

# Applications of Deep Learning and Machine Learning in Trading

Hongxuan Paul Liu

Department of Mathematics, The University of Hong Kong, Hong Kong, China

liupaul@connect.hku.hk

**Abstract.** This review paper investigates applications of machine learning and deep learning in trading, with a particular emphasis on recent advances in deep learning. It provides an overview of algorithms, including support vector machines (SVMs), random forests, deep neural networks (DNNs), long short-term memory networks (LSTM networks), and deep reinforcement learning (DRL). Findings show that while machine learning and deep learning models were able to surpass traditional strategies in general in terms of profitability, they were also better at risk management. However, despite their performances showing superiority, their performances varied significantly under different market conditions, including markets during periods of high and low volatility. In particular, LSTM networks and random forests can generate substantial returns, higher Sharpe ratios, and lower drawdowns compared to the benchmarks, whereas DNNs struggled during highly volatile periods, as reflected in the returns in the periods. Moreover, the improved DRL agent TradeNet-CR can manage risk significantly better than another, despite not surpassing the original TradeNet-CR model.

**Keywords:** Deep learning, Machine learning, forecasting, stocks.

## 1. Introduction

Accurately forecasting stock prices has long been an important goal for many researchers. Detailed analysis of market trends is necessary to minimize losses and maximize profits. Traditional strategies often rely on explicit pattern recognition of past price data, for example, support and resistance levels or breakout patterns [1]. Other rule-based strategies are based on indicators, such as moving average crossovers and using moving averages as supports or resistances [1]. Market price fluctuations are influenced by many factors, for instance a variety of firm characteristics, while simple strategies are only restricted to a couple of them and thus may not effectively forecast the future [1].

With the development of technology and computational power, machine learning and deep learning are applied to financial markets due to their capability to capture hidden and complex patterns and handle non-linear relationships and unorganized data, such as news and content in social media [1]. At the same time, algorithmic trading—computer programs that implement these strategies—has evolved to perform faster and is used more frequently in trading, thereby reducing arbitrage opportunities and increasing volatility [2]. It has greatly affected price forecasting, supported by Chaboud and his team, in that the increasing future uncertainty simultaneously occurs with more active algorithmic trading [2].

This review paper provides an overview of recent developments in the application of machine learning and deep learning in stock price forecasting, with more weightings on deep learning.

## 2. Literature Review

### 2.1 Machine Learning

Various machine learning algorithms have been applied to analyze history and predict the trends. This section will look at some machine learning algorithms.

#### 2.1.1 Support Vector Machine/SVM

SVM is simply the solution to the convex quadratic problem:

$$\text{Min } \frac{1}{2}w^T w + C\xi^T 1 \quad \text{subject to } D(\Phi w - b1) \geq 1 - \xi \quad (1)$$

Where  $b \in R, w \in R^q, \xi \geq 0$ ;  $C \in R^+$  is the penalty for misclassifying data, and it is a predefined parameter,  $\xi$  is a vector consisting of slack variables,  $D$  is the diagonal matrix of labels for each data point, and  $\Phi$  is the matrix of nonlinear mapping between all  $x_i$  and  $x_j$  [3].

Peng and his team collected data from the Dow Jones Industrial Average, EURO STOXX 50, and Bovespa, spanning from April 2021 to September 2022 [3]. SVM was applied to forecast the price direction movement in the time interval  $[t, t+1]$ . At the same time, the independent variables are the nine technical analysis indicators—simple moving average, exponential moving average, moving average convergence divergence, momentum, rate of change, on balance volume, relative strength index, stochastic K%, and William's R% [3]. Lastly, 10-fold cross-validation was applied to evaluate the robustness of the model on the training set [3].

A benchmark was set to be the buy-and-hold strategy; that is, the stock is bought and forgotten, so the profit is equivalent to the changes in the market [3]. Their experiments on the American, European, and Brazilian markets showed that SVM performed better during the war than in pre-war periods. Around 95% of models outperformed the benchmark in the American and European markets. However, this number drops below 75% during the pre-war period [3].

### 2.1.2 Random Forest

Another common machine-learning model is the random forest. Random forests are simply a group of decision trees, and their predictions are combined using various methods.

Researchers trained it on the lagged returns of stocks from the S&P 500 to predict the likelihood of individual stocks beating the overall market, ranging from December 1992 to October 2015. The trading strategy applied will choose the top k number of stocks to buy and short the worst k stocks [4]. Despite not being able to outperform some of the other models (ensembles), its feature selection process ensures no overfitting and makes it more stable in noisy space; thus, the performance is the best among the base learners [4]. Moreover, the strategy described above based on the random forest had the lowest risk; its maximum drawdown is 67%, while other models exceeded 70% [4]. The Sharpe ratio of the random forest after transaction costs is around 1.9, putting it the highest among all [4].

