# Stock Market Prediction Based on CNN with Attention Mechanism

SYLVIA CHU*† and WEIGE HUANG‡

†Department of Industrial Engineering and Operations Research, Columbia University, New York, NY, USA
‡Shenzhen MSU-BIT University, Shenzhen, China

Image-based deep learning has shown that price charts can help predict the cross-section of stock returns, and attention layers are often added in the hope of better accuracy and interpretability. We ask what a CNN with spatial attention (CNNam) can and cannot learn from 30-day candlestick charts of a single, highly liquid index, the S&P 500, using a strictly chronological design with 4,195 out-of-sample days from 2008 to 2024 and five random seeds. Against a plain CNN, three recurrent networks and a logistic regression, no model beats the rule that always predicts an up day, and all AUCs lie between 0.48 and 0.50. Attention nonetheless does something useful: it significantly improves the calibration of the CNN's probabilities, which makes CNNam the only network statistically indistinguishable from the best constant forecast, without adding discrimination. We then show how the same networks look predictive under a random split with overlapping five-day labels (AUC 0.57–0.59, which vanishes under a purged block split) or under a label computed inside the chart window (94% accuracy). We provide a reproducible protocol of benchmarks and diagnostics for image-based forecasting.

*Keywords*: Candlestick charts; Convolutional neural networks; Attention mechanism; Directional forecasting; Forecast evaluation; Data leakage

*JEL Classification*: C45; C53; G14; G17

## 1. Introduction

Technical analysts have read candlestick charts for centuries (Nison 2001), and whether visual price patterns contain information about future prices has long been debated in finance (Lo *et al.* 2000, Marshall *et al.* 2006). Convolutional neural networks (CNNs) have given this question new life. Jiang *et al.* (2023) train CNNs on images of daily price and volume histories of U.S. stocks and show that the resulting signals predict the cross-section of returns and generate large long–short portfolio returns. Murray *et al.* (2024) reach a related conclusion, and a growing literature applies image-based models to candlestick charts of stock indices (Chen and Huang 2024, Lin *et al.* 2026). A natural next step is to add an attention layer (Bahdanau *et al.* 2015, Vaswani *et al.* 2017), which lets the network weight different regions of a chart and promises both better accuracy and a view of which parts of the chart the model uses. This paper follows that path to the place where it is hardest to walk: a single, highly liquid index, a daily horizon and a strictly chronological evaluation. We render 30-day candlestick charts of the S&P 500 with a 20-day moving average and trading volume, and we compare a CNN with additive spatial attention (CNNam) with an otherwise identical CNN, three recurrent networks (RNN, LSTM, GRU) that read the same information as numbers, and a logistic regression. All models are estimated on 1962–1999, tuned on 2000–2007

*Corresponding author. Email: zc2882@columbia.edu.

and evaluated on 2008–2024. We ask three questions. Can a CNN with attention read the direction of the index from its chart? If not, what exactly does attention add? And how would a researcher know the difference between a model that predicts and one that only appears to? The first answer is that the index keeps its secrets. Over the 4,195 test days, the index rises on 54.2% of days, which is also the accuracy of the rule that always predicts 'up'. No model exceeds it: CNNam attains 52.3%, the CNN 51.1% and the best recurrent network (LSTM) 54.0%. The AUC of every model is between 0.48 and 0.50, so the predicted probabilities do not rank up days above down days, and the Pesaran–Timmermann test (Pesaran and Timmermann 1992) does not reject the null of no directional predictability for any model. Market-timing strategies built on the forecasts earn lower Sharpe ratios than holding the index. These findings hold at a five-day horizon, in three subperiods and when the candle body is redefined to avoid an artefact in the historical opening prices. The second answer is where attention earns its place. CNNam's probability forecasts have a significantly lower Brier score than the CNN's, and CNNam is the only network whose Brier score is not significantly worse than that of the best constant forecast (Diebold and Mariano 1995, Harvey *et al.* 1997). The two networks have the same AUC, so the gain comes from calibration, not discrimination: the CNN's probabilities are systematically too low, while attention keeps the forecasts close to the unconditional probability, which is the best forecast when charts carry no information. In our setting, attention acts as a regularizer rather than as a detector of patterns. Perturbation experiments support this view: removing the moving average or the volume panel does not reduce the accuracy of either network, and grayscale charts perform as well as colour charts. The attention maps themselves look sensible, concentrating on price extremes and on the first and last days of the window, which shows that an intuitive map is not evidence of predictive content. The third answer concerns how predictability can be manufactured. The same data and the same networks can be made to look predictive with two ordinary design choices. When five-day labels overlap and the sample is split at random rather than by date, the AUC of the CNN rises from 0.52 to 0.57 and that of CNNam from 0.53 to 0.59, because training and test charts a few days apart share most of their future returns. A purged block split, which keeps the random assignment but removes overlapping observations, brings the AUCs back to 0.51 and 0.50. When the label is the direction of the five-day return that ends on the last day of the chart, which is visible in the image itself, CNNam reaches an accuracy of 94% and an AUC of 0.99. Neither number says anything about the future, and both are easy to obtain by accident. The paper contributes in three ways. First, it provides a transparent, reproducible baseline for image-based forecasting of a stock index, anchored by the always-up and constant-probability benchmarks, threshold-free and proper scoring measures, formal tests and multiple seeds, against which future architectures can be compared. The successful applications of image-based models in the literature are cross-sectional (Jiang *et al.* 2023, Murray *et al.* 2024, Chen and Huang 2024, Lin *et al.* 2026), and our results mark the boundary of what the same tools can do for a single index at a daily horizon. Second, it adds to work on attention in financial forecasting (Qin *et al.* 2017, Niu *et al.* 2021) by separating the effect of attention on calibration from its effect on discrimination, and by showing that attention maps can look interpretable without predictive power. Third, it contributes to the evaluation of machine-learning forecasts in finance (Welch and Goyal 2008, Harvey *et al.* 2016, Gu *et al.* 2020, Hou *et al.* 2020) by quantifying how much overlapping labels, random splits and in-window labels can inflate performance (Kaufman *et al.* 2012, López de Prado 2018, Kapoor and Narayanan 2023), and by documenting two features of the widely used Yahoo Finance S&P 500 data that affect any study that draws candlesticks from them. The rest of the paper is organized as follows. Section 2 reviews the literature. Sections 3 and 4 describe the data and the methods. Section 5 reports the main results, section 6 examines what the networks learn and section 7 reports robustness checks. Section 8 shows how evaluation choices can inflate performance, and section 9 concludes. Appendix A lists implementation details.

## 2. Related literature

### 2.1. *Machine learning for stock market prediction*

Stock prices are hard to forecast because they are noisy and nonstationary (Abu-Mostafa and Atiya 1996). Even so, many studies find that machine-learning methods extract some predictive information that linear models miss. Early work combined neural networks with genetic algorithms for feature discretization (Kim and Han 2000), and Chen and Hao (2017) propose feature-weighted support vector machines and $k$-nearest neighbours for Chinese stock indices. In empirical asset pricing, Gu *et al.* (2020) show that tree ensembles and neural networks improve the prediction of individual stock returns relative to linear benchmarks, mainly because they capture nonlinear interactions among predictors, and Chen *et al.* (2024) estimate asset-pricing models with deep networks. Recurrent architectures such as LSTM and GRU are widely used for financial time series (Bhambu 2023), hybrid designs combine convolutional and recurrent layers (Rasheed *et al.* 2020), and convolutional networks have been applied both to numerical predictors (Hoseinzade and Haratizadeh 2019) and to measures of financial stress (Mezghani and Abbes 2023).

