Reading the Tea Leaves: A Neural Network Perspective on Technical Trading

Sid Ghoshal
Department of Engineering Science
Oxford-Man Institute of Quantitative Finance
University of Oxford
sghoshal@robots.ox.ac.uk

Stephen Roberts
Department of Engineering Science
Oxford-Man Institute of Quantitative Finance
University of Oxford
sjrob@robots.ox.ac.uk

ABSTRACT

Technical analysis in finance is the discipline of graphically analysing the price history of assets, on the premise that specific geometric shapes in charts reliably foreshadow future movements. Though widely used in practice, both the recognition of its patterns and interpretation of their meaning remain a highly subjective form of 'domain knowledge'. We investigate the predictive value of these visual patterns, applying machine learning and signal processing techniques to 22 years of US equity data. By reframing technical analysis as a poorly specified, arbitrarily preset feature-extractive layer in a deep neural network, we show that better convolution filters can be learned directly from the data, and provide visual representations of the features being identified.

KEYWORDS

Technical analysis, machine learning, neural networks.

1 INTRODUCTION

In financial media, extensive attention is given to the study of charts and visual patterns. Known as *technical analysis* or *chartism*, this form of financial analysis relies solely on historical price and volume data to produce forecasts, on the assumption that specific graphical patterns hold predictive information for future asset price fluctuations (Blume et al, 1994). Early research into genetic algorithms devised purely from technical data showed promising results, sustaining the view that there could be substance to the practice (Neely et al, 1997; Allen and Karjailainen, 1999). Research in finance has typically restricted itself to the time series of closing prices and the visuals emerging from line charts (Lo et al, 2000), relying on kernel regression to smooth out the price process and enable pattern recognition.

An equally common visual representation of price history in finance is the candlestick. Candlesticks encode opening price, closing price, maximum price and minimum price over a discrete time interval, visually represented by a vertical bar with lines extending on either end. Much as with line charts, technical analysts believe that specific sequences of candlesticks reliably foreshadow impending price movements. A wide array of such patterns are commonly watched for (Taylor and Allen, 1992), each with their own pictogram and associated colourful name ('inverted hammer', 'abandoned baby', etc).

Little effort has gone into the systematic evaluation of this richer but still purely technical dataset, and there is a very real possibility that candlestick chartism represents a form of modern day financial astrology, with just as little predictive prowess as horoscopes yet somehow held in higher regard. The human propensity towards *confirmation bias* is partly to blame: finance's low signal-to-noise ratio makes it far too easy to imagine patterns where there are none.

After defining a format for cross-correlating time series data with chartist filters (Section 2.1-2.3), we undertake a comprehensive statistical assessment of the predictive prowess of the most common candlestick patterns (Section 2.4). We draw on a modern intuition for pattern recognition in vision and language (Bengio, 2009): we re-frame candlestick patterns as a form of *feature engineering* intended by chartists to extract salient features, facilitating the classification of future returns with higher fidelity than the raw price process would otherwise allow. Feeding candlestick data through a neural network involving separate filters for each technical pattern, we classify next-day returns with the filters implied by chartist doctrine (Section 3.1-3.2) and set this cross-correlational approach as a baseline to improve upon (Romaszko, 2015). We then compare the model's accuracy when the filters are not preset but instead learned by a convolutional neural network (CNN) during its training phase (Section 3.3), and evaluate its performance against recurrent neural network (RNN) architectures, considered the state of the art in time series analysis (Section 3.4). Finally we assess the significance of our findings statically (Section 3.5) and through time (Section 3.6), and benchmark deep learning in finance against alternative methods (Section 3.7).

Our results find little evidence to support the practice of chartism. We agree with Lo et al (2000) that the distribution of future returns conditioned on observing technical patterns diverges significantly from the unconditional distribution, but upon close inspection the resulting classifier barely outperforms guesswork. By contrast, filters learned and tested on 22 years of S&P500 price data in the same CNN architecture yield modest gains in classification accuracy.

