

# Baseline win rates for neural-network based trading algorithms

Andreas Krause  
Department of Economics  
University of Bath  
Bath, Great Britain  
mnsak@bath.ac.uk

Michael Fairbank  
School of Computer Science and Electronic Engineering  
University of Essex  
Colchester, Great Britain  
m.fairbank@essex.ac.uk

**Abstract**—Neural networks and other machine-learning systems are used to create automatic financial forecasting and trading systems. To aid comparison of such systems, there is a need for reliable performance metrics. One such metric that may be considered is the win rate. We show how in certain circumstances the win-rate statistic can be very misleading, and to counter this, we propose and define baseline win rates for comparison. We develop empirical and closed-form models for such baselines and validate them against financial data and a neural forecaster.

**Index Terms**—Financial Forecasting, Win rate, Neural Networks, Algorithmic Trading

## I. INTRODUCTION

Neural-network based automatic financial forecasting and trading systems have been widely studied [1]–[11]. A common way to assess their viability (along with that of other automatic trading/forecasting systems) is through the use of “win rates”, i. e. the fraction of forecasts that provide the correct sign. This is also referred to as the “hit rate”, “profitability percentage”, or the “accuracy” of the forecast, amongst many others variations of these names. If the win rate is high, it is then claimed that the trading scheme is “working” and thus profitable, see [12]–[14].

In this paper we demonstrate that relying on such win rates does not imply having found a profitable strategy, nor does it suffice for comparing one neural-forecasting system to another. In particular, when a neural trading system is developed that opens and closes trades based on the movement of market prices, there is scope for win rates to be very misleading. We demonstrate that fact in this paper by producing a simple baseline method that is capable of achieving a win rate over 80%, yet without having any intelligence at all, and without it being profitable.

Examples of when the timing of such strategies can skew the win rates include some very common basic order types, which are often included as sub-components of both human and automatic trading systems. These basic order types include stop-loss orders, where the position is closed if the price falls below a certain level; take-profit orders, where the position is closed if the price rises above a certain level; trailing stop-loss orders, where the position is closed if the price falls below a level determined from the maximum price reached

during the time the position is open; or trailing take-profit orders proposed by [13], where the position is closed if the price rises above a level determined from the minimum price reached during the time the position is open. These variable barriers also include more complex rules as found in many fully automated trading systems where positions are opened and closed according to trading indicators, or by decisions taken by a neural network.

To illustrate the point of arbitrarily skewed win rates, this paper focuses on trading systems that make use of basic stop-loss and take-profit orders, and also on financial forecasting systems which forecast over a fixed time horizon, leaving the consideration of more complex trading systems for future work.

It is well known by day traders that the win-rate alone is not the sole objective for trading, but needs at least balancing by the risk-reward ratio [15]. With [16] having shown empirically that arbitrarily high win rates can be generated by using stop-loss orders, we extend this result by deriving closed-form solutions for such win rates and demonstrate that despite high win rates no excess returns, beyond that of the market trend, can be generated. This provides backing for the emphasis in [17] that the actual profitability needs to be evaluated as not only the fraction of winning trades are important, but also the size of the profits made, balanced against the loss-making trades. If the latter are far bigger, the overall strategy may not be profitable, despite high win rates.

In a steadily rising or falling market, high win rates are particularly easy to obtain by following the market trend, without being able to outperform this trend; this has surely contributed to the high win rate of 79% in [18]. In this paper we will take into account the impact of rising or falling markets on win rates to exclude such biases, and produce a neural forecaster capable of an 78% win-rate due to market trend.

There are many potential pit-falls when evaluating and developing automated trading systems based on machine-learning methods. [19] and [20] describe lists of common ways that automated trading systems can fail. [21] describe ways that using back-tested results to evaluate trading systems can be extremely misleading, and [22] show how trading methods can fail once their mechanism becomes public knowledge. It

is the intention of our paper to highlight another potentially misleading way to evaluate neural trading systems; that of trusting quoted win rates without comparison baseline figures.

