. 40, pp. 94–107, 09 2014.
[26] M. Lopez de Prado and M. Lewis, “Detection of false investment strategies using unsupervised learning methods,” Quantitative Finance, vol. 19, pp. 1–11, 07 2019.
[27] J. Da Costa, “Jse price relative dataset from 2003-2018.” https://zivahub.uct.ac.za/articles/dataset/JSE_Top40_Closing_Price_Relative_Data_for_2003-2018/1897628]. 2020. Accessed: 2020-06-30.
[28] J. Da Costa, “Julia libraries for online non-linear prediction of financial time series patterns.” https://github.com/joel11/Masters. 2020. Accessed: 2020-06-30.
[29] Y. Lecun, L. Bottou, G. Orr, and K.-R. Miller, Efficient BackProp, vol. 1524, pp. 546–546, 01 1998.
[30] L. Bottou and N. Murata, Stochastic Approximations and Efficient Learning, The Handbook of Brain Theory and Neural Networks. The MIT Press, 2 ed., 2019.
[31] L. Bottou and Y. Lecun, “Large scale online learning., Advances in Neural Information Processing Systems 16., 03 2004.
Fig. 7. Overall Process Flow: The flow diagram here offers a visual representation of the framework from data processing through to training and price prediction processing, as detailed in Section II. The PBO & DSR validation steps are not shown here, though use the IS and OOS Price Predictions from step 8 as inputs.