### 2.2. *Candlestick charts and technical analysis*

Candlestick charts originated in Japanese rice trading and were popularized in the West by Nison (2001); see also Wagner and Matheny (1994), Tudela (2008) and Bulkowski (2008). Lo *et al.* (2000) develop a statistical framework for detecting chart patterns and find that several patterns carry incremental information, whereas Marshall *et al.* (2006) find that traditional candlestick trading rules do not create value for Dow Jones Industrial Average stocks. Studies that automate pattern recognition include Kamo and Dagli (2009), who combine candlestick rules with neural networks; Martiny (2012), who discovers significant patterns without supervision; Prado *et al.* (2013), who evaluate candlestick analysis in the Brazilian market; Tsai and Quan (2014), who predict prices by searching for similar historical charts; and Ahmadi *et al.* (2016, 2018), who use candlestick inputs for market timing. Fengqian and Chao (2020) use decomposed candlestick features in a deep reinforcement learning trading system, and Unlu (2020) proposes a colour-coded variant of candlesticks for deep learning.

### 2.3. *Image-based return prediction*

A growing literature treats price histories as images rather than as numerical sequences. Early work in computer science converts time series into two-dimensional representations and feeds them to convolutional networks (Sezer and Ozbayoglu 2018, Hoseinzade and Haratizadeh 2019), or learns representations directly from candlestick charts (Hu *et al.* 2018). The approach entered mainstream empirical finance with Jiang *et al.* (2023), who train CNNs on standardized black-and-white images of daily open, high, low and close prices, moving averages and trading volume for U.S. stocks. They show that the image-based signals predict the direction of future returns, generate economically large long–short portfolio returns and are not subsumed by conventional trend signals such as momentum and short-term reversal. Murray *et al.* (2024) reach a related conclusion using CNNs trained on historical price paths. Closest to our setting, Chen and Huang (2024) apply a CNN to candlestick charts of the Chinese and U.S. stock indices and find that the charts carry more predictive information in the Chinese market, which they relate to its larger share of retail investors. More recently, Lin *et al.* (2026) compare and combine line, candlestick and bar charts as inputs to image-based deep learning models.

Two features of this literature matter for our study. First, the strongest evidence in Jiang *et al.* (2023) and Murray *et al.* (2024) is cross-sectional: the CNN ranks thousands of stocks against each other, and the gains come from sorting stocks into portfolios. Predicting the time-series direction of a single, highly liquid index is a different and harder problem, because any predictable component

of index returns is small relative to their volatility (Fama 1970, Welch and Goyal 2008). Second, attention mechanisms (Bahdanau *et al.* 2015, Vaswani *et al.* 2017) have improved forecasting for financial time series (Qin *et al.* 2017); see Niu *et al.* (2021) for a survey. Attention is often motivated by interpretability, because its weights show which parts of the input the model uses. To our knowledge, no study has tested under a strict chronological design whether attention improves the out-of-sample accuracy of CNNs that read candlestick charts of a stock index.

### 2.4. *Evaluating machine-learning forecasts of returns*

Many published forecasting gains in finance do not survive careful out-of-sample evaluation. Welch and Goyal (2008) show that most predictors of the equity premium fail to beat the historical mean out of sample, and Harvey *et al.* (2016) and Hou *et al.* (2020) document that many cross-sectional anomalies do not survive multiple-testing adjustments or replication. In machine learning, *leakage*, that is, information in the training data that would not be available at the time of the forecast, is a common reason why reported performance cannot be reproduced (Kaufman *et al.* 2012, Kapoor and Narayanan 2023). Financial data are especially prone to leakage because labels computed over overlapping horizons are serially dependent, so randomly shuffled training and test sets share information; López de Prado (2018) proposes purging and embargoing overlapping observations as a remedy. Repeatedly backtesting many specifications causes a related problem, backtest overfitting (Bailey *et al.* 2014).

A second issue is the choice of benchmark and loss function. For a binary target, the natural benchmarks are the unconditional frequency of up days and a test of directional predictability (Pesaran and Timmermann 1992, Christoffersen and Diebold 2006). Probability forecasts should be judged by a proper scoring rule such as the Brier score (Brier 1950), and differences in loss between models should be tested formally (Diebold and Mariano 1995, Harvey *et al.* 1997). A study that reports only the mean squared error of a continuous output against a binary label, without a naive benchmark, cannot tell whether a model predicts better than a constant.

## 3. Data

We use daily data on the S&P 500 index (ticker `^GSPC`) from Yahoo Finance, which include the open, high, low and close prices and trading volume. Figure 1 plots the closing price. Two features of these data matter for chart construction, and we report them because they affect any study that builds candlestick charts from this source. First, from 1950 to 1961 the open, high, low and close are identical on every trading day, so the candlesticks have neither bodies nor shadows. We therefore start the sample on 2 January 1962. Second, between 1962 and 2006 the reported opening price equals the previous day's close on 83% of trading days, and on at least 85% of days in 31 of these 45 years; from 2008 onwards this happens on only 4% of days. Before 2007, a candle's body therefore mostly measures the close-to-close return rather than the intraday move. Section 7 shows that our conclusions do not change when we impose the close-to-close definition of the body throughout the sample.

The final sample runs from January 1962 to 3 September 2024. Each observation is a 30-day chart that ends on day $t$. The label $y_t$ equals one if the index closes higher on day $t + h$ than on day $t$ and zero otherwise. Our main horizon is $h = 1$ (the next trading day), and we report $h = 5$ as a robustness check. We split the sample chronologically: models are estimated on 1962–1999 (training set), tuned and early-stopped on 2000–2007 (validation set), and evaluated on 2008–2024 (test set), which the models never see during estimation. To prevent labels from overlapping across the split boundaries, we drop the last $h$ observations of the training and validation periods. In the test period the index rises on 54.2% of days, so a naive rule that always predicts 'up' attains an accuracy of 54.2%; this is the relevant benchmark for every model below.

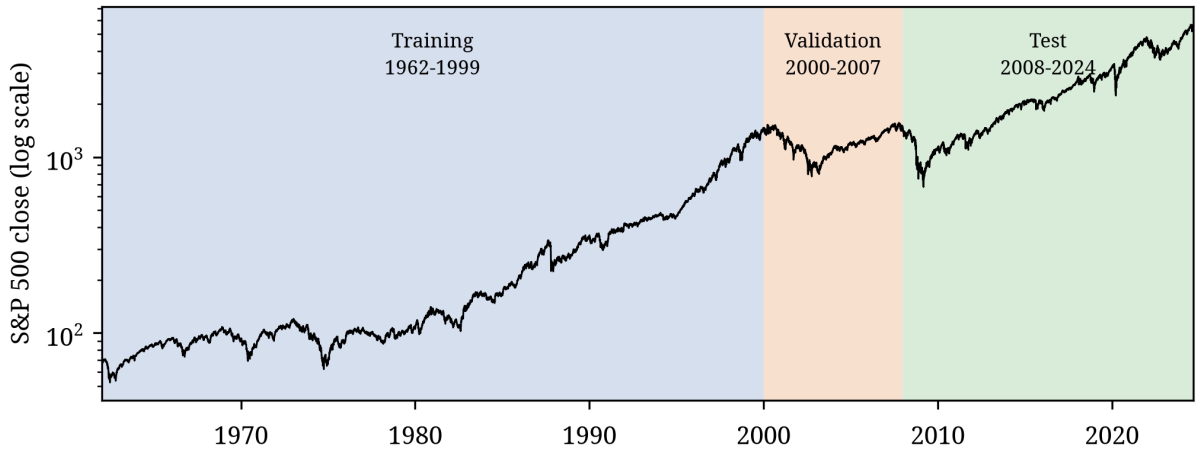


Figure 1. S&P 500 closing price (log scale), January 1962 to September 2024, and the chronological split into training, validation and test periods.