The studies showed certain challenges regarding the applications of machine learning. Machine learning algorithms cannot perform consistently in different types of market conditions. Krauss et al. illustrate that random forests, ensembles, and gradient-boosted trees have deteriorated in recent years, and the trends will continue [4]. Peng et al. show that support vector machines were able to perform better in more volatile markets [3]. The trend also indicates that machine learning models are less capable of arbitrating profits due to the increasing popularity of the algorithms and increasing computational power.

## 2.2 Deep Learning

Deep learning is a type of representation learning characterized by multiple levels of representation [5]. The architecture is composed of modules stacking together, most of which have to improve themselves via learning [5]. These multilayer structures can learn from data and optimize themselves using backpropagation; it is a process to find the gradient of the objective function with respect to the parameters and backpropagate to the previous layers using the chain rule [5]. For example, convolutional neural networks (ConvNets) are designed to handle data in the form of arrays [5]. This can be applied to image recognition [5].

### 2.2.1 Application for Deep Learning in Financial Markets

#### 2.2.1.1 Long Short-Term Memory Networks (LSTM Networks)

LSTM networks are under the group of recurrent neural networks, and they can study long-term dependencies in datasets and solve the previously inherent problems of recurrent neural networks [6]. LSTM networks are like the usual neural networks; they have an input and output layer and a certain

number of hidden layers [6]. However, their hidden layers are comprised of memory cells, which are designed to store data from previous time steps [6].

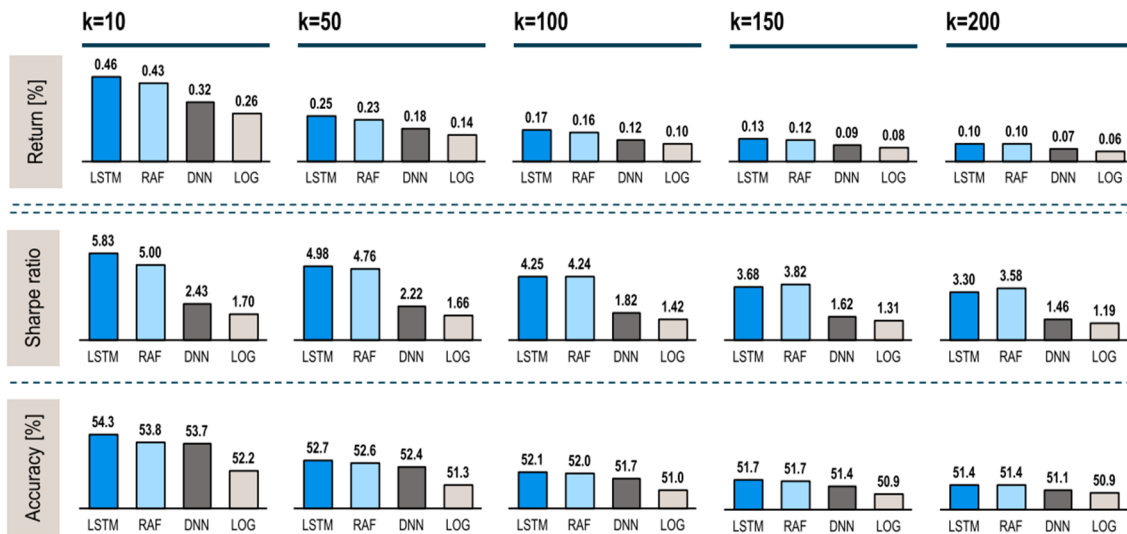


Figure 1: Return rate, Sharpe ratio, and accuracy before transaction costs of the models. From “Deep learning with long short-term memory networks for financial market predictions,” by Fischer, T., & Krauss, C.

Like the strategy based on random forests, among 2k stocks, the strategy will go long on the top k stocks and short on the worst k stocks [6]. Figure 1 shows that in terms of returns per day, LSTM networks generally outperform common machine learning models and deep learning models at 0.26% per day after transaction costs; for instance, they have used random forests, logistic regression, and deep neural networks [6]. Regarding the risk metric of standard deviation, both LSTM networks and random forests exhibited lower values than the other two models [6]. From the Diebold and Mariano test and Pesaran-Timmermann test, LSTM was at a 5% significance level superior to the three models in terms of forecasting accuracy and shows statistically significant predictive accuracy, as shown in Figure 1 [6]. Lastly, the Sharpe ratio after commission fees for LSTM ranked at the top with around 2.3365. Random forests immediately follow with a Sharpe ratio of 1.8657 [6].