2 EVALUATING TECHNICAL ANALYSIS

2.1 Definition of Candlestick Data

Both the financial time series data and the candlestick technical filters used by chartists take the same form. Asset price data for a discrete time interval is represented by four features: the opening price (price at the start of the interval), closing price (price at the end of the interval), high price (maximum over the interval) and low price (minimum over the interval). The candlestick visually encodes this information (Fig. 1): the bar's extremities denote the open and close prices, and the lines protruding from the bar (the candle's 'wicks' or shadow) denote the extrema over the interval. The colour of the bar determines the relative ordering of the open

and close prices: a white bar denotes a positive return over the interval (close price > open price) and a black or shaded bar denotes a negative return (close price < open price).

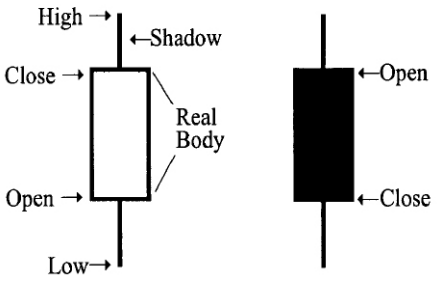

Figure 1: Candlestick representation of financial time series data.

We can therefore summarise the candlestick representation of a financial time series of length n timesteps as a $4 \times n$ price signal matrix F capturing its four features. Throughout this paper we rely on daily market data, but the methods can be extended to high-frequency pattern recognition using tick data.

2.2 Definitions of Technical Patterns

We focus on eight of the most popular candlestick patterns cited by practitioners of technical analysis: the abandoned baby (2 variants), evening star, morning star, three black crows, three white soldiers, three inside down and three inside up. Fig. 2 provides both the visual template associated with each pattern, as well as the future price direction it is meant to presage. As before, we summarise a technical pattern P of length m timesteps as a $4 \times m$ matrix T_P, standardised for comparability to have zero mean and unit variance.

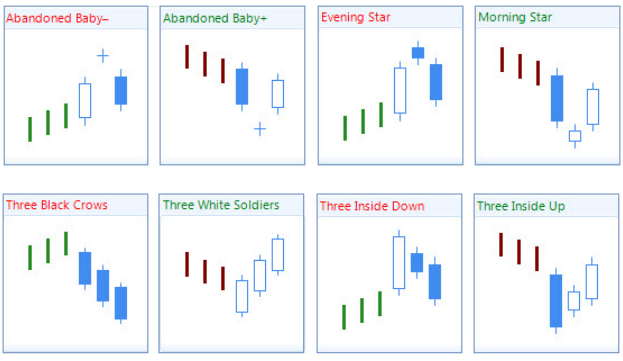

Figure 2: Eight technical patterns and the future direction they predict (red for negative returns, green for positive returns).

2.3 Identification by Template Matching

Matrix representations for both the template T_P and equal-length, standardised rolling windows F_n of the full price signal F at timestep n can be cross-correlated together to generate a time series S_P measuring the degree of similarity between the price signal and the filter. For a given pattern P, at each timestep n:

$$S_{P,n} = \left\langle \frac{T_P}{\|T_P\|}, \frac{F_n}{\|F_n\|}, \right\rangle \tag{1}$$

where $\langle \cdot, \cdot \rangle$ is the inner product of the two matrices and $\| \cdot \|$ is the L^2 norm.

Our algorithm extracts the top quantile (in our study, decile and centile) of similarity scores S_P as pattern matches and produces a distribution of next-day returns conditional on matching pattern P.

2.4 Evaluating Technical Analysis

We run several diagnostics to assess separately the informativeness and predictive prowess of each technical pattern.

2.4.1 Empirical Data. Throughout our work, we use technical (i.e. open, close, high and low price) data from the S&P500 stock market index constituents for the period Jan 1994 - Dec 2015, corresponding to n = 2,182,516 entries of financial data in the price signal F. This dataset covers a representative cross-section of US companies across a wide timeframe suitable for learning the patterns, if any, of both expansionary and recessionary periods in the stock market.

2.4.2 Informativeness. We begin by comparing the top quantiles of the conditional returns with their unconditional counterparts, with the view that conditioning on informative patterns should yield significantly different distributions. Denoting by $\{R_P{}_{t=1}^{n_1}\}$ the subset of returns conditioned on matching pattern P and $\{R_{t=1}^{n_2}\}$ the full set of unconditional returns, we compute their empirical cumulative distribution functions $F_1(z)$ and $F_2(z)$. The two-sample Kolmogorov-Smirnov (K-S) test evaluates the null hypothesis that the distributions generating both samples have identical cdfs, by computing the K-S statistic:

$$\gamma = \left(\frac{n_1 n_2}{n_1 + n_2} \right)^{1/2} \sup_{-\infty < z < \infty} |F_1(z) - F_2(z)| \tag{2}$$

The limiting distribution of γ provides percentile thresholds above which we reject the null hypothesis. When this occurs, we infer that conditioning on the pattern does materially alter the future returns distribution. As an example of this approach, we provide the empirical cdfs of both unconditional returns and returns conditioned on the pattern: 'Three Black Crows' (Fig. 3).

