Financial Portfolio Construction for Quantitative Trading Using Deep Learning Technique

Rasha Abdel Kawy[1](✉), Walid M. Abdelmoez[2](✉), and Amin Shoukry[3,4](✉)

[1] Computer Science Department, College of Computing and Information Technology, Arab Academy for Science Technology and Maritime transport, Alexandria 1029, Egypt
Rasha.Shokry@student.aast.edu

[2] Software Engineering Department, College of Computing and Information Technology, Arab Academy for Science Technology and Maritime transport, Alexandria 1029, Egypt
walid.abdelmoez@aast.edu

[3] Computer Science and Engineering Department, Egypt-Japan University of Science and Technology, Alexandria 21934, Egypt
amin.shoukry@ejust.edu.eg

[4] Computer and Systems Engineering Department, Faculty of Engineering, Alexandria University, Alexandria 21544, Egypt

Abstract. Stock portfolio construction is a difficult task which involves the simultaneous consideration of dynamic financial data as well as investment criteria (e.g.: investors required return, risk tolerance, goals, and time frame). The objective of this research is to present a two phase deep learning module to csonstruct a financial stocks portfolio that can be used repeatedly to select the most promising stocks and adjust stocks allocations (namely quantitative trading system). A deep belief network is used to discover the complex regularities among the stocks while a long short-term memory network is used for time series financial data prediction. The proposed deep learning architecture has been tested on the american stock market and has outperformed other known machine learning techniques (support vector machine and random forests) in several prediction accuracy metrices. Furthermore, the results showed that our architecture as a portfolio construction model outperforms three benchmark models with several financial profitability and risk-adjusted metrics.

Keywords: Portfolio construction · Quantitative trading system · Deep learning · DBN · LSTM

1 Introduction

The Portfolio is a collection of financial investment assets such as stocks, bonds, commodities, etc. Portfolio management is the science of selecting a suitable

© Springer Nature Switzerland AG 2021

group of assets with the right proportions to meet the investors' strategic objectives. Managing the portfolio includes repeated evaluation of the portfolio value, making the necessary trading processes to generate a balance between the required return and the acceptable level of risk. Quantitative trading is defined as using mathematical and statistical methods to identify and build an automatic rule-based model to trade assets in the financial markets. Quantitative trading has the advantage of taking out emotions, rumors and fraud from the trading decisions which allow investors to invest with confidence and clarity. Machine learning techniques were widely used in quantitative trading. In [1], integrating support vector machines (SVM) with other classification methods has been proposed to forecast the weekly movement direction of NIKKEI 225 index. In [2], a forecasting model based on chaotic mapping firefly algorithm and SVR is proposed to predict stock market price, while [3] proposed a risk-adjusted profitable trading rule based on technical analysis and pattern recognition techniques. Moreover, [4,5] presented a comprehensive literature review on the application of evolutionary computation (EC), including genetic algorithms, genetic programing and multi-objective evolutionary algorithms, in stock trading and other financial applications, while [6] suggested the formation of a recursive clustering technique. Once the assets are hierarchically clustered, a risk- adjusted capital allocation is applied. In [7], the author explores the use of Gaussian processes and Bayesian optimization in modeling "the structure of interest rates", and building the trend-following optimization strategies. In [8], a comprehensive survey of particle swarm optimization (PSO) algorithm, in studying market behaviors is given, as well as, the potential future research directions for enhancing PSO-based stock market prediction.

The objective of this research is to present a deep learning quantitative trading architecture that can be used for stocks selection and stocks allocation to achieve maximum return with an accepted risk level. The proposed architecture has two modules. The first selects the efficient stocks for investment using deep learning while the second decides the budget to be invested in each selected stock. Our contributions can be summarized as:

1. Proposing a novel automatic quantitative trading model that relies on a Deep Belief Network (DBN) to extract the financial data regularities and reducing its dimensionality while using an LSTM (Long Short Term Memory Neural Network) to learn/model the dependency in the input financial data time series.
2. Conducting intensive experiments to evaluate and compare the performance, of the proposed deep learning architecture, to other conventional machine learning techniques in stocks' price prediction based on several prediction accuracy metrics.
3. Capturing the performance of the proposed architecture as a quantitative trading module and comparing it to other markets' strategies benchmarks using several risk-adjusted profitability metrics.