This paper derives closed-form solutions of the win rates and expected returns arising from a pure forecasting system in Section II, and for any trading system that uses stop-loss orders in Section III. Empirical results for both situations are included within those sections, tested on two different financial assets. Section IV concludes our findings.

## II. BASELINE WIN RATES AND RETURNS FOR FIXED LOOK-AHEAD PURE-FORECASTING MODELS

Let us consider a time series of prices  $P_t$ , with  $t = 1, \dots, T$ . The aim of a fixed look-ahead neural forecaster is to forecast the price change from time  $t$  to time  $t + \Delta t$ , where  $\Delta t$  is a constant. The actual “return” over this time period is defined by  $R_{t,\Delta t} = P_{t+\Delta t}/P_t - 1$ . We focus our attention here only on the prediction of the sign of the price movement, i. e. whether the price increases or decreases, and denote this forecast by  $\hat{y}_{t,\Delta t}$ , which is  $+1$  for an increase and  $-1$  for a decrease of the price. In this case the “win-rate” for our neural forecasting model, over horizon  $\Delta t$ , is given by

$$\hat{\pi}_{\Delta t} = Prob(R_{t,\Delta t}\hat{y}_{t,\Delta t} > 0), \quad (1)$$

and its average return is

$$\hat{R}_{\Delta t} = \mathbb{E}[R_{t,\Delta t}\hat{y}_{t,\Delta t}]. \quad (2)$$

where  $\mathbb{E}[\cdot]$  denotes expectation.

To form a simple benchmark to compare our neural forecaster against, assume that the benchmark system always forecasts a constant sign, i.e. it always forecasts  $+1$ , or always forecasts  $-1$ . To avoid potentially expensive false positives when rating a neural forecaster<sup>1</sup>, we furthermore assume that the overall trend of the time series is known to the benchmark forecaster. In this case we get the *baseline win-rate* to be

$$\bar{\pi}_{\Delta t} = \max(Prob(R_{t,\Delta t} > 0), Prob(R_{t,\Delta t} < 0)), \quad (3)$$

and *baseline return* to be

$$\bar{R}_{\Delta t} = |\mathbb{E}[R_{t,\Delta t}]|, \quad (4)$$

where the *max* and modulus signs occurring in these two equations have arisen out of our assumption that the baseline has the benefit of knowing in advance the overall trend of the time series.

These baselines (3) and (4) can be evaluated empirically by sampling  $R_{t,\Delta t}$  at every point of the time series. Again to avoid potentially expensive false positives, the baseline should be calculated and quoted separately for the test set. The performance of any neural forecasting system should be explicitly compared against these two baselines (3) and (4); in the same way that the base-rate of any classification system should be stated for comparison.

<sup>1</sup>For example, if by pure chance the neural forecaster happens to have a forecasting bias consistent with the actual market trend (as will be the case for a proportion of neural models), then the neural forecaster would look inappropriately effective.

### A. Approximations using Geometric Brownian Motion

If we assume that prices follow Geometric Brownian Motion (GBM), then it is possible to obtain closed-form expression for  $\bar{\pi}_{\Delta t}$  and  $\bar{R}_{\Delta t}$ .

For a price  $P$  moving randomly under GBM in continuous time, with constant drift rate  $\mu$ , and constant volatility  $\sigma$ , the GBM motion is defined as,

$$\frac{dP}{P} = \mu dt + \sigma dz, \quad (5)$$

where  $dz$  is a Wiener process [23]. From the properties of this stochastic process, it is possible to derive that

$$Prob(R_{t,\Delta t} > 0) = 1 - \Phi\left(-\frac{\hat{\mu}}{\hat{\sigma}}\right), \quad (6)$$

where  $\hat{\mu} = e^{\mu\Delta t} - 1$ ,  $\hat{\sigma}^2 = e^{2\mu\Delta t}(e^{\sigma^2\Delta t} - 1)$ , and  $\Phi(\cdot)$  denotes the cumulative standard normal distribution.  $\mu$  and  $\sigma$  are the mean and sample standard deviation of the step-wise returns,  $P_{t+1}/P_t - 1$ , calculated over the full range of the time series being studied,  $t = 1, \dots, T - 1$ .