Researchers noticed that LSTM performed exceptionally in the early times, from January 1993 to December 2000, converting 1 USD into 11 USD [6]. However, from January 2001 to December 2009, the momentum reduced and even performed worse than the random forests [6]. The profit continued to deteriorate after 2010 until late 2015, when the returns of LSTM were only able to fluctuate around zero, with random forests losing money [6].

### 2.2.1.2 Deep Neural networks (DNN)

Deep neural networks are just like artificial neural networks except that they have more hidden layers. The study opted for a 31-31-10-5-2 architecture; there are 31 input nodes and 2 output nodes [4]. Like random forests, the same strategy is applied to the lagged returns of stocks in the S&P 500 [4]. For each stock, the model will forecast the probability of outperforming the general market and will output a binary response variable, which is 1 if the stock’s return is higher than the cross-sectional median return of all stocks and 0 otherwise [4].

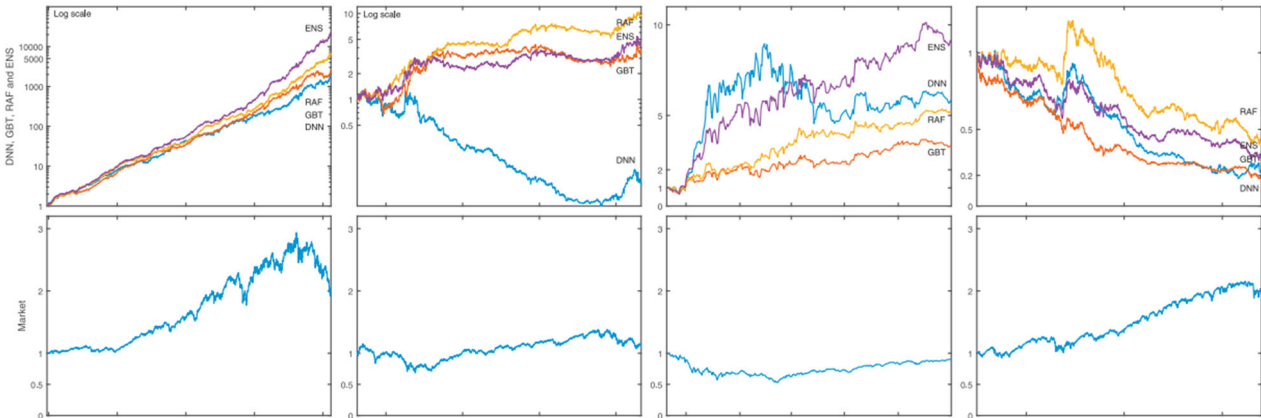


Figure 2: Performance of DNN, gradient-boosted trees, random forests, and ensembles at periods 12/1992-03/2001, 04/2001-08/2008, 09/2008-12/2009, and 01/2010-10/2015. From “Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500,” by Krauss, C., Do, X. A., & Huck, N.

However, compared to other models, such as random forests, gradient-boosted trees, and ensembles, DNN has the worst performance [4]. Its average daily returns prior to and after transaction costs are 0.33% and 0.13%, respectively [4]. The directional accuracy was ranked the lowest, at around 54% [4]. Moreover, it has demonstrated the highest drawdown of 95% after transaction costs [4]. As shown in Figure 2, researchers noticed that DNN performed the worst during the 2008 financial crisis, while other models were able to generate profits [4]. By changing the number of nodes in the first hidden layer, the study showed that it does not affect the overall performance of the strategy; however, the returns drop significantly if one of the hidden layers is removed, resulting in dropout regularization and choosing tanh as the activation function [4].

### 2.2.2 Deep Reinforcement Learning (DRL)

Like usual reinforcement learning, the learning process in deep reinforcement learning is sequential and aims to maximize its total rewards while using deep learning techniques [7]. At each time step, the agent looks for its trading position and OHLCV stock data [7]. At time step  $t$ , let  $\pi(a_t|h_t)$  be the payoff of the agent choosing action  $a_t$  given historical data  $h_t$  [7]. Action of agent is defined as the number of shares bought by the agent  $a_t = Q_t$ ,  $Q_t = Q_t^{long}$  represents maximizing the number of shares owned by converting from cash value, and  $Q_t = Q_t^{short}$  converts share value into cash [7]. Two constraints are imposed to ensure the cash value the agent holds must be positive and sufficient to cover losses when short selling [7]. Lastly, rewards are defined as the daily returns [7].