## 4. Methodology

### 4.1. *Candlestick images*

Each chart covers $W = 30$ trading days and is rendered as a $64 \times 90$ pixel image, with three pixel columns per day (figure 2). The top 51 rows show the price panel and the bottom 12 rows show trading volume, separated by one blank row. For each day, the body spans the open and close over all three columns, and the shadow spans the high and low in the centre column. A 20-day moving average of the close is drawn as a one-pixel line. Within each image, prices are scaled to the range between the window's lowest low and highest high (including the moving average), and volume is scaled by the window's largest volume. Scaling each window in this way removes the price level, so the model sees only the shape of the chart. The design follows Jiang *et al.* (2023), except that we draw candlesticks rather than OHLC bars.

We consider three renderings of the same information. In the *colour* version (our baseline), up days (close $\geq$ open) are drawn in green and down days in red on a black background. In the *grayscale-hollow* version, up days are drawn as hollow bodies and down days as filled bodies, so that direction is encoded by shape instead of colour. In the *grayscale-filled* version, every body is filled with the same intensity, so the direction of a single candle is not shown directly.

### 4.2. *CNN and CNNam*

The baseline CNN has three convolutional blocks. Each block applies a $5 \times 3$ convolution, batch normalization, a leaky ReLU activation and $2 \times 2$ max pooling, with 16, 32 and 64 filters. The final $64 \times 8 \times 11$ feature map is flattened, passed through dropout (rate 0.5) and mapped to a single logit for $P(y_t = 1)$.

CNNam adds an attention layer between the last convolutional block and the classifier. Let $\mathbf{f}_i \in \mathrm{R}^{64}$ denote the feature vector at spatial position $i$ ($i = 1, \ldots, N$, $N = 88$) of the final feature map. The layer computes an energy score

$$e_i = \mathbf{v}^\top \tanh\left(\mathbf{W}\mathbf{f}_i + \mathbf{b}\right), \tag{1}$$

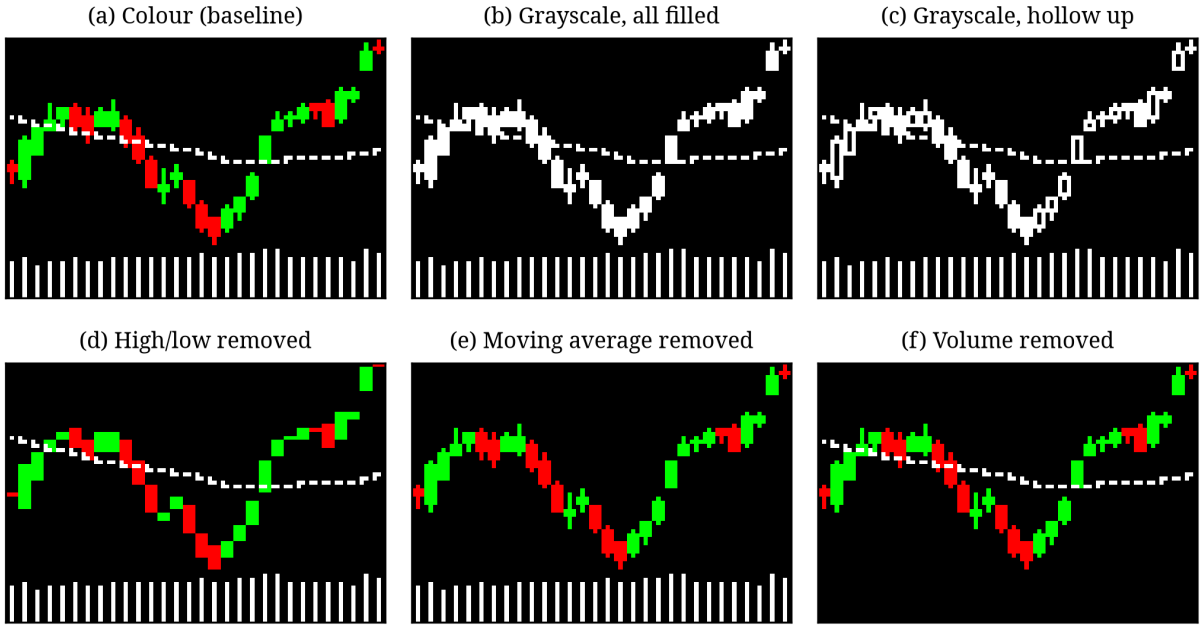


Figure 2. One 30-day chart in the six renderings used in the paper. Each image is $64 \times 90$ pixels, with the price panel (candlesticks and the 20-day moving average) on top and the volume panel at the bottom. Panels (a)–(c) are used for training and testing; panels (d)–(f) are used only at test time in the perturbation analysis of section 6.1.

normalizes the scores across positions,

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{N} \exp(e_j)}, \tag{2}$$

and reweights the features, $\tilde{\mathbf{f}}_i = N\alpha_i \mathbf{f}_i$, so that the average weight equals one. This is additive attention in the sense of Bahdanau *et al.* (2015), applied over the spatial positions of the chart. Because $\alpha_i$ depends on the input, the network can emphasize different parts of each chart, for example the most recent days or the volume panel, and the weights can be averaged across charts to show which regions the model relies on. Apart from the attention layer, CNNam is identical to the CNN, so any difference in performance can be attributed to attention.

### 4.3. *Benchmark models*

The RNN, LSTM and GRU benchmarks each have one recurrent layer with 64 hidden units. They take as input the same 30-day window in numerical form: the open, high, low, close and moving average expressed relative to the last close, plus log volume standardized within the window. A logistic regression on the same 180 inputs serves as a linear benchmark.

### 4.4. *Training and evaluation*

All models are trained with the binary cross-entropy loss and the Adam optimizer (learning rate $10^{-3}$, batch size 128, weight decay $10^{-5}$). Training stops when the validation loss has not improved for four epochs, and we keep the parameters with the lowest validation loss. Because the results of neural networks depend on the random initialization, we train each model with five random seeds (three in the robustness checks). We report the mean and standard deviation across seeds and also evaluate the ensemble that averages the predicted probabilities across seeds.