Hence the baseline win-rate (3) simplifies to

$$\bar{\pi}_{\Delta t}^{GBM} = 1 - \Phi\left(-\frac{|\hat{\mu}|}{\hat{\sigma}}\right), \quad (7)$$

making use of the symmetry properties of the normal distribution, and the expected baseline return is

$$\bar{R}_{\Delta t}^{GBM} = |\hat{\mu}|. \quad (8)$$

### B. Comparison of actual and theoretical outcomes

We have derived the actual and analytical baseline win-rates for a neural fixed look-ahead forecaster. Here we compare these results with those of an actual time series to give an indication for the applicability of our results in assessing forecasting strategies. The asset chosen is the stock of Amazon, sampled at 1 minute intervals from 2 January 2015 to 29 December 2017, consisting of 293550 data points (Fig. 1). This dataset was obtained from Interactive Brokers [24]. The dataset was chosen because the prices show a strong and very persistent upwards trend, as confirmed by Table I. The data was partitioned into two halves: The first half (up to 1 July 2016) being the “training set”, and the second half being the “test set”.

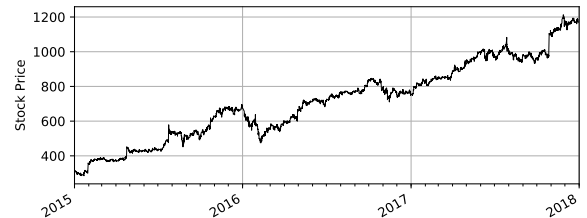


Fig. 1: Amazon prices

Using this setup we can now compare the results of our financial asset with that predicted from theory, with all results being calculated and quoted for the test set.

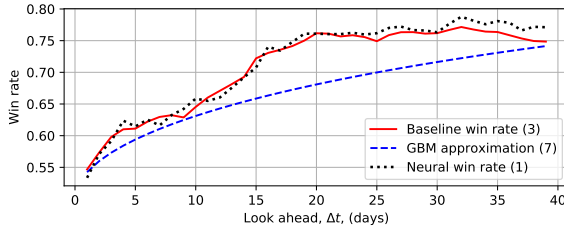
	Mean ( $\mu$ )	Volatility ( $\sigma$ )	Skewness	Kurtosis
EUR/USD	0.0541	52.7683	1.3136	255.8216
Amazon	0.4871	96.5738	45.0907	8513.4447

TABLE I: Descriptive statistics of the annualized returns of the chosen assets

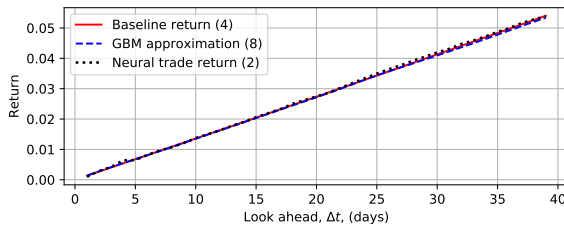
1) *Results for fixed look-ahead baseline win rate and returns:* Fig. 2 shows the baseline win rates and returns for fixed look-ahead forecasting of Amazon prices. These compare the actual baseline win-rates and returns, given by (3) and (4) respectively, with those approximated by the GBM model, (7) and (8), respectively. The graphs show the GBM equations make a close approximation to the empirically sampled values, over the full range of look ahead durations considered, and hence how the baseline values can be quickly worked out from the readily available  $\sigma$  and  $\mu$  dataset statistics.

The results also show that high win rates and returns are possible if the benchmarking is on a sufficiently strongly trending stock, and if the forecasting period  $\Delta t$  is sufficiently large.

This has enabled baseline win-rates for the trivial benchmark strategy to reach as high as nearly 77% for Amazon forecasting, with  $\Delta t = 32$  days for the test set, and returns that steadily increase with the increasing  $\Delta t$  value.