2.4.3 Predictive Prowess. Whilst these patterns may bear some information, it does not follow that their information is actionable, or even aligns with the expectations prescribed by technical analysis. Notched boxplots of both unconditional returns and returns conditioned on each of the filters (Fig. 4) allow us to gauge whether the pattern's occurrence does in fact yield significant returns in the intended direction.

A closer examination suggests that conditioning on 7 of the 8 patterns produces no significant alteration in the median of next-day

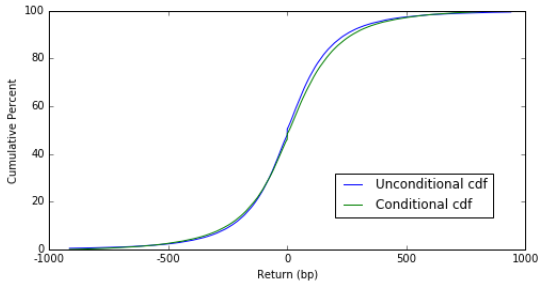

Figure 3: Empirical cumulative distribution functions of unconditional returns and returns conditioned on matching the pattern 'Three Black Crows', where a match is deemed to have occurred when the similar score S_P is in its top decile.

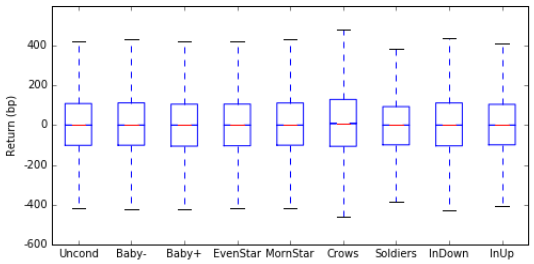

Figure 4: Notched boxplots of the distributions of returns in basis points (one hundredth of a percent), conditional on observing each of the technical patterns (similarity score S_P in its top decile). At a glance, none of the conditional distribution medians diverge substantively from the unconditional baseline, and the distributions' standard deviations dwarf their medians by two orders of magnitude.

returns distributions (Fig. 5). Only 'Three Black Crows' produces a conditional distribution for which the 95% confidence interval of the median (denoted by the notch) bears no overlap with its unconditional counterpart. But even then, the deviation is actually positive, the polar opposite of what chartist doctrine would imply.

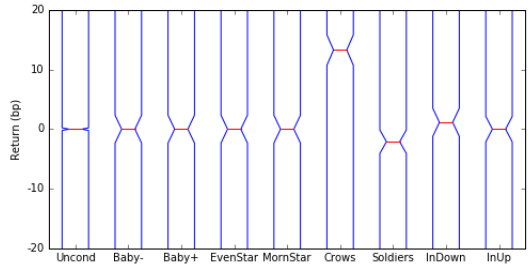

Figure 5: Close-up of boxplot notches for the distributions of returns in basis points (one hundredth of a percent), conditional on observing each of the technical patterns (similarity score S_P in its top decile). Almost all of the patterns exhibit notches that overlap with the unconditional distribution's, implying that the distribution medians are not meaningfully changed by conditioning. Only 'Three Black Crows' seems to be significant - as a harbinger of better times, despite its name.

2.4.4 Results. Table 1 reports the empirical results of the K-S goodness-of-fit tests and top decile conditional distribution summary statistics, using daily stock data from the S&P500. Though several of the patterns do indeed bear information altering the distribution of future returns, their occurrence is neither a reliable predictor of price movements (high standard deviation relative to the mean) nor even, in many instances, an accurate classifier of direction. The results found when the metric for pattern recognition is more stringent (top centile of similarity score S_P) are reported in Table 2, and come to the same conclusion. Conceptually, the notion of using filters in financial data to extract informative feature maps may bear merit - but the chartist filter layer is demonstrably an improper specification.