Table 1. Out-of-sample directional prediction of the S&P 500, one-day horizon (test period 2 January 2008 to 30 August 2024, $N = 4{,}195$).

| Model | Accuracy | AUC | MCC | Brier | PT | DM vs. CNNam | DM vs. const. |
|---|---|---|---|---|---|---|---|
| Always up | 0.542 | 0.500 | 0.000 | – | – | – | – |
| Constant, $p = 0.524$ | – | 0.500 | – | 0.2485 | – | – | – |
| LOGIT | 0.513 (0.013) | 0.498 (0.011) | −0.001 | 0.2500 | −0.05 | −1.71 [0.088] | 3.56 [0.000] |
| RNN | 0.534 (0.004) | 0.483 (0.002) | −0.026 | 0.2493 | −1.70 | −0.38 [0.702] | 2.86 [0.004] |
| LSTM | 0.540 (0.001) | 0.484 (0.002) | −0.019 | 0.2489 | −1.25 | 0.48 [0.633] | 3.06 [0.002] |
| GRU | 0.537 (0.000) | 0.484 (0.001) | −0.008 | 0.2492 | −0.55 | −0.04 [0.966] | 3.26 [0.001] |
| CNN | 0.511 (0.017) | 0.501 (0.003) | −0.005 | 0.2504 | −0.32 | −3.40 [0.001] | 3.24 [0.001] |
| CNNam | 0.523 (0.012) | 0.500 (0.005) | −0.002 | 0.2491 | −0.14 | – | 1.59 [0.113] |

Notes: Accuracy and AUC are the mean (standard deviation) across five random seeds. MCC, Brier score and the Pesaran–Timmermann (PT) statistic are computed for the ensemble that averages predicted probabilities across seeds. 'Constant' forecasts the training-sample frequency of up days (0.524) on every day. DM vs. CNNam is the Diebold–Mariano statistic for the Brier-loss differential CNNam minus model, so a negative value favours CNNam. DM vs. const. is the statistic for model minus constant, so a positive value means that the model is worse than the constant forecast. Both use the Harvey *et al.* (1997) correction and a Newey–West variance; two-sided $p$-values are in brackets.

We evaluate the out-of-sample predictions with several measures. Accuracy is the share of days whose direction is predicted correctly, which we compare with the always-up benchmark. The AUC measures how well the predicted probabilities rank up days above down days and does not depend on a classification threshold. We also report the Matthews correlation coefficient (MCC) and the Brier score. To test whether directional forecasts contain information, we use the Pesaran and Timmermann (1992) test. To compare models, we apply the Diebold and Mariano (1995) test to the Brier-loss differential, with the small-sample correction of Harvey *et al.* (1997) and a Newey–West variance estimate. Finally, to measure economic value, we form a market-timing strategy that holds the index when the predicted probability of an up day exceeds 0.5 and holds cash otherwise, and we report its Sharpe ratio before and after a transaction cost of 5 basis points per switch.

## 5. Main results

### 5.1. *Directional accuracy*

Table 1 reports the out-of-sample performance of all models at the one-day horizon, and figure 3 plots accuracy and AUC. The first row is the always-up rule, whose accuracy of 0.542 equals the share of up days in the test period. No model beats this benchmark. The recurrent networks come closest (LSTM 0.540, GRU 0.537, RNN 0.534), but they do so by predicting 'up' on almost every day: the LSTM ensemble predicts an up day 99% of the time. CNNam attains 0.523 and the CNN 0.511. The seed-to-seed standard deviation of accuracy for the image models (1.2 and 1.7 percentage points) is as large as the differences between models, which shows why results based on a single training run can be misleading.

Accuracy depends on the classification threshold and rewards models that simply follow the majority class. The AUC does not have these problems. It equals 0.500 for CNNam, 0.501 for the CNN and between 0.483 and 0.498 for the other models, so none of the models ranks up days above down days better than chance. The Matthews correlation coefficients are essentially zero, and the Pesaran–Timmermann statistics range from −1.70 to −0.05, far from rejecting the null of no predictability.

### 5.2. *Probability forecasts and the role of attention*

The Brier score evaluates the predicted probabilities directly and is a proper scoring rule (Brier 1950). A natural benchmark is a constant forecast equal to the frequency of up days in the training

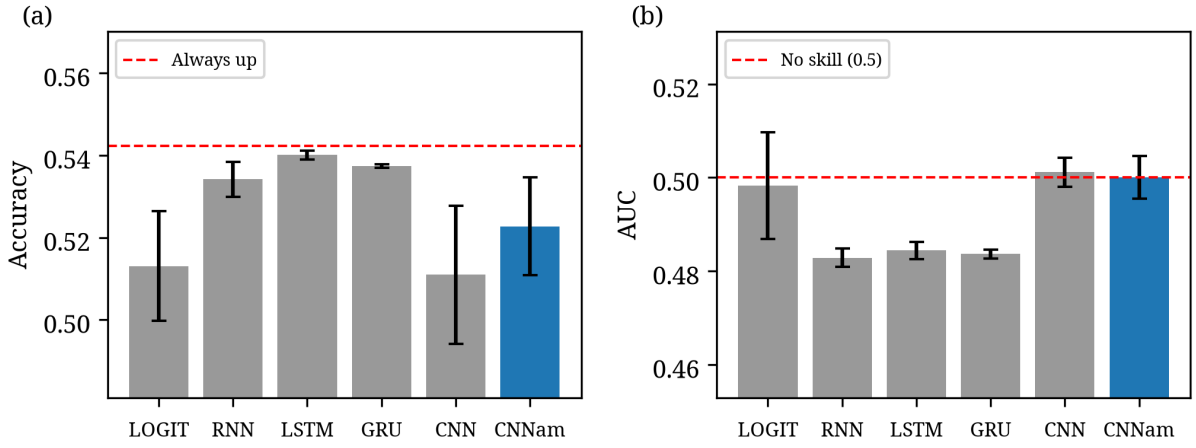


Figure 3. Out-of-sample accuracy (a) and AUC (b) at the one-day horizon, 2008–2024. Bars show the mean across five seeds and whiskers one standard deviation. The dashed lines mark the always-up accuracy and the AUC of an uninformative forecast.

sample, 0.524, which uses no information from the charts. Its Brier score of 0.2485 is lower than that of every model in table 1. The last column shows that the difference is statistically significant for the logistic regression, the three recurrent networks and the CNN. CNNam is the only network that is not significantly worse than the constant forecast (DM $= 1.59$, $p = 0.11$).

Attention therefore matters, but not in the way it is usually motivated. CNNam has a significantly lower Brier score than the CNN (DM $= -3.40$, $p = 0.001$), whereas the two models have the same AUC. The improvement is in calibration, not discrimination: the CNN's average predicted probability is 0.511, below the base rate, and it predicts an up day on only 61% of test days, while CNNam's average probability is 0.524 and it predicts an up day on 82% of days. In other words, attention keeps the network's forecasts closer to the unconditional probability, which is the best available forecast when the charts carry no information.

### 5.3. *Five-day horizon*

Daily returns may be too noisy for chart patterns to show up, so table 2 repeats the analysis for the direction of the five-day return. The always-up benchmark is now 0.595, and again no model beats it. CNNam has the highest AUC (0.523) and the highest Pesaran–Timmermann statistic (1.67), but the statistic is below the 5% critical value of a two-sided test (1.96), and the Diebold–Mariano tests do not distinguish CNNam from the CNN or the recurrent networks. The constant forecast (0.560) has a Brier score of 0.2423, again lower than that of every model, and the difference is significant for all models except the CNN and CNNam.