(a) Win rates for Amazon



(b) Returns for Amazon

Fig. 2: Baseline fixed look-ahead win rates and returns, with results for a neural-network fixed look-ahead forecaster, for Amazon test set (Jul 2016-Dec 2017). One trade is opened every minute, and closed at  $\Delta t$  days later.

2) *Neural forecasting results:* To demonstrate the practicality of the benchmark figures, we created a neural-network forecaster. The neural network was a feed-forward, fully connected multilayer-perceptron network, with 8 inputs, 1 hidden layer of 10 inputs, and 2 output nodes. All hidden nodes used the *tanh* activation function, and the output layer used

a softmax probability function. The network was trained with the cross-entropy loss function, with added L2 regularization (weight decay) on all neural weights, to prevent overfitting.

The neural network received inputs of the most recent 8 daily returns of the asset’s end-of-day closing prices, and was tasked to forecast the sign of the future  $\Delta t$ -day price return, for various different values of  $\Delta t$ .

The network was trained using the training set, and results are reported for the (out-of-sample) test set. Results are shown as the black dotted curves in Fig. 2. The network achieved a win-rate of 78%, when  $\Delta t = 32$  days, with significant positive return (up to 5% growth per 40 days). Although these results seem superficially impressive, the baseline win-rate for this task given by (3) is 77% (from Fig 2), and the neural returns are almost indistinguishable from the baseline figures. This indicates the neural results are not significant, and most likely that the neural network is not advising anything more profound than “buy and hold”.

### III. THEORETICAL WIN RATES OF A BENCHMARK TRADING STRATEGY WITH STOP-LOSSES

The previous section described the win-rates for a pure forecasting system, i.e. without any trades being actually made, and in the advantageous situation of trying to forecast in a strongly trending market. However many neural financial systems go further by simulating the opening and closing of trades, and when this happens, the closing strategy may involve stop-loss orders or similarly related order types. In this case it is possible to artificially skew the win rates even if the time series is not trending strongly, and hence baseline win rates are particularly important.

For the sake of generating a baseline trading strategy which uses stop-losses, let us define an upper price barrier  $\bar{P}$  and a lower price barrier  $\underline{P}$  such that with the current price  $P_t$  we have  $0 < \underline{P} < P_t < \bar{P} < +\infty$ . Having bought the asset at price  $P_t$  the simple trading strategy is to sell as soon as either of the price barriers is breached, i. e. as soon as the latest price,  $P_{t'}$ , satisfies either  $P_{t'} > \bar{P}$  or  $P_{t'} < \underline{P}$ , for some  $t' > t$ . Thus the position would be liquidated at a loss if the lower barrier is breached, and at a profit if the upper barrier is breached. For the sake of simplicity of exposition, short positions in the asset are not allowed.

Hence, the time it takes to breach the upper and lower barrier respectively, can be defined as

$$\begin{aligned} \bar{\tau} &= \inf \{t' > t | P_{t'} > \bar{P}\} - t, \\ \underline{\tau} &= \inf \{t' > t | P_{t'} < \underline{P}\} - t, \end{aligned} \quad (9)$$

and the time until either barrier is breached is

$$\tau = \min \{\bar{\tau}, \underline{\tau}\}. \quad (10)$$

As our trading strategy is to buy the asset and then sell it if one of these barriers is breached, then the win rate will be the fraction of trades that hit the upper barrier first,

$$\bar{\pi}_b = Prob(\bar{\tau} < \underline{\tau}), \quad (11)$$

where the  $b$  subscript is to indicate barriers are present.

Define for convenience  $\bar{\Delta}$  and  $\underline{\Delta}$  such that

$$\begin{aligned}\bar{\Delta} &= \bar{P}/P_t, \\ \underline{\Delta} &= \underline{P}/P_t,\end{aligned}\quad (12)$$

with  $0 < \underline{\Delta} < 1 < \bar{\Delta} \leq +\infty$ . Then, using the trading strategy outlined above, and assuming that trades are opened with zero intelligence (i.e. with completely random timings), the expected ratio of closing to opening price is given by

$$\mathbb{E}[P_{t+\tau}/P_t] = \bar{\pi}_b \bar{\Delta} + (1 - \bar{\pi}_b) \underline{\Delta}, \quad (13)$$

where  $P_{t+\tau}$  is the price at which the position was closed.