The remaining of this paper is organized as follows: Sect. 2 defines what is meant by a quantitative trading system and provides technical background on the deep learning networks used in this paper, in addition to presenting a review about the related work (using deep belief networks and long short-term memory networks (LSTM) in predicting stocks prices). Section 3 presents the proposed deep learning architecture. Section 4 describes the experiments performed on the proposed model and its performance evaluation. Finally, conclusions are summarized in Sect. 5.

2 Background and Literature Review

2.1 Quantitative Trading System

The quantitative trading system is based on forecasting the market movement. First, the proper mix of assets with the right proportions are chosen (portfolio construction) then - based on the market movement - trading rules are built that aim to achieve the maximum return given the risk tolerance set by the investor. A periodic evaluation is performed to determine whether the obtained performance is satisfactory or not. Necessary adjustments are made in the stocks to preserve the investment objectives.

2.2 Deep Learning Networks

2.2.1 Deep Belief Networks

Deep Belief Networks (DBN) [9] are probabilistic generative neural networks composed of multiple layers of restricted boltzmann machines (RBM) to capture higher-level representations of input features with the advantage of avoiding getting stuck at local optima.

2.2.2 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) [10] networks are special type of recurrent neural networks (RNN) capable of learning long-term dependence in time series data with the advantage of avoiding vanishing gradient problems existing in the training of RNN.

2.3 Related Work (LSTM and DBN in Financial Forecasting)

Numerous studies have shown that long short-term memory neural networks are very effective networks in the forecasting of financial times series data. Research in [11] introduced multivariate denoising wavelet transforms (WT), in order to eliminate the noise in the time-series data, then combined stacked autoencoders (SAEs) and long-short term memory (LSTM) networks to forecast six market indices. Fischer et al. [12], deployed LSTM networks for predicting out-of-sample directional movements for the S&P 500 stocks in the period from 1992 to 2009. Research in [13] proposed a long short-term memory (LSTM) network to predict stock movement in order to construct multiple portfolio optimization techniques using equal-weighted modeling (EQ), simulation modeling Monte Carlo simulation (MCS), and mean variant optimization (MVO). The work in [14] presented a DBN model with strong ability to generate high level features representation for accurate financial prediction that has been tested on a real dataset of French companies. In [15] a DBN has been used to forecast the currencies exchange rates. Conjugate gradient method was applied to accelerate the learning for DBN. The results showed that DBN outperforms other traditional methods. While in [16] an RBM is combined with SVM to detect trends in the Brazilian Stock Market prices. The obtained results were better compared to those obtained by SVMs only. Rasha et al. [17] proposed a novel multi-stock end-to-end trading model based on DBN and multi-agent deep reinforcement learning. Its efficiency, compared to existing techniques has been verified on datasets of different characteristics obtained from the American stock market.

3 Proposed Deep Learning Model Architecture

The proposed deep learning architecture is shown in Fig. 1 It is specialized in quantitative trading. Per each trading time period T, the model is fed with the data received from the financial market. Depending on this data, the model generates predictions on the stocks' returns during the next trading period. According to the errors between the predicted and the actual stocks' returns values, the back-propagation learning algorithm is applied to update the weights of the model's NNs. The model consists of two phases.

1. **Phase I (Deep Learning module):** in which financial raw data, for a set of M market stocks is received. For each stock, a DBN module extracts discriminant features from the high-dimensional raw financial data and reduces its dimensionality. Next, an LSTM module is fed with the stock DBN features and predicts the sum of the relative changes in the stock returns during the next trading period as given in Eq. (1) below.
2. **Phase II (Portfolio construction and assets allocation module):** the portfolio is dynamically constructed from the best N, as determined in Phase I, out of the available M stocks. The fund allocated to each selected stock depends on its predicted sum of relative changes in the stock returns, during the next period, as given in Eq. (2) below.