The model Trading Deep Q-Network algorithm (TDQN) is proposed [7]. Modifications and improvements were made to the DQN algorithm: the CNN was replaced by a feedforward DNN; double DQN; the classical RMSProp optimizer was replaced by the ADAM optimizer; the Huber loss instead of MSE; and more [7].

Four benchmarks are introduced to assess the performance of the model; they are “buy and hold,” “sell and hold,” “trend following with moving averages,” and “mean reversion with moving averages” [7].

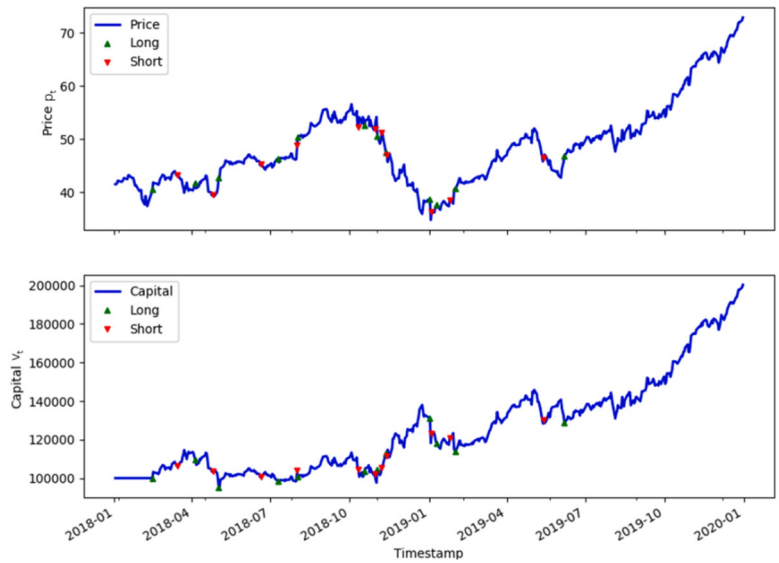


Figure 3: TDQN algorithm for Apple stock in the test set.

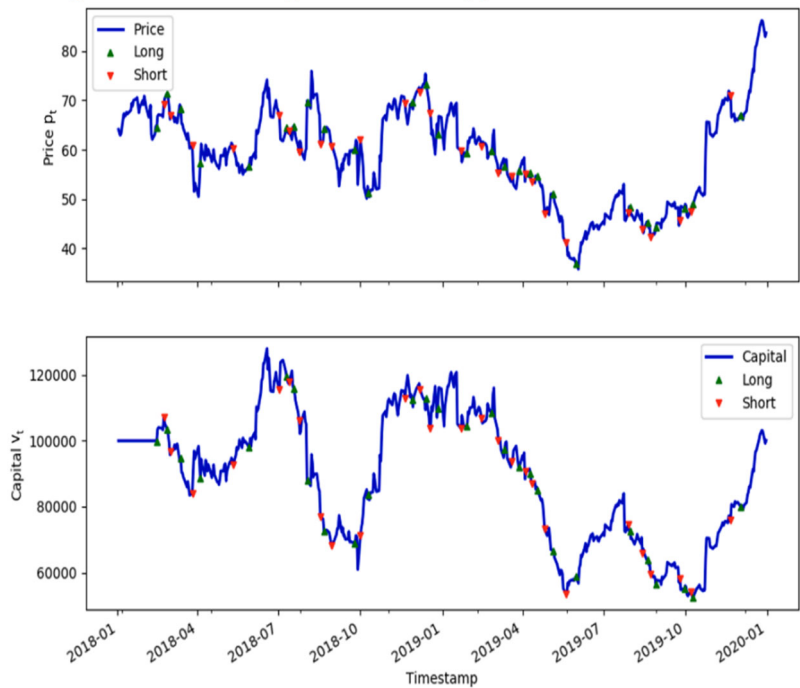


Figure 4: TDQN algorithm for Tesla stock in the test set.

Out of the many stocks tested, the team focuses on Apple stock and Tesla stock [7]. The test of TDQN on the Apple stock showed promising results in terms of annualized return and risks by outperforming the four benchmarks; its Sharpe ratio is 1.484, the annualized return is 32.81%, and the maximum drawdown is 17.31% [7]. While it can benefit well from major trends, it struggles to perform well in markets with increasing volatility, as shown in Figure 3 [7]. The team observed that the model is less proactive and more reactive; however, its volatility experiences a sudden increase before a trend inversion, as shown in Figure 3 [7].