Equation (13) is justified since the right-hand side evaluates the expectation for the two possible outcomes: each position must be closed with either a win (with probability  $\bar{\pi}_b$ ; resulting in  $P_{t+\tau}/P_t = \bar{\Delta}$ ) or a loss (with probability  $1 - \bar{\pi}_b$ ; resulting in  $P_{t+\tau}/P_t = \underline{\Delta}$ ).

Note that generally, the left-hand side of (13) will be a property of the time-series being assessed or trained on, and upon the choices of  $\underline{\Delta}$  and  $\bar{\Delta}$ .<sup>2</sup>

Equation (13) can be rearranged to obtain an explicit expression for the win rate:

$$\bar{\pi}_b = \frac{\mathbb{E}[P_{t+\tau}/P_t] - \underline{\Delta}}{\bar{\Delta} - \underline{\Delta}}. \quad (14)$$

This shows us that the win rate,  $\bar{\pi}_b$ , can be determined in advance from the value of  $\mathbb{E}[P_{t+\tau}/P_t]$  and on the choices of the values of the price barriers,  $\underline{\Delta}$  and  $\bar{\Delta}$ . It also shows that the win rate can be artificially changed, simply by choosing  $\underline{\Delta}$  and  $\bar{\Delta}$  carefully.

The average price return of this trading system is defined by

$$\bar{R}_b = \mathbb{E}[P_{t+\tau}/P_t] - 1. \quad (15)$$

When considering high-frequency trading (which will arise under this trading scenario whenever the barriers are placed sufficiently close to the opening price), then  $\tau$  will be small and thus the time for the market trend to have any effect is negligible, so that  $\bar{R}_b \approx 0$ . In this case (14) can be approximated by

$$\bar{\pi}_b \approx \frac{1 - \underline{\Delta}}{\bar{\Delta} - \underline{\Delta}}. \quad (16)$$

It may also be considered that in the absence of any further knowledge about future market movements, then it could be assumed that  $\bar{R}_b \approx 0$ , making (16) more appropriate than (14).

This approximation for the baseline win-rate, (16), shows that we can get a win rate arbitrarily close to 1 by simply setting  $\bar{\Delta}$  close to 1, i.e. the upper barrier will be set close to the current price, and  $\underline{\Delta}$  close to zero, i.e. the lower barrier far removed from the current price. In this case the lower barrier is hardly ever reached, thus losses rarely realized, even though any realization would be large, while the upper barrier is frequently reached, resulting in a large number of

<sup>2</sup>For example, if the time series is strongly upwardly trending, and  $\underline{\Delta}$  and  $\bar{\Delta}$  are chosen such that the holding period  $\tau$  is reasonably long, then it is likely that  $\mathbb{E}[P_{t+\tau}/P_t] > 1$ .

profitable trades, although each individual profit will be small. Furthermore, to compound this problem further in practical trading situations when  $\bar{\Delta} \approx 1$ , the many tiny winning trades would be dwarfed by their transaction costs of trading.

In all practical circumstances of training neural networks on historical price data, the expression for  $\mathbb{E}[P_{t+\tau}/P_t]$  used in (14) should be sampled from the entire set of data being evaluated (i.e. usually just the test set), evaluating  $\tau$  at each time step via (10).

Again, for reasons of avoiding potentially expensive false positives, if the market is down-trending, then the long-only trading baseline system described in this section should be flipped into a short-only trading system.