3.1 Phase I (Deep Learning Module)

3.1.1 Stocks' Features

The following types of data are used at the input:

1. **OHLCV data:** include the opening price, highest price, lowest price, closing price and the trading volume of each stock.

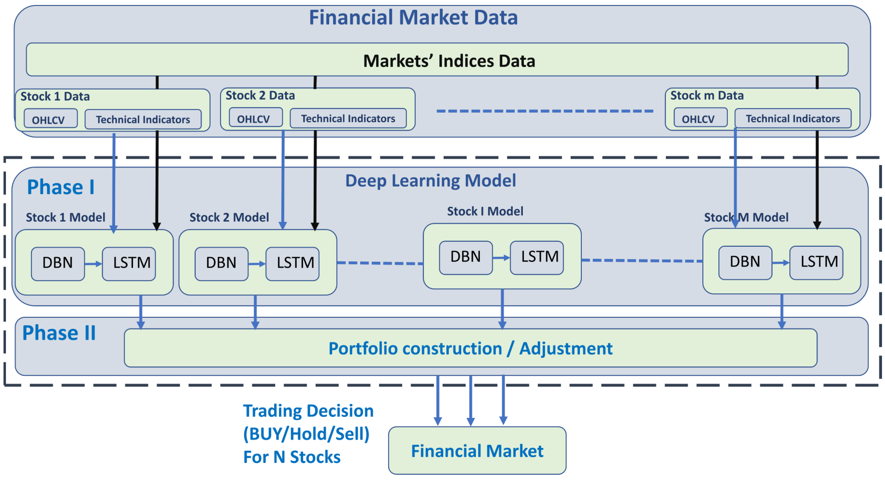

Fig. 1. Proposed deep learning model architecture.

2. **Technical indicators:** which correspond to equations applied on the OHLCV data for a stock to obtain indicators about its future position(price trend/oscillation/volatility/moving average) in the market: e.g. Directional Movement Indicator (DMI), Exponential Moving Average (EMA), Relative Strength Index (RSI), Stochastic Momentum Index (SMI), and Weighted Moving Average (WMA).
3. **Financial market indices:** which correspond to the averages of the OHLCV data for a group of stocks that is used to get an indication about the whole market direction. In our research, five indices are fed to the system, namely, (ˆGSPC, ˆ DJI, ˆIXIC, ˆNYAC and ˆXAX).

The data of the financial time series is fed to our proposed module in steps where each step T consists of 5 d. Each day includes 50 features corresponding to 5 features from the OHLCV data values of the stock, 25 features from the financial market indices and 20 features from the technical indicators of the stock. At each time step T, the features of each of the five days are arranged sequentially and in order to form a feature vector of length 250.

3.1.2 DBN Module

The deep DBN network consists of three hidden layers with 100-80-60 neurons and an output layer of 20 neurons (representing the compressed and uncorrelated features extracted by the DBN from the raw financial data relative to the higher dimensional input vector of length 250).

3.1.3 LSTM Module

The LSTM receives 20 input features from the DBN and has one hidden layer with 28 units (each unit consists of input, output, memory and forget gates) while the output consists of only one neuron that represents a prediction of the stock return during the next period T. The actual stock return during the period T, is denoted R_T, and is calculated as follows:

$$R_T = \sum_{t \in 1}^{T} (\frac{P_t}{P_{t-1}} - 1) \tag{1}$$

where P_t is the stock closing price during day t, $t \in [1, T]$.