Table 1: Summary statistics for the next-day return distributions conditioned on matching technical patterns. A match on pattern P is deemed to have occurred when the cross-correlational similarity score S_P is in its top decile. K-S statistics above 1.95 are significant at the 0.001 level. Mean return μ for each pattern is expressed as a difference from the unconditional baseline. The incremental mean returns are dwarfed by their standard deviation, and do not even always move in the direction prescribed by chartism.

Pattern	γ	$\mu(bp)$	$\sigma(bp)$
Unconditional		4.26	229.40
Abandoned Baby-	4.03	-4.24	224.60
Abandoned Baby+	2.88	+4.87	227.16
Evening Star	2.53	-2.32	223.24
Morning Star	2.79	+4.86	228.15
Three Black Crows	14.28	+5.62	265.14
Three White Soldiers	12.97	-7.98	208.90
Three Inside Down	2.91	+0.45	231.62
Three Inside Up	3.27	+0.71	220.71

Table 2: Summary statistics for the next-day return distributions conditioned on matching technical patterns more stringently. A match on pattern P is deemed to have occurred when the cross-correlational similarity score S_P is in its top centile.

Pattern	γ	$\mu(bp)$	$\sigma(bp)$
Unconditional		4.26	229.40
Abandoned Baby-	3.29	-4.04	232.45
Abandoned Baby+	1.27	+2.94	232.28
Evening Star	2.89	-0.27	231.76
Morning Star	1.80	+2.59	231.89
Three Black Crows	6.85	+13.09	229.40
Three White Soldiers	6.30	-11.77	203.26
Three Inside Down	1.63	+2.72	233.12
Three Inside Up	2.50	+0.13	220.75

3 FEATURE ENGINEERING IN FINANCE

The concept of searching for informative intermediate feature maps in classification problems has seen widespread success in domains ranging from acoustic signal processing (Hinton et al, 2012) to computer vision (Krizhevsky et al, 2012). Where technical analysis uses filters that are arbitrarily pictographic in nature, we propose to learn layers for feature extraction from data.

We begin by splitting our S&P500 time series data into training and test sets corresponding to stock prices from 1994-2004 and 2005-2015 respectively.[1] We evaluate the performance of passing the raw data both with and without chartist filters, and subsequently measure the incremental gain from learning optimal feature maps by convolution. The findings are then benchmarked against widely recognised approaches to time series forecasting, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, nearest neighbours classifiers and support vector machines (SVM).

3.1 Raw Data Neural Network

To address issues of scale and stationarity, we process the original 4 $\times$ n price signal matrix F into a new 12 $\times$ n price signal matrix F^*,[2] where each column is a standardised encoding of 3 days of price data. We pass F^* through a multilayer perceptron (MLP) involving fully-connected hidden layers. Preliminary cross-validation experiments with financial time series determined the network topology required for the model to learn from its training data. Insufficient height (neurons per hidden layer) and depth (number of hidden layers) led to models incapable of learning their training data (cross-validation accuracy plateau'ing at 52.9% in-sample for a single-layer neural network with 100 neurons). We settled on 2 fully-connected layers of 1000 neurons with ReLU activation functions, followed by a softmax output layer to classify positive and negative returns. Even so, the raw data does not lend itself well to classification, generalising poorly (out-of-sample accuracy of 50.1% after 1000 epochs, Table 3). Regularisation was achieved via the inclusion of dropout (Srivastava, 2014) in the dense layers of the network, limiting the model's propensity towards excessive co-adaptation across layers.

3.2 Technically-Filtered Neural Network

Reframing technical patterns as pre-learned cross-correlational filters, we now standardise and stack the 8 pattern matrices T_P, each of dimension 4×3 in our study, along the depth dimension, producing a 4×3×8 tensor T whose inner product with standardised windows of F yields a new $8 \times n$ input matrix F_T.[3]

$$F_T = \left\langle T, F \right\rangle \tag{3}$$

[1] Our classes are defined as 'negative return' and 'strictly positive return'. We therefore anticipate some class imbalance in the dataset, as zero return days occur (albeit infrequently) in assets with low denomination. The training set class imbalance is negatively skewed (47.6% strictly positive return days, 52.4% negative return days), against a positively skewed test set (50.9% strictly positive return days, 49.1% negative return days). Models that learn only the training set's bias may if anything perform worse than random chance, as is the case with several of our benchmarks (Section 3.7).
[2] The price signal F is zero-padded along the temporal axis to preserve length n for F^*.
[3] As before, the price signal F is zero-padded along the temporal axis during cross-correlation, yielding an unchanged length n for F_T.