### 5.4. *Economic value*

Statistical accuracy and economic value need not coincide (Christoffersen and Diebold 2006), so we also evaluate a market-timing rule that holds the index on days when the predicted probability of an up day exceeds 0.5 and holds cash (with zero return) otherwise. Table 3 reports annualized returns and Sharpe ratios computed from daily returns without subtracting the risk-free rate. Buy-and-hold earns 10.1% per year with a Sharpe ratio of 0.50 over the test period, and every timing strategy earns less. CNNam comes closest, with a Sharpe ratio of 0.40 before and 0.31 after a cost of 5 basis points per switch. The CNN-based strategy, which is out of the market 39% of the time, earns 0.37 before and 0.24 after costs. The forecasts thus have no economic value for an investor

Table 2. Out-of-sample directional prediction of the S&P 500, five-day horizon (test period 2 January 2008 to 26 August 2024, $N = 4{,}191$).

| Model | Accuracy | AUC | MCC | Brier | PT | DM vs. CNNam | DM vs. const. |
|---|---|---|---|---|---|---|---|
| Always up | 0.595 | 0.500 | 0.000 | – | – | – | – |
| Constant, $p = 0.560$ | – | 0.500 | – | 0.2423 | – | – | – |
| LOGIT | 0.534 (0.033) | 0.476 (0.010) | −0.044 | 0.2480 | −2.83 | −3.10 [0.002] | 5.45 [0.000] |
| RNN | 0.590 (0.003) | 0.488 (0.001) | −0.002 | 0.2445 | −0.10 | −0.22 [0.829] | 4.13 [0.000] |
| LSTM | 0.594 (0.002) | 0.484 (0.001) | −0.018 | 0.2444 | −1.17 | −0.12 [0.906] | 3.55 [0.000] |
| GRU | 0.579 (0.021) | 0.487 (0.001) | −0.018 | 0.2447 | −1.19 | −0.35 [0.725] | 3.56 [0.000] |
| CNN | 0.561 (0.023) | 0.513 (0.009) | 0.011 | 0.2446 | 0.71 | −0.30 [0.766] | 1.43 [0.151] |
| CNNam | 0.554 (0.015) | 0.523 (0.015) | 0.026 | 0.2442 | 1.67 | – | 1.49 [0.137] |

Notes: See table 1. The constant forecast equals the training-sample frequency of positive five-day returns (0.560). The Newey–West lag length accounts for the overlap of five-day labels.

Table 3. Market-timing strategies based on one-day predictions (long the index when $\hat{p} > 0.5$, otherwise cash), 2008–2024.

| Model | Ann. return | Ann. volatility | Sharpe | Sharpe (5 bp) | Time in market |
|---|---|---|---|---|---|
| Buy and hold | 0.101 | 0.203 | 0.50 | 0.50 | 1.000 |
| LOGIT | 0.046 | 0.169 | 0.27 | 0.02 | 0.708 |
| RNN | 0.042 | 0.182 | 0.23 | 0.19 | 0.963 |
| LSTM | 0.060 | 0.185 | 0.33 | 0.31 | 0.987 |
| GRU | 0.062 | 0.180 | 0.34 | 0.32 | 0.969 |
| CNN | 0.061 | 0.162 | 0.37 | 0.24 | 0.608 |
| CNNam | 0.075 | 0.186 | 0.40 | 0.31 | 0.815 |

Notes: Strategies use the seed-averaged ensemble forecasts. Returns are price returns of the S&P 500, excluding dividends. Sharpe ratios are annualized ratios of the mean to the standard deviation of daily returns. The transaction cost is charged each time the position switches between the index and cash.

who can simply hold the index.

## 6. What do the networks learn?

### 6.1. *Perturbation analysis*

To see which chart components the image models rely on, we remove one component at a time from the test images and keep everything else unchanged: the high–low shadows, the moving average or the volume panel (figure 2, panels (d)–(f)). The networks are not retrained, so any change in accuracy shows how much the trained model depends on that component. Table 4 and figure 4 report the changes with 95% moving-block bootstrap confidence intervals.

None of the removals reduces accuracy. Removing the shadows has no significant effect on either network. Removing the moving average or the volume panel *raises* the CNN's accuracy, by 3.1 and 3.9 percentage points, and both intervals exclude zero. The mechanism is again the base rate: without these components the CNN predicts an up day on 86% and 96% of test days, compared with 61% for the complete images, and so moves towards the always-up rule. CNNam's accuracy changes by at most 1.1 percentage points, and no change is significant, which is consistent with attention making the network's output less sensitive to the input. Because the components can be removed without loss, we find no evidence that any of them carries information the models use to predict the next day's direction.

Table 4. Perturbation analysis: change in out-of-sample accuracy (percentage points) when one chart component is removed from the test images, one-day horizon.

| Removed component | ΔAccuracy, CNN | ΔAccuracy, CNNam | CNNam − CNN |
|---|---|---|---|
| High/low (shadows) | 0.17 [−0.81, 1.14] | −0.45 [−1.43, 0.48] | −0.62 [−1.91, 0.57] |
| Moving average | 3.10 [1.53, 4.70] | 1.05 [−0.17, 2.26] | −2.05 [−3.53, −0.48] |
| Volume | 3.86 [2.07, 5.65] | 0.41 [−0.38, 1.19] | −3.46 [−5.36, −1.36] |

Notes: Changes are for seed-averaged ensembles; a positive value means that accuracy rises when the component is removed. 95% moving-block bootstrap confidence intervals (block length 20 days, 2,000 draws) are in brackets.

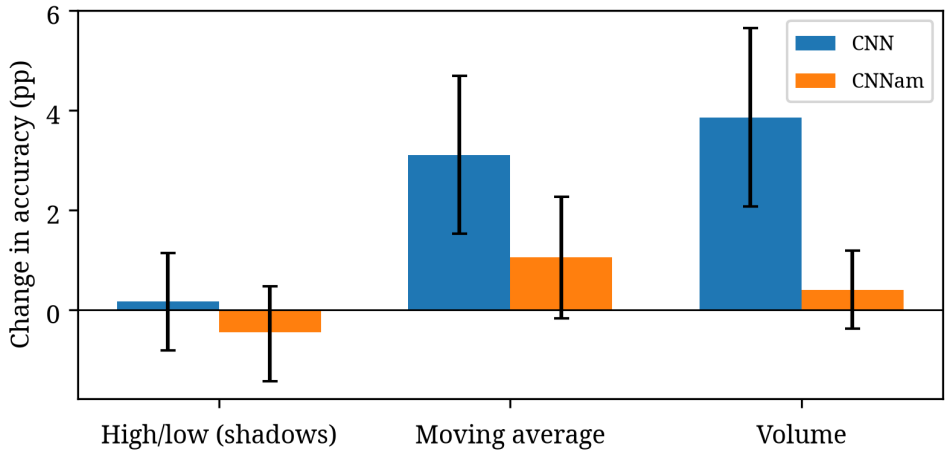


Figure 4. Change in out-of-sample accuracy when one chart component is removed from the test images. Whiskers show 95% moving-block bootstrap confidence intervals.