#### A. Approximation for win rate with barriers, using Geometric Brownian Motion

To avoid the need to empirically sample  $\mathbb{E}[P_{t+\tau}/P_t]$  in (14), if we again assume that prices follow GBM then it is possible to obtain a closed-form expression for  $\bar{\pi}_b$  directly, as follows. We know from [23, Ch. 3] that the probability of the price in (5) reaching the upper barrier first, under the GBM approximation, is given by

$$\bar{\pi}_b^{GBM} = \begin{cases} \frac{1 - \underline{\Delta}^\xi}{\bar{\Delta}^\xi - \underline{\Delta}^\xi} & \text{if } \mu \neq \frac{1}{2}\sigma^2 \\ \frac{\ln \underline{\Delta}}{\ln \underline{\Delta} - \ln \bar{\Delta}} & \text{if } \mu = \frac{1}{2}\sigma^2 \end{cases}, \quad (17)$$

where  $\xi = \frac{\sigma^2 - 2\mu}{\sigma^2}$ . This formula (17) is a GBM-alternative to (14), giving a closed-form win rate in terms of  $\mu$  and  $\sigma$ . The constants  $\mu$  and  $\sigma$  would be sampled from the step-wise returns, specifically for the test set, as described in Section II-A.

#### B. Empirical results for trading with barriers

We have derived the win rate when trading is conducted such that the asset is sold as a fixed upper or lower barrier is reached. Here we validate those baseline win rates.

In these experiment, we chose the exchange rate between the EURO and US Dollar (EUR/USD), sampled in 1 minute intervals from 4 January 2016 to 29 December 2017, consisting of 725725 data points, obtained from Interactive Brokers [24]. This dataset was chosen as an example of a dataset which does not show such any particularly strong trend, in contrast to the previous example. As can be seen from Fig. 3, this asset exhibits very different properties from Amazon, and is much less consistently trending during the time period in question.

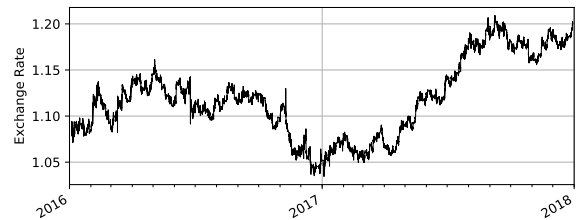


Fig. 3: EUR/USD Exchange Rate

We ran the baseline trading strategy from Section III for a range of barriers, that are up to 2.5% away from the price at which we open the position either side, on the whole EUR/USD dataset. In the simulation, a new trade of fixed size was made at every data point in the time series and closed according to the strategy. Any positions that were not closed at the end of our sample period were disregarded from the analysis.

At the same time as running this trading simulation, we numerically calculate the sample estimate of the win rate  $\bar{\pi}_b$ , so that it can be compared to its theoretical approximate value given by (16) and (17). The values of  $\mu$  and  $\sigma$ , referred to in (17) were calculated as described in Section III-A, using the full date range and dataset shown in Fig. 3. These same values are quoted in their annualised form in Table I.

We see from Fig. 4 that the actual win rates follow the theoretical win rates closely, even though a small gap persists. This arises from the non-normality of returns, in particular due to the high kurtosis shown in Table I, which somewhat violates the assumptions required for true GBM.

From the approximations for  $\bar{\pi}_b$ , i.e. (16) and (17), the closest match for the asset arises with (16), where we assumed a zero drift rate of the stochastic process. While this is clearly not exactly true, the skewness and kurtosis of this asset will compensate for this effect. While the match between our asset and the theoretical results are not perfect, they show a clear indication that using GBM is a valid approximation of actual win rates as the observed patterns are identical.

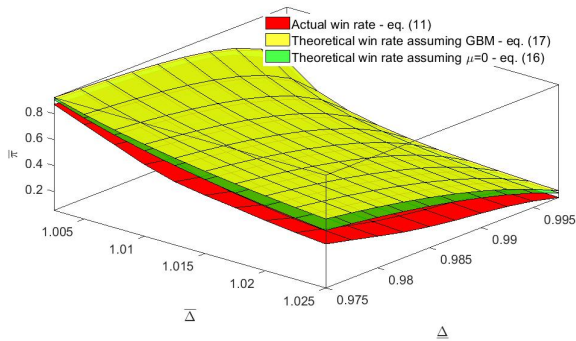


Fig. 4: EUR/USD trading win rates, under varying price barriers. One trade is opened every minute, and held open for duration  $\tau$  given by (10) and (9).