On the other hand, the model did poorly on Tesla stock [7]. The team suggests that this was due to its high volatility, which we see is the case in Apple stock [7]. From Figure 4, although it maintains a positive annual return rate, it is only about 13% and is inferior to the “buy and hold” and “mean reversion with moving averages” strategies [7]. Moreover, its maximum drawdown is nearly 60%, which does not make it a practical strategy to deploy [7]. The differences in Sharpe ratio during training and testing indicate overfitting; the model’s Sharpe ratio tops out at around 2.5 during training but only 0.5 when testing [7].

On average, the Sharpe ratio of the strategy on all of the stock is 0.404, which barely surpasses the buy-and-hold strategy with a Sharpe ratio of 0.369 [7]. The team also noted that the TDQN algorithm performs close to the buy-and-hold strategy and the sell-and-hold strategy for a number of stocks [7]. In this subsection, we will still use deep reinforcement learning [8]. The team proposes an extension to the model by other researchers, TradeNet-CR. It consists of a DRL agent, called TraderNet, with a rewarding function, called Round-Trip Strategy [8]. They have modified the Round-Trip Strategy to merge both market and limit orders [8]. Moreover, the model also includes two other modules [8]. Firstly, there is a rule-based safety mechanism called N-Consecutive; its purpose is to assess if the action chosen by the agent is safe to execute [8]. The last module is named Smurf, another safety mechanism. It trains the DRL agent and avoids risky trading activity [8].

The team tweaks the Round-Trip Strategy so that the agent can open a round-trip (buy-sell round or sell-buy round) using market orders and close the orders with limit orders in a fixed number of time steps, and instead it will maximize the cumulative sum of the logarithmic PNL percentage returns [8]. Smurfing is introduced to manage risks better [8]. Smurf is another agent trained similarly to TraderNet, with higher transaction fees, so Smurf is a conservative trader [8]. Although the original TraderNet outperformed any of its improvements on four out of five cryptocurrencies, it showed a higher and more frequent drawdown on average than the proposed models [8].

### 3. Discussion and Conclusion

This paper has provided a review of the applications of machine learning and deep learning in trading. We see that these techniques show an overall advantage over conventional rule-based strategies. Despite offering only a slight advantage over simple strategies in terms of profits, it is preferred due to the ability to process a large variety of data and adapt to the ever-changing markets. Models like LSTM networks have shown an overall remarkable performance; however, their advantages are diminishing as the usage of similar models increases. Similar patterns are observed with random forests; they have dominated DNN and a couple of other base machine learning models. DNN does not perform very well, but surprisingly it can generate significant profits during the financial crisis. Lastly, DRL emerged as a promising model in trading and was able to integrate with other indicators and data for practical applications.

On the contrary, it is worth noting that there are still challenges to overcome, including the common mistake of overfitting, ensuring consistent performance in highly volatile markets, and demanding computational power and resources. As we see from the studies, deep neural networks struggle in unpredictable markets, for example, Tesla stock. However, these do not diminish the potential of machine learning and deep learning in finance. Future research should focus on developing more robust and adaptive machine learning and deep learning models so that they can navigate in highly volatile markets.

### References

- [1] Muhammad, D., Ahmed, I., Naveed, K., & Bendeche, M. (2024). An explainable deep learning approach for stock market trend prediction. *Heliyon*, 10(21), e40095. <https://doi.org/10.1016/j.heliyon.2024.e40095>
- [2] Chaboud, A. P., Chiquoine, B., Hjalmarsson, E., & Vega, C. (2014). Rise of the Machines: Algorithmic Trading in the Foreign Exchange Market. *The Journal of Finance (New York)*, 69(5), 2045–2084. <https://doi.org/10.1111/jofi.12186>
- [3] Peng, Y., & Souza, J. G. de M. (2024). Machine learning methods for financial forecasting and trading profitability: Evidence during the Russia–Ukraine war. *Revista de Gestão*, 31(2), 152–165. <https://doi.org/10.1108/REG-05-2022-0079>
- [4] Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2), 689–702. <https://doi.org/10.1016/j.ejor.2016.10.031>
- [5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature (London)*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>

- [6] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654-669. <https://doi.org/10.1016/j.ejor.2017.11.054>
- [7] Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, 173, 114632-. <https://doi.org/10.1016/j.eswa.2021.114632>
- [8] Kochliaridis, V., Kouloumpri, E., & Vlahavas, I. (2023). Combining deep reinforcement learning with technical analysis and trend monitoring on cryptocurrency markets. *Neural Computing & Applications*, 35(29), 21445–21462. <https://doi.org/10.1007/s00521-023-08516-x>