### 6.2. *Attention weights*

Figure 5 shows CNNam's attention weights averaged over the test charts. Each cell corresponds to one of the 88 positions of the final feature map, which covers roughly $8 \times 8$ pixels of the input, or about three trading days. The weights are highest in the top rows of the price panel, where the window's highest prices are drawn, and in the lower right of the price panel, where recent prices lie after a decline. The volume panel receives less than half of the uniform weight of 1/88. Summed over rows, the weights are close to uniform across time, with slightly more weight on the first and last days of the window.

The map looks sensible: a chartist might also focus on price extremes and recent days. Our other results, however, show that the network does not turn this focus into predictive power. Attention weights describe where a model looks, not whether what it sees is informative, and they should not be read as evidence of predictability on their own.

### 6.3. *Colour versus grayscale*

Candlestick charts usually encode up and down days by colour. To test whether colour helps, we retrain CNNam on the two grayscale versions of the same charts. Table 5 shows that the three versions perform the same: accuracy is 0.523 for colour, 0.527 for hollow/filled and 0.524 for all-filled, the AUC is about 0.50 in every case, and the Diebold–Mariano tests do not reject equal Brier scores. Colour does not help the network extract information, which is unsurprising given that there is little information to extract.

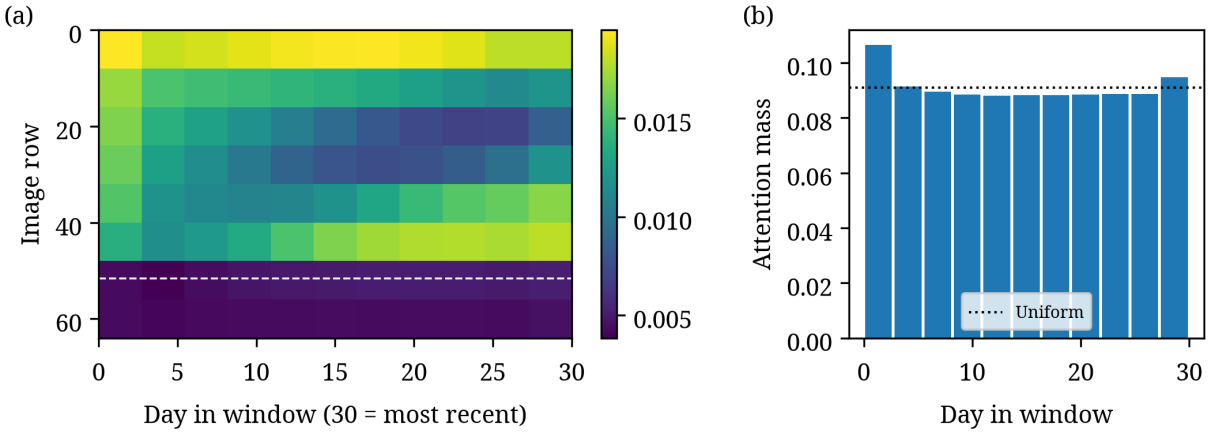


Figure 5. Average CNNam attention weights over the test period. (a) Weight of each position of the final feature map, plotted on the coordinates of the input image (price panel above the dashed line, volume below). (b) Total weight by position within the 30-day window; the dotted line is the uniform weight.

Table 5. CNNam with colour and grayscale candlestick images, one-day horizon.

| Input | Accuracy | AUC | Brier | DM (colour vs. this) |
|---|---|---|---|---|
| Colour (green up / red down) | 0.523 (0.012) | 0.500 (0.005) | 0.2491 | – |
| Grayscale, hollow up / filled down | 0.527 (0.005) | 0.498 (0.004) | 0.2493 | −0.52 [0.604] |
| Grayscale, all bodies filled | 0.524 (0.007) | 0.497 (0.005) | 0.2494 | −0.78 [0.436] |

Notes: Accuracy and AUC are the mean (standard deviation) across five seeds; the Brier score is for the seed-averaged ensemble. DM is the statistic for the Brier-loss differential colour minus grayscale, with two-sided $p$-values in brackets.

## 7. Robustness

### 7.1. *Opening prices*

As described in section 3, before 2007 the Yahoo Finance opening price usually equals the previous close, so a candle's body measures the close-to-close change in most of the training data but the intraday change in the test data. To remove this inconsistency, we redraw every chart with the open set to the previous day's close throughout the sample, so that the body has the same meaning in all periods, and retrain the CNN and CNNam with three seeds each. The ensemble accuracy is 0.497 for the CNN and 0.522 for CNNam, the AUCs are 0.500 and 0.501, and neither Pesaran–Timmermann statistic is significant. The change in the meaning of the body therefore does not explain the absence of predictability.

### 7.2. *Subperiods*

Table 6 splits the test period into 2008–2012, which contains the global financial crisis, 2013–2018 and 2019–2024, which contains the COVID-19 crash. In each subperiod the always-up rule is at least as accurate as every model, and no model has an AUC above 0.52. The image models are not more accurate in turbulent periods, when chart patterns might be expected to matter most.

Table 6. Accuracy and AUC by subperiod, one-day horizon (seed-averaged ensembles).

| | 2008–2012 | | 2013–2018 | | 2019–2024 | |
|---|---|---|---|---|---|---|
| Model | Accuracy | AUC | Accuracy | AUC | Accuracy | AUC |
| Always up | 0.541 | 0.500 | 0.540 | 0.500 | 0.546 | 0.500 |
| LOGIT | 0.540 | 0.515 | 0.493 | 0.474 | 0.523 | 0.502 |
| RNN | 0.528 | 0.470 | 0.534 | 0.496 | 0.539 | 0.477 |
| LSTM | 0.535 | 0.481 | 0.539 | 0.477 | 0.542 | 0.491 |
| GRU | 0.537 | 0.481 | 0.538 | 0.478 | 0.540 | 0.489 |
| CNN | 0.509 | 0.503 | 0.505 | 0.484 | 0.506 | 0.518 |
| CNNam | 0.512 | 0.488 | 0.528 | 0.497 | 0.536 | 0.507 |

## 8. How evaluation choices can inflate performance

A natural question follows from the results above: if the charts carry no usable information about the next day's direction, why do studies of image-based forecasts of index returns often report accuracies well above 50%? Without access to every study's code we cannot say why any particular result differs from ours, but we can show how much common design choices change the numbers when applied to our data and our networks. Table 7 reports the results, using seed 0 throughout, so that rows within each panel differ only in the design choice.

### 8.1. *Random splits with overlapping labels*

Some machine-learning studies divide the sample into training and test sets at random. With daily data and a one-day label, this is largely harmless for our purposes because the labels of neighbouring days do not overlap, and panel A of table 7 confirms that the AUC stays at about 0.49–0.50 under a random split. With a five-day label, however, the charts that end on days $t$ and $t+1$ are almost identical and their labels share four of five daily returns. A random split puts many such pairs on both sides of the split, so the network can score well on the test set by recognizing charts that are nearly the same as training charts. Panel B shows that the AUC rises from 0.520 to 0.568 for the CNN and from 0.534 to 0.588 for CNNam, gains that would be easy to mistake for predictability.