Another observation we can make is that the win rate can be very high. In our parameter settings it can go well above 80%. This is achieved whenever the upper barrier is set close to the value of the asset when opening a position and the lower barrier is set far below this value. In this case the much closer upper barrier will be breached much more frequently than the further removed lower barrier and we thus observe a large number of profitable trades. This high win rate has been attained despite the asset not showing any consistent price trend.

Even if we observe a large number of profitable trades, these profits will be small given the closeness of the upper barrier, while any losses from breaching the lower barrier will be much larger. We further examined the returns generated during the baseline strategy, but it did not turn out that any of the parameters  $\bar{\Delta}$  or  $\underline{\Delta}$  considered in this experiment led to a significantly profitable system; beyond the returns that would have been generated by a buy-and-hold strategy. This conclusion is consistent with what would be expected from such a zero-intelligence baseline strategy.

#### IV. CONCLUSIONS

We have demonstrated that win rates for neural trading systems can become artificially high by appropriate positioning of stop-loss orders. Similarly for pure-forecasting systems, the win-rate can be misleadingly high when considering a test dataset that is strongly trending. For both of these situations we have defined baseline win rates which should be stated for comparison in any published win rates. We have also defined closed-form approximations for the baseline win rates, in terms of  $\mu$  and  $\sigma$  for the time-series' step-wise returns, under the assumption of Geometric Brownian Motion (GBM). We have demonstrated the validity of the GBM approximations using actual asset prices, and have demonstrated the purpose of the baseline figures against a neural forecasting system.

Our empirical results achieved win rates approaching 80% from zero-intelligence strategies. Hence we are sceptical about the use of win rates to assess neural forecasting and trading strategies with a high degree of accuracy (high win rates) as the basis of a trading strategy. In particular, win rates cannot be used to compare one neural forecasting system or trading system against another.

We advise that any presentation of win rates would be incomplete without publishing a baseline win rate alongside. We have described how to empirically sample a baseline win rate for pure fixed-look ahead forecasting models (3), and for trading systems that use stop loss-orders (14). The closed-form approximations for these win-rates are given by (7), (16) and (17).

When stop-loss orders are used, it is very easy to obtain a high win rate, regardless of any trend direction, as evidenced here, by using an upper barrier close to the starting price and a lower barrier further removed and selling the asset when either is breached. The frequent but small profits at the upper barrier are matched by less frequent but larger losses at the lower barrier. Hence using any trading strategy involving asymmetric barriers to close and/or open positions can easily be manipulated to show high win rates.

Our results suggest that for a trading system with stop-loss orders, the expected returns given by (13) are harder to fake than the win-rates, and thus make a better metric for comparing trading systems than win rates. Furthermore, any forecast can also not be deemed accurate if the win rate does not exceed the given baseline figure.

Also, we have advised that the sampling and calculation of benchmark win rates is done separately for the testing set from

the training set; so as to make it as challenging as possible for a neural system to beat the benchmark (so that the user is not tricked into trusting false results). Similarly, we have argued that the benchmark should be given the advantage of knowing the overall market-trend direction.

The trading system considered in this paper has focussed on fixed stop-loss barriers and fixed take-profit orders. We could extend our work in numerous ways to include more sophisticated orders, such as trailing stop losses, strategies that involve positions of differing sizes depending on previous outcomes, amongst many others. We are confident that the fallacy of relying on win rates for the basis of a trading strategy remains valid, but leave future research to provide evidence to this effect.