Table 3: Accuracy obtained after training the model using raw data through a set number of epochs. Adding further epochs does not help it to generalise any better.

Epochs	In Sample (%)	Out-of-Sample (%)
1	51.6	49.8
5	52.7	50.1
10	52.9	50.0
50	53.3	50.2
100	53.5	50.2
250	54.8	50.1
500	56.2	50.1
1000	58.0	50.1

This new input is the result of cross-correlating the raw price signal F with the technical analysis filter tensor T, and can be interpreted as the feature map generated by technical analysis. We now use F_T as the input to the same MLP as before and look for improvements in model forecasts. The results we find are consistent with Section 2: using technical analysis for feature extraction hinders the classifier, slightly degrading model performance (out-of-sample accuracy of 49.5% after 1000 epochs, Table 4).

Table 4: Accuracy obtained after training the model using technical analysis filters through a set number of epochs. The technical analysis filters produce feature maps with even less discernible structure.

Epochs	In Sample (%)	Out-of-Sample (%)
1	50.4	49.5
5	52.2	49.5
10	52.4	49.3
50	52.4	49.3
100	52.5	49.5
250	52.5	49.4
500	52.7	49.5
1000	52.8	49.5

3.3 Convolutional Neural Network

We now deepen the neural network by adding a single convolutional layer with 8 filters to our earlier MLP (architecture detailed in Table 5). The CNN finds much greater structure in its training data than the technically-filtered MLP could, and generalises slightly better than both earlier iterations (out-of-sample accuracy of 50.8% after 1000 epochs, Table 6). Crucially, unlike the earlier models, the CNN's in-sample and out-of-sample accuracy rise together, suggesting that the feature representation being learned may have the potential to generalise successfully. That being said, even deep learning yields only very marginal gains in predictive prowess over pure chance, calling into question whether future price inference from past price - a core tenet of technical analysis - can be achieved at all.

The convolution filters learned by the network provide a basis for feature extraction. In particular, the convolutional layer's filters define patches whose convolution with zero-padded raw input data minimised the model's in-sample categorical cross-entropy. We

Table 5: Details of the CNN architecture. The number of filters in the convolution layer was deliberately kept low (8) and their dimensions (4×3) match the technical patterns used in Section 3.2, to enable like-for-like comparability with the technical filter approach.

#	Layer	Units	Activation Function	Dropout	Filter Shape	Outgoing dimensions
0	Input	-	-	-	-	(input) $[4 \times 3]$
1	Convolutional	8	ReLU	-	$[4 \times 3]$	$[8 \times 12]$
3	Dense	1000	ReLU	0.5	-	[1000]
4	Dense	1000	ReLU	0.5	-	[1000]
5	Softmax	-	-	-	(output, 2 classes) [2]	

Table 6: Accuracy obtained after training a deep neural network with a single convolution layer through a set number of epochs.

Epochs	In Sample (%)	Out-of-Sample (%)
1	51.6	49.2
5	52.7	50.0
10	52.9	50.3
50	53.3	50.4
100	53.7	50.5
250	54.8	50.5
500	56.1	50.7
1000	57.3	50.8

produce a mosaic of these filters as Hinton diagrams (Fig. 6) and visualise them in the language of technical analysis as candlestick patterns (Fig. 7 and 8) by reversing the convolutional filters, turning them into cross-correlational templates whose occurrence is informative for financial time series forecasting. Unlike technical patterns however, these templates have no set meaning: the purpose of individual neurons in a convolutional layer is not readily interpretable.

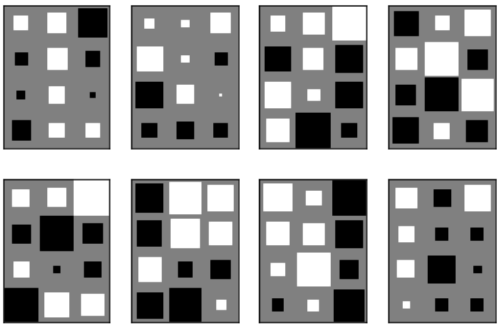

Figure 6: Weight-space visualisation as Hinton diagrams for the 8 cross-correlational filters learned from the first layer of the CNN. These cross-correlational templates were obtained by reversing the filters of the convolutional layer.