To confirm that overlap rather than the random assignment itself causes the gain, we also use a purged block split in the spirit of López de Prado (2018). We divide the sample into consecutive 21-day blocks, assign the blocks at random to training, validation and test sets in the same 70/10/20 proportions, and remove from the training and validation sets every observation within $h$ days of a test observation. The test periods are still scattered across 1962–2024, but no training label overlaps a test label. Under this split the AUC falls to 0.510 for the CNN and 0.504 for CNNam, at or below the level of the chronological design (0.52 and 0.53), and the accuracies of 0.547 and 0.524 are no better than the up share of the purged test set (0.544).

### 8.2. *Labels computed inside the window*

A more severe problem arises if the label is defined with information that is already in the image, for example the direction of the price movement over the last days of the chart rather than after it. To illustrate, we train CNNam with the label equal to the sign of the five-day return that ends on the last day of the window. Under the same chronological split, CNNam reaches an accuracy of 0.939 and an AUC of 0.986 (panel C). The network has learned to read the chart, but the result has no forecasting content.

Table 7. How design choices change measured performance (seed 0).

| Design | Model | Accuracy | AUC | Share of up labels |
|---|---|---|---|---|
| *Panel A: one-day label* | | | | |
| Chronological split | CNN | 0.491 | 0.501 | 0.542 |
| Random split | CNN | 0.509 | 0.488 | 0.550 |
| Chronological split | CNNam | 0.527 | 0.500 | 0.542 |
| Random split | CNNam | 0.494 | 0.496 | 0.550 |
| *Panel B: five-day label (overlapping)* | | | | |
| Chronological split | CNN | 0.536 | 0.520 | 0.595 |
| Random split | CNN | 0.584 | 0.568 | 0.577 |
| Purged block split | CNN | 0.547 | 0.510 | 0.544 |
| Chronological split | CNNam | 0.537 | 0.534 | 0.595 |
| Random split | CNNam | 0.583 | 0.588 | 0.577 |
| Purged block split | CNNam | 0.524 | 0.504 | 0.544 |
| *Panel C: label inside the chart window* | | | | |
| Chronological split | CNNam | 0.939 | 0.986 | 0.594 |

Notes: Chronological split: training 1962–1999, validation 2000–2007, test 2008–2024. Random split: all usable observations are shuffled and assigned 70%/10%/20% to training, validation and test. Purged block split: 21-day blocks are assigned at random in the same proportions, and training and validation observations within five days of a test observation are removed. In panel C the label is the sign of the five-day return ending on the last day shown in the chart.

### 8.3. *Benchmarks and loss functions*

Finally, accuracy and mean squared error are uninformative without a naive benchmark. In our test period an accuracy of 54% sounds respectable but equals that of a rule that ignores the data, and the recurrent networks in table 1 attain it by predicting 'up' almost every day. Similarly, the mean squared error of a probability forecast is the Brier score, and every model in our study is beaten by a constant. We recommend that image-based forecasting studies report the always-up and constant-probability benchmarks, a threshold-free measure such as the AUC, formal tests, results across several random seeds, and the split and label definitions in enough detail for readers to check for overlap.

## 9. Conclusion

We set out to learn what attention adds when a CNN reads candlestick charts of the S&P 500, and the evidence tells a consistent story. In a strictly chronological evaluation over 2008–2024, no model, whether CNNam, a plain CNN, a recurrent network or a logistic regression, beats the rule that always predicts an up day, and no AUC exceeds chance. What attention does add is calibration: CNNam's probabilities are significantly closer to the truth than the CNN's, although they are no better at separating up days from down days. The attention maps look intuitive, but they describe where the network looks, not what it knows. We also show how easily the same networks can be made to look predictive, by random splits with overlapping labels and by labels computed inside the chart window, and that purging overlapping observations removes the spurious gain. These findings do not contradict the evidence that image-based signals predict the cross-section of stock returns (Jiang *et al.* 2023, Murray *et al.* 2024). Ranking thousands of stocks against each other uses information that timing a single index cannot exploit, and the S&P 500 is among the most efficiently priced assets in the world. The results instead mark out where image-based models and attention are most likely to be useful: in the cross-section, in less efficient markets (Chen and Huang 2024), or at horizons where returns are more predictable. For any of these applications, the protocol in this paper, which consists of naive benchmarks, proper scoring rules, several seeds and a strict chronological or purged design, offers a way to tell a real signal from an artefact of the

evaluation. Our study has limitations. We consider one index, one chart format and one attention design. Other architectures, such as transformers on image patches, might extract information that ours cannot, although the absence of any signal in all of our models makes large gains unlikely. We also do not tune hyperparameters extensively, in order to avoid overfitting the validation period.


## Funding

The authors report that no funding was received for this research.


## Disclosure statement

The authors report there are no competing interests to declare.

## Data availability statement

The S&P 500 data are publicly available from Yahoo Finance (ticker `^GSPC`). The code that downloads the data, renders the charts, trains all models and produces every table and figure is available from the corresponding author upon request.

## References


Abu-Mostafa, Y.S. and Atiya, A.F., Introduction to financial forecasting. *Appl. Intell.*, 1996, **6**, 205–213.

Ahmadi, E., Abooie, M.H., Jasemi, M. and Zare Mehrjardi, Y., A nonlinear autoregressive model with exogenous variables neural network for stock market timing: The candlestick technical analysis. *Int. J. Eng.*, 2016, **29**, 1717–1725.

Ahmadi, E., Jasemi, M., Monplaisir, L., Nabavi, M.A., Mahmoodi, A. and Amini Jam, P., New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the Support Vector Machine and Heuristic Algorithms of Imperialist Competition and Genetic. *Expert Syst. Appl.*, 2018, **94**, 21–31.

Bahdanau, D., Cho, K. and Bengio, Y., Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015 (San Diego, CA).

Bailey, D.H., Borwein, J.M., López de Prado, M. and Zhu, Q.J., Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance. *Notices Amer. Math. Soc.*, 2014, **61**, 458–471.

Bhambu, A., Stock market prediction using deep learning techniques for short and long horizon. In *Soft Computing for Problem Solving*, Lecture Notes in Networks and Systems, pp. 121–135, 2023 (Springer: Singapore).

Brier, G.W., Verification of Forecasts Expressed in Terms of Probability. *Mon. Weather Rev.*, 1950, **78**, 1–3.

Bulkowski, T.N., *Encyclopedia of Candlestick Charts*, 2008 (John Wiley & Sons: Hoboken, NJ).

Chen, L., Pelger, M. and Zhu, J., Deep learning in asset pricing. *Manage. Sci.*, 2024, **70**, 714–750.

Chen, Y. and Hao, Y., A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Syst. Appl.*, 2017, **80**, 340–355.

Chen, Z. and Huang, W., Can candlestick charts help predict stock price movement? A machine learning approach. Technical report, SSRN Working Paper No. 5000007, 2024.

Christoffersen, P.F. and Diebold, F.X., Financial Asset Returns, Direction-of-Change Forecasting, and Volatility Dynamics. *Manage. Sci.*, 2006, **52**, 1273–1287.

Diebold, F.X. and Mariano, R.S., Comparing predictive accuracy. *J. Bus. Econ. Stat.*, 1995, **13**, 253–263.

Fama, E.F., Efficient Capital Markets: A Review of Theory and Empirical Work. *J. Finance*, 1970, **25**, 383–417.

Fengqian, D. and Chao, L., An adaptive financial trading system using deep reinforcement learning with candlestick decomposing features. *IEEE Access*, 2020, **8**, 63666–63678.