3.2 Phase II (Portfolio Construction and Assets Allocation Module)

Consider $S = (S_1, S_2, \ldots\ldots, S_M)$ a set of M market stocks. For each stock in the set S, the model will run and predict the return of this stock during the next period. Let $R_T = (R_T^1, R_T^2, \ldots\ldots, R_T^M)$ be the set of stocks' returns for the period T. let N be the preferred number of stocks to trade in (the number of the stocks in the constructed portfolio). The set of largest N stocks' returns from the set R_T will be chosen $R_T^N = (R_T^1, R_T^2, \ldots\ldots, R_T^N)$ to construct the portfolio L, and the weight of each stock S in portfolio L will be estimated as :

$$W_T^S = \frac{R_T^S}{\sum_{s=1}^{N} R_T^S} \tag{2}$$

While other stocks will have zero weights. At time interval T = 0, the amount of fund F, is used to buy shares of the portfolio's stocks, using: $F_T^S = F * W_T^S$. However, in other time series steps the (Buy/Sell/Hold) signals are generated according to the following rules:

If $W_T^S > W_{T-1}^S$ then a Buy signal with $F * (W_T^S - W_{T-1}^S)$ fund is generated, If $W_T^S < W_{T-1}^S$ then a sell signal with $F * (W_{T-1}^S - W_T^S)$ fund is generated.

In case the fund required for Buying/Selling a stock is less than the minimum transaction cost allowed, a hold signal is generated and the fund will be reallocated to other stocks buying/selling transactions.

4 Experiments Design

4.1 Datasets Used

The thirty industrial stocks registered in the Dow Jones Industrial Average "DJI" index are used as the market's stocks to be traded in. All the financial data are available online at "Yahoo Finance". The datasets from January 2009 to December 2019 have been downloaded. Each input dataset is composed of a window of fixed length (Fifty weeks) moving along the time series (continuously shifted for 5 weeks periods) as shown in Fig. 2. Each input dataset has been divided into three parts, 80% of which (around 40 weeks) is used for training, i.e. to set the model parameters, while 10% (around 5 weeks), is used for validation, i.e. for hyper-parameters tuning, and the last 10% (around 5 weeks) is used for testing, i.e. to test the model predictions. The experiments were conducted on a personal computer (with Microsoft windows 10 Enterprise operating system) with Intel Core i5-3210M (2.50 GHz), 2 core(s), and 128 GB RAM. Python 3.7 with TensorFlow 2.0 backend have been used for implementation.

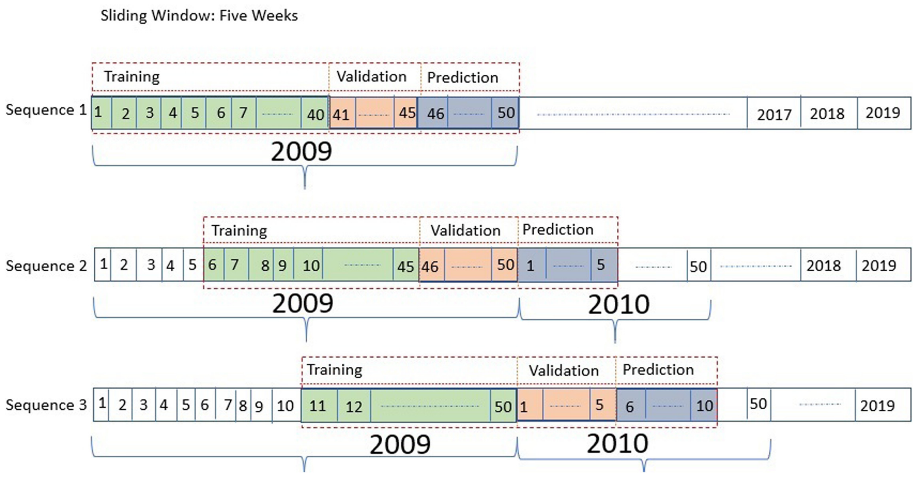

Fig. 2. Moving windows for training, validation and prediction (testing) datasets. The window is continuously shifted for 5 weeks periods.