3.4 Recurrent Neural Network

Deep learning for time series analysis has typically relied on recurrent architectures capable of learning temporal relations in the data. In particular, Long Short-Term Memory (LSTM) networks have achieved prominence for their ability to memorise patterns across significant spans of time (Hochreiter and Schmidhuber, 1997)

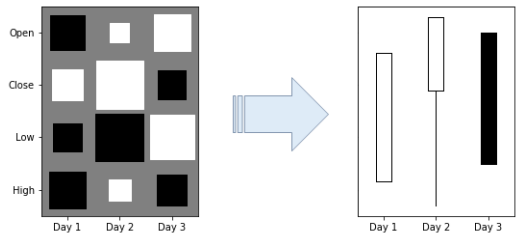

Figure 7: Hinton diagram of the fourth cross-correlational filter learned in the first layer of the CNN. The relative values of the standardised open, close, low and high for each column in the filter define, in a chartist sense, a specific candlestick sequence (or patch thereof, in instances where the filter's open or close is incompatible with the high-low range) which the neural network extracted as informative for time series forecasting.

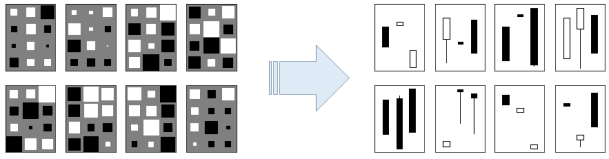

Figure 8: Candlestick pattern approximations of the cross-correlational filter mosaic.

by addressing the vanishing gradient problem. A thorough RNN architecture search (Jozefowicz et al, 2015) identified a small but persistent gap in performance between LSTMs and the recently-introduced Gated Recurrent Unit (GRU, Chung et al, 2014) on a range of synthetic and real-world datasets. To benchmark the effectiveness of our approach against the state of the art, we feed the processed price signal matrix F^* through recurrent neural networks built using a preliminary layer of 8 RNN units (LSTM and GRU in separate experiments), followed by 2 dense layers of 1000 neurons with dropout as before. RNNs perform better than the MLPs on raw and technically filtered data (out-of-sample accuracies of 50.6% and 50.7% after 1000 epochs for LSTM and GRU respectively, Tables 7 and 8), but fall short of matching the performance of the convolutional approach. The GRU architecture marginally outperforms the LSTM model on our financial data, with the further benefit of greater computational efficiency (17% faster to train).

3.5 Significance of Model Results

To investigate whether the predictive performance of the 3 neural network classifiers is statistically significant, we derive the area under the curve (AUC) of each model's receiver operating characteristic curve (ROC), and exploit an equivalence between the AUC

Table 7: Accuracy obtained after training a deep neural network with a single LSTM layer through a set number of epochs.

Epochs	In Sample (%)	Out-of-Sample (%)
1	51.4	49.8
5	52.5	50.0
10	52.7	50.0
50	53.0	49.8
100	53.1	50.1
250	53.4	50.2
500	54.3	50.4
1000	54.9	50.6

Table 8: Accuracy obtained after training a deep neural network with a single GRU layer through a set number of epochs.

Epochs	In Sample (%)	Out-of-Sample (%)
1	51.6	49.2
5	52.7	50.0
10	52.7	49.7
50	53.1	50.4
100	53.3	50.3
250	54.0	50.4
500	55.4	50.6
1000	56.5	50.7

and Mann-Whitney-Wilcoxon test statistic U (Mason and Graham, 2002):

$$AUC = \frac{U}{n_P n_N} \tag{4}$$

where n_P and n_N are the number of positive and negative returns in the test set, respectively. In our binary classification setting, the Mann-Whitney-Wilcoxon test evaluates the null hypothesis that a randomly selected value from one sample (e.g., the subset of test data classified as positive next-day returns) is equally likely to be less than or greater than a randomly selected value from the complement sample (the remaining test data, classified as negative next-day returns). Informally, we are testing the null hypothesis that our models have classified at random. U is approximately Gaussian for our sample size, so we compute each model's standardised Z-score and look for extreme values that would violate this null hypothesis.