## REFERENCES

- [1] D. S. P. Salazar, P. J. L. Adeodato, and A. L. Arnaud, "Continuous dynamical combination of short and long-term forecasts for nonstationary time series," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 241–246, 2014.
- [2] J. Moody and M. Saffell, "Learning to trade via direct reinforcement," *IEEE transactions on neural networks*, vol. 12, no. 4, pp. 875–889, 2001.
- [3] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 653–664, 2017.
- [4] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity, "Detecting wash trade in financial market using digraphs and dynamic programming," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 11, pp. 2351–2363, 2016.
- [5] S. Raudys, "Portfolio of automated trading systems: Complexity and learning set size issues," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 3, pp. 448–459, 2013.
- [6] K. K. Ang and C. Quek, "Stock trading using RSPOP: A novel rough set-based neuro-fuzzy approach," *IEEE Trans. Neural Networks*, vol. 17, no. 5, pp. 1301–1315, 2006.
- [7] M. A. Dempster, T. W. Payne, Y. Romahi, and G. W. Thompson, "Computational learning techniques for intraday FX trading using popular technical indicators," *IEEE Transactions on neural networks*, vol. 12, no. 4, pp. 744–754, 2001.
- [8] P. Tino, C. Schittenkopf, and G. Dorffner, "Financial volatility trading using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 865–874, 2001.
- [9] J. Doering, M. Fairbank, and S. Markose, "Convolutional neural networks applied to high-frequency market microstructure forecasting," in *Computer Science and Electronic Engineering (CEECE), 2017*. IEEE, 2017, pp. 31–36.
- [10] S. Kim and M. Kang, "Financial series prediction using attention lstm," *arXiv preprint arXiv:1902.10877*, 2019.
- [11] R. M. I. Kusuma, T.-T. Ho, W.-C. Kao, Y.-Y. Ou, and K.-L. Hua, "Using deep learning neural networks and candlestick chart representation to predict stock market," *arXiv preprint arXiv:1903.12258*, 2019.
- [12] H. G. Zimmermann, L. Bertolini, R. Grothmann, A. M. Schäfer, and C. Tietz, "A technical trading indicator based on dynamical consistent neural networks," in *International Conference on Artificial Neural Networks*. Springer Verlag, 2006, pp. 654–663.
- [13] A. Bakhach, E. Tsang, W. L. Ng, and C. V. L. Raju, "Backlash agent: a trading strategy based on directional change," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–9.
- [14] M. A. H. Dempster and V. Leemans, "An automated FX trading system using adaptive reinforcement learning," *Expert Systems with Applications*, vol. 30, no. 3, pp. 543–552, 2006.
- [15] C. Mitchell, "Day trade using win rate and risk/reward ratios," 2020, <https://www.thebalance.com/day-trade-better-using-win-rate-and-risk-reward-ratios-1031457>, accessed: 2020-03-31.
- [16] C. Rosenbloom, "Revisiting stop-loss and profit target affect on win rate," 2008, <http://blog.afraidtotrade.com/revisiting-stop-loss-and-profit-target-affect-on-win-rate/>, accessed: 2018-06-01.
- [17] G. Cheng, "The myth of profit/loss ratios," 2007, [https://www.investopedia.com/articles/forex/07/profit\\_loss.asp](https://www.investopedia.com/articles/forex/07/profit_loss.asp), accessed: 2018-06-01.
- [18] Y. Dai and Y. Zhang, "Machine learning in stock price trend forecasting," 2013, working Paper, Stanford University.
- [19] M. Lopez de Prado, "The 10 reasons most machine learning funds fail," *Journal of Portfolio Management*, vol. forthcoming, 2018.
- [20] E. Hurwitz and T. Marwala, "Common mistakes when applying computational intelligence and machine learning to stock market modelling," 2012, arXiv:1208.4429 [stat.AP].
- [21] F. J. Fabozzi and M. Lopez de Prado, "Being honest in backtesting reporting: A template for reporting multiple tests," *Journal of Portfolio Management*, vol. forthcoming, 2018.
- [22] R. D. McLean and J. Pontiff, "Does academic research destroy stock return predictability?" *Journal of Finance*, vol. 71, no. 1, pp. 5–32, 2016.
- [23] A. K. Dixit and R. S. Pindyck, *Investment under Uncertainty*. Princeton, NJ: Princeton University Press, 1994.
- [24] Interactive Brokers, <https://www.interactivebrokers.com>.