4.2 Performance Evaluation

4.2.1 Benchmark Models

For benchmarking our proposed module Deep Learing (DBN with LSTM) the following conventional machine learning module are chosen:

1. **Random Forests (RAF):** introduced in [18] and developed in [19]. In this technique multiple decorrelated decision trees (ensemble of B trees each with a maximum depth J) are built on different samples of the training data. The prediction decision is based on the majority voting from the B committee trees. The number of trees B is set to be 100 while the maximum depth J is set to be 15.
2. **Support Vector Regression (SVR):** introduced in [20] and developed in [21]. The SVR uses the same principles as the SVM for classification. In this technique a linear regression function in a high dimensional feature space is computed to map the input data and minimize the generalization error bound to ensure that the distance between the input data points and the hyperplane generated by this function is not farther than epsilon. The hyper-parameters of SVR C, gamma and epsilon are set to 0.1, 0.01 and 0.1, respectively.

For financial bench-marking, the previous machine learning techniques are used to predict the stocks' prices, and the portfolio construction strategy along with the buy and hold strategy, described in Sect. 3.2, are used.

4.2.2 Predictive Accuracy Metrics

The conventional indicators adopted to evaluate the performance of the proposed model are given in Table 1. each row contains the metric name, acronym, how it is estimated and a brief description.

Table 1. Predictive accuracy metrices

Metric	Acronym	Estimation
Mean squared error	MSE	$\frac{1}{N}\sum_{t=1}^{N}(x_t - x_t^*)^2$
Mean absolute percent error	MAPE	$\frac{1}{N}\sum_{t=1}^{N}\lvert\frac{x_t - x_t^*}{x_t}\rvert$
Correlation coefficient	R	$\frac{\sum_{t=1}^{N}(x_t - x_t^-)(x_t^* - x_t^{*-})}{\sqrt{\sum_{t=1}^{N}(x_t - x_t^-)^2(x_t^* - x_t^{*-})^2}}$
Theil's inequality coefficient	Theil U	$\frac{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(x_t - x_t^*)^2}}{\sqrt{\frac{1}{N}\sum_{t=1}^{N}(x_t)^2}+\sqrt{\frac{1}{N}\sum_{t=1}^{N}(x_t^*)^2}}$

In Table 1, x_t and x_t^* stand for the actual and predicted values, respectively. N represents the number of test prediction periods. x_t^- is the mean value of the actual vector values $(x_1, x_2, \ldots, x_N)$, and x_t^{*-} is the mean value of the predicted vector values $(x_1^*, x_2^*, \ldots, x_N^*)$. The smaller the MSE/MSEP/Theil U values, the better the prediction results. On the other hand, the larger the R value the better the prediction results.

4.2.3 Profitability Metrics

The profitability test is implemented to find how the proposed model can earn real profits for the investors when implemented in real stock markets. Table 2 contains the profitability metrics used in our experiment. The higher the annualized return (AR) and the calmar ratio values, the better for the investors. While the lower the values of the standard deviation (SD), sharp ratio (SR) and maximum drawdown, the lower the risk taken in the investment and the better for the investors.

Table 2. Profitability metrics

Metric name	Short	Description
Volatility	SD	Annualized volatility, standard deviation of the profit and loss during the given time interval
Annualized Return	AR	The geometric average of the amounts of money earned by an investment each year over a given time period
Sharp ratio	SR	The ratio between (annualized return minus free-risk return) and annualized volatility
Maximum Drawdown	MDD	Maximum loss from a peak to a trough in the value of the trading portfolio during time interval T
Calmar Ratio	Calmar	The ratio between annualized return and max drawdown

4.3 Results

4.3.1 Predictive Accuracy Metrics Results

Figure 3 shows the average results obtained from the predicted data by the three models plus the average actual results. The results are given for the ten year periods from 2010 to 2019.