$$Z = \frac{U - \mu_U}{\sigma_U} \tag{5}$$

where:

$$\mu_U = \frac{n_P n_N}{2} \tag{6}$$

and

$$\sigma_U = \sqrt{\frac{n_P n_N (n_P + n_N + 1)}{12}} \tag{7}$$

Table 9 provides the AUC, Z-statistic and significance of each model, where significance measures the area of the distribution below Z. We disregard significance for negative Z scores (as is the case for the technically-filtered neural network) as they imply classifiers that performed (significantly) worse than random chance. Learning neural network filter specifications via convolution yields a significant boost to predictive prowess over the baseline model of Section 3.1 and technically-filtered variant of Section 3.2, and also compares favourably with the recurrent architectures of Section 3.4.

Table 9: AUC, Z-statistic and significance level for the neural network classifiers.

Model	AUC (%)	Z	Significance
NN-Raw	50.2	2.267	0.9881
NN-Technicals	49.4	−10.773	–
CNN	51.0	18.677	> 0.9999
RNN-LSTM	50.8	14.534	> 0.9999
RNN-GRU	50.9	16.907	> 0.9999

3.6 Interpreting Accuracy over Time

In this section, we investigate potential failure modes in our best-performing classifiers: the CNN and RNN-GRU. We evaluate model accuracy over 3-month rolling windows of the test set to identify periods of time where the classifiers struggle, and find a divergence in the regions where the convolutional and recurrent approaches underperform (Fig. 9).

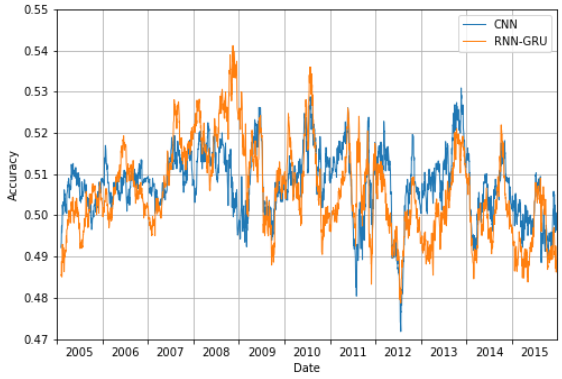

Figure 9: 3-month rolling mean of CNN and RNN-GRU model accuracy over the test window. The models struggle to varying degrees with periods of systemic uncertainty such as the global financial crisis of 2008, US debt-ceiling crisis of 2011 and Eurozone debt crisis of 2012.

In particular, the RNN architecture handles the global financial crisis (Q4-2008) far better than the CNN. Conversely, convolution outperforms recurrence in periods of stable market growth such as 2013. Both models dip markedly, and synchronously, below 50% following the US debt-ceiling crisis (Q3-2011) and the Eurozone

sovereign debt crisis (Q2-2012), suggesting some underlying sensitivity to systemic risk. Nevertheless, the divergent efficacy of the two approaches raises the prospect of better performance through ensemble modelling or joint CNN-RNN architectures, presenting further avenues for investigation.

3.7 Performance Benchmarks

Deep learning has garnered significant attention in recent years for its ability to outperform alternative methods, setting the state-of-the-art in computer vision and speech recognition benchmarks. The lack of commonly-agreed datasets such as MNIST for digit recognition or ImageNet for image classification means finance has lacked a stable backdrop for model benchmarking. For our purposes, we propose the use of the S&P500 technicals dataset for Jan 1994 - Dec 2015 as a baseline against which to evaluate other classifiers and benchmark deep learning in finance.

3.7.1 k-Nearest Neighbours (k-NN). We begin with a range of nearest-neighbours classifiers, labeling each day of the test set with the most frequently observed class label (positive or negative next-day return) in the k training points that were closest in Euclidean space.

3.7.2 Support Vector Machines (SVM). SVMs have been applied to financial time series forecasting in prior literature, and achieved moderate success when the input features were not raw price data but hand-crafted arithmetic derivations thereof called technical indicators (Kim, 2003). We report SVM performance under different kernel assumptions (linear and RBF), where the model hyperparameters (penalty parameter C, RBF kernel coefficient γ) were selected by cross-validation on a subset of the training data.