Gu, S., Kelly, B. and Xiu, D., Empirical asset pricing via machine learning. *Rev. Financ. Stud.*, 2020, **33**, 2223–2273.

Harvey, C.R., Liu, Y. and Zhu, H., . . . and the Cross-Section of Expected Returns. *Rev. Financ. Stud.*, 2016, **29**, 5–68.

Harvey, D., Leybourne, S. and Newbold, P., Testing the equality of prediction mean squared errors. *Int. J. Forecast.*, 1997, **13**, 281–291.

Hoseinzade, E. and Haratizadeh, S., CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Syst. Appl.*, 2019, **129**, 273–285.

Hou, K., Xue, C. and Zhang, L., Replicating Anomalies. *Rev. Financ. Stud.*, 2020, **33**, 2019–2133.

Hu, G., Hu, Y., Yang, K., Yu, Z., Sung, F., Zhang, Z., Xie, F., Liu, J., Robertson, N., Hospedales, T. and Miemie, Q., Deep stock representation learning: From candlestick charts to investment decisions. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2706–2710, 2018 (Calgary, Canada).

Jiang, J., Kelly, B. and Xiu, D., (Re-)Imag(in)ing Price Trends. *J. Finance*, 2023, **78**, 3193–3249.

Kamo, T. and Dagli, C., Hybrid approach to the Japanese candlestick method for financial forecasting. *Expert Syst. Appl.*, 2009, **36**, 5023–5030.

Kapoor, S. and Narayanan, A., Leakage and the Reproducibility Crisis in Machine-Learning-Based Science. *Patterns*, 2023, **4**, 100804.

Kaufman, S., Rosset, S., Perlich, C. and Stitelman, O., Leakage in Data Mining: Formulation, Detection, and Avoidance. *ACM Trans. Knowl. Discov. Data*, 2012, **6**, 1–21.

Kim, K.j. and Han, I., Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Syst. Appl.*, 2000, **19**, 125–132.

Lin, W.C., Wang, M.C., Tsai, C.F. and Hsu, J.P., Image-based deep learning models for stock predictions: Combining line, candlestick, and bar charts. *J. Forecast.*, 2026, **45**, 1350–1367.

Lo, A.W., Mamaysky, H. and Wang, J., Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *J. Finance*, 2000, **55**, 1705–1765.

López de Prado, M., *Advances in Financial Machine Learning*, 2018 (Wiley: Hoboken, NJ).

Marshall, B.R., Young, M.R. and Rose, L.C., Candlestick technical trading strategies: Can they create value for investors?. *J. Bank. Finance*, 2006, **30**, 2303–2323.

Martiny, K., Unsupervised discovery of significant candlestick patterns for forecasting security price movements. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval (KDIR)*, 2012 (SciTePress: Barcelona, Spain).

Mezghani, T. and Abbes, M.B., Forecast the role of GCC financial stress on oil market and GCC financial markets using convolutional neural networks. *Asia-Pac. Financ. Mark.*, 2023, **30**, 505–530.

Murray, S., Xia, Y. and Xiao, H., Charting by machines. *J. Financ. Econ.*, 2024, **153**, 103791.

Nison, S., *Japanese Candlestick Charting Techniques: A Contemporary Guide to the Ancient Investment Techniques of the Far East*, 2nd , 2001 (Prentice Hall Press: New York).

Niu, Z., Zhong, G. and Yu, H., A review on the attention mechanism of deep learning. *Neurocomputing*, 2021, **452**, 48–62.

Pesaran, M.H. and Timmermann, A., A simple nonparametric test of predictive performance. *J. Bus. Econ. Stat.*, 1992, **10**, 461–465.

Prado, H.A.d., Ferneda, E., Morais, L.C.R., Luiz, A.J.B. and Matsura, E., On the effectiveness of candlestick chart analysis for the Brazilian stock market. *Procedia Comput. Sci.*, 2013, **22**, 1136–1145.

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G. and Cottrell, G.W., A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2627–2633, 2017 (Melbourne, Australia).

Rasheed, J., Jamil, A., Hameed, A.A., Ilyas, M., Özyavaş, A. and Ajlouni, N., Improving stock prediction accuracy using CNN and LSTM. In *Proceedings of the 2020 International Conference on Data Analytics for Business and Industry (ICDABI)*, pp. 1–5, 2020 (Sakheer, Bahrain).

Sezer, O.B. and Ozbayoglu, A.M., Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Appl. Soft Comput.*, 2018, **70**, 525–538.

Tsai, C.F. and Quan, Z.Y., Stock prediction by searching for similarities in candlestick charts. *ACM Trans. Manage. Inf. Syst.*, 2014, **5**, 1–21.

Tudela, F., *The Secret Code of Japanese Candlesticks*, 2008 (John Wiley & Sons: Chichester).

Table A1. Implementation details.

| Item | Setting |
|---|---|
| Data | Yahoo Finance `^GSPC`, daily, 2 January 1962 to 3 September 2024 |
| Window and image size | 30 trading days; $64 \times 90$ pixels, 3 columns per day |
| Image layout | Price panel 51 rows, 1 blank row, volume panel 12 rows |
| Moving average | 20-day simple moving average of the close |
| Labels | $y_t = \mathbf{1}\{P_{t+h} > P_t\}$, $h \in \{1, 5\}$ |
| Split | Train 1962–1999, validation 2000–2007, test 2008–2024 |
| CNN blocks | Conv $5 \times 3$ (16, 32, 64 filters), batch norm, leaky ReLU, $2 \times 2$ max pool |
| Attention | Additive, $e_i = \mathbf{v}^\top \tanh(\mathbf{W}\mathbf{f}_i + \mathbf{b})$ over $8 \times 11$ positions, weights scaled by $N = 88$ |
| Classifier | Dropout 0.5, linear layer to one logit |
| Recurrent models | One layer, 64 hidden units; inputs OHLC and MA relative to last close, log volume |
| Logistic regression | 180 inputs (30 days $\times$ 6 features) |
| Loss and optimizer | Binary cross-entropy; Adam, learning rate $10^{-3}$, weight decay $10^{-5}$ |
| Batch size and stopping | 128; early stopping with patience 4 on validation loss, at most 30 epochs |
| Seeds | 5 (main results), 3 (robustness), 1 (section 8) |
| DM test | Brier loss, HLN correction, Newey–West lag $\max\{h-1, \lfloor 4(n/100)^{2/9} \rfloor\}$ |
| Bootstrap | Moving blocks of 20 days, 2,000 draws |


Unlu, E., RGBSticks: A new deep learning based framework for stock market analysis and prediction. *J. Soft Comput. Artif. Intell.*, 2020, **1**, 20–28.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 30, 2017 (Long Beach, CA).

Wagner, G.S. and Matheny, B.L., *Trading Applications of Japanese Candlestick Charting*, 1994 (John Wiley & Sons: New York).

Welch, I. and Goyal, A., A Comprehensive Look at the Empirical Performance of Equity Premium Prediction. *Rev. Financ. Stud.*, 2008, **21**, 1455–1508.


## Appendix A: Implementation details

Table A1 lists the settings used for all experiments. All models are implemented in PyTorch and trained on CPUs; one CNN or CNNam training run takes 5–10 minutes.