3.7.3 Random Forests (n-RF). In their study of European financial markets, Ballings et al (2015) evaluated the classification accuracy of ensemble methods against single classifiers. Their empirical work highlighted the effectiveness of random forests in classifying stock price movements and motivates their inclusion in our list of benchmarks, under varying assumptions for the number of trees hyperparameter n.

3.7.4 Summary. The results summarised in Table 10 underscore the scale of the challenge for pattern recognition in finance: deep learning achieved the best results but only by a small margin, and none of the methods achieved accuracies materially distinctive from guesswork.

Table 10: Benchmark performance across a range of supervised learning models trained on S&P500 technical data for Jan 1994 - Dec 1994 and tested on Jan 2005 - Dec 2015.

Model	Accuracy (%)	AUC (%)	Z	Significance
NN-Raw	50.1	50.2	2.267	0.9881
NN-Technicals	49.5	49.4	−10.773	–
CNN	50.8	51.0	18.677	> 0.9999
RNN-LSTM	50.6	50.8	14.534	> 0.9999
RNN-GRU	50.7	50.9	16.907	> 0.9999
1-NN	50.0	50.0	0.020	0.5080
10-NN	48.0	50.1	1.215	0.8874
100-NN	50.4	49.9	-2.270	–
Linear SVM	50.5	49.9	-2.061	–
RBF SVM	49.9	49.8	-2.416	–
10-RF	50.0	49.9	-1.082	–
50-RF	49.9	49.8	-2.929	–
100-RF	49.9	49.9	-2.793	–

4 CONCLUSION

Our results present to our knowledge the first rigorous statistical evaluation of candlestick patterns in time series analysis, using normalised signal cross-correlation to identify pattern matches. We find no evidence of predictive prowess in any of the pictograms, and suspect that the enduring quality of such practices owes much to their subjective and hitherto unverified nature. Nevertheless, it is not inconceivable that price history might contain predictive information, and much of quantitative finance practice relies on elements of technical pattern recognition (e.g., momentum-tracking) for its success. Through a deep learning lens, technical analysis is merely an arbitrary and incorrect specification of the feature-extractive early layers of a neural network. Even within relatively shallow architectures, learning more effective filters from data enhances performance - though only up to a point. The predictive information embedded in price history appears limited, and even state-of-the-art techniques in pattern recognition remain subject to that upper bound.

REFERENCES

[1] Allen, F. and Karjailainen, R. (1999). Using Genetic Algorithms to find Technical Trading Rules. *Journal of Financial Economics*, 51:245–271.

[3] Ballings, M., Van den Poel, D., Hespeels, N. and Gryp, R. (2015). Evaluating Multiple Classifiers for Stock Price Direction Prediction. *Expert Systems with applications*, 42(20):7046–7056.

[3] Bengio, J. (2009). Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127.

[4] Blume, L., Easley, D. and O'Hara, M. (1994). Market Statistics and Technical Analysis: The Role of Volume. *Journal of Finance*, 49(1):153–181.

[5] Chung, J, Gulcehre, C., Cho, K. and Bengio Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *NIPS 2014 Deep Learning and Representation Learning Workshop*.

[6] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[7] Hinton, G. E., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

[8] Jozefowicz, R., Zaremba, W and Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. *Journal of Machine Learning Research*, 37:2342–2350.

[9] Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307–319.

[10] Krizhevsky, A, Sutskever, I and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 2012*, 1106-1114.

[11] Lo, A. W., Mamaysky, H. and Wang, J. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *Journal of Finance*, 55(5):1706–1765.

[12] Mason, S. J., Graham, N. E. (2002). Areas beneath the Relative Operating Characteristics (ROC) and Relative Operating Levels (ROL) Curves: Statistical Significance and Interpretation. *Quarterly Journal of the Royal Meteorological Society*, 128:2145-fi?!2166.

[13] Neely, C., Weller, P. and Dittmar, R. (1997). Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach. *Journal of Financial and Quantitative Analysis*, 32(4):405–426.

[14] Romaszko, L. (2015). Signal Correlation Prediction Using Convolutional Neural Networks. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 46:45–56.

[15] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
[16] Taylor, M. P. and Allen, H. (1992). The Use of Technical Analysis in the Foreign Exchange Market. *Journal of International Money and Finance*, 11(3):304–314.