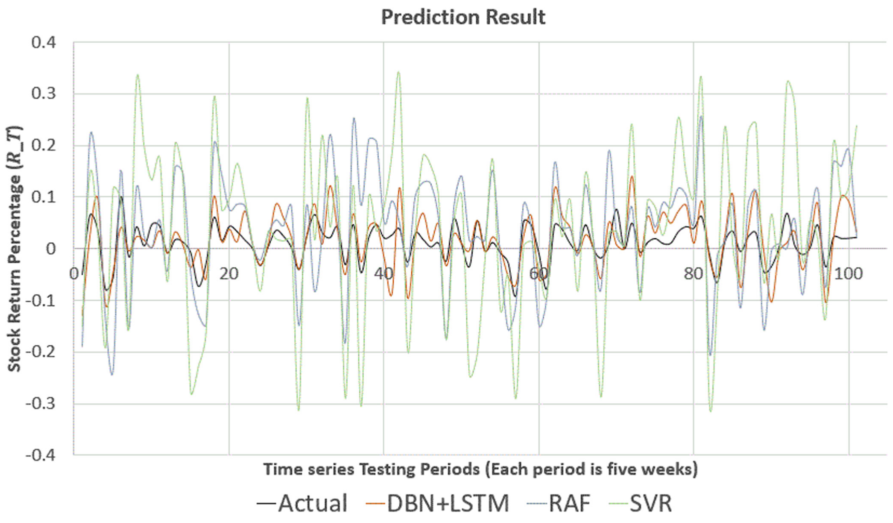

Fig. 3. Stocks' return percentage prediction data (R_T) versus actual data. Each interval of length 10 on the horizontal axis corresponds to one year.

Table 3 shows that the proposed Deep Learning model outperforms the other models since all metrics, except R, are required to be of low values. The SVR has the worst prediction result, as it has the largest deviations from the actual data.

Table 3. Prediction accuracy metrics of the different models calculated for the period from 2010 to 2019. For the MSE metric the shown results should be multiplied by 10^{-2}.

Deep Learning (DBN with LSTM)											
Metric	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
MSE	0.189	0.087	0.076	0.142	0.294	0.058	0.16	0.199	0.241	0.23	0.168
MAPE	0.0879	0.0919	0.248	0.102	0.189	0.103	0.186	0.209	0.245	0.244	0.171
R	2.219	2.008	1.931	1.848	1.529	2.01	1.718	1.537	2.406	1.286	1.85
Theil U	0.356	0.362	0.392	0.379	0.588	0.278	0.419	0.471	0.442	0.509	0.42
Random Forests (RAF)											
Metric	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
MSE	1.163	0.948	0.305	2.049	0.79	0.613	0.604	0.336	1.132	0.912	0.885
MAPE	0.276	0.4488	0.559	0.376	0.671	0.4.01	0.604	0.541	0.544	0.595	0.499
R	2.557	2.025	1.62	0.505	1.397	2.1	1.621	1.506	2.229	1.147	1.67
Theil U	0.515	0.594	0.547	0.721	0.654	0.564	0.549	0.54	0.634	0.675	0.6
Support Vector Regression (SVR)											
Metric	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
MSE	1.997	2.431	1.68	1.967	2.27	1.949	1.014	1.252	2.992	1.979	1.953
MAPE	0.574	0.856	0.523	0.323	0.925	0.515	0.765	0.603	0.768	0.894	0.678
R	1.93	2.152	1.66	1.648	1.582	0.784	2.214	2.055	2.206	1.505	1.77
Theil U	0.626	0.695	0.742	0.679	0.755	0.712	0.627	0.677	0.745	0.725	0.699

4.3.2 Profitability Metrics Results

The profitability of each model is compared against the returns of the buy-and-hold strategy. For each model a portfolio, with ten stocks, is constructed and the trading strategy mentioned in Sect. 3.2 is applied to all models. Table 4 presents the profitability of each model. The results demonstrate that the proposed Deep learning model outperforms the other models in both the profitability return and the risk-return metrics. The RAF outperforms the SVR by a clear margin also. The Worst technique in the profitability return and the risk-return metrics is the buy and hold technique.

Table 4. Profitability metrics of the different models calculated for the period from 2010 to 2019.

Deep Learning (DBN with LSTM)											
Metric	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
SD	0.069	0.044	0.041	0.054	0.066	0.046	0.052	0.046	0.076	0.063	0.0557
AR	19.5%	16.4%	20.6%	23.6%	18.6%	16.7%	18.6%	25.1%	16.8%	28.2%	20.4%
SR	2.1	2.59	3.8	3.4	2.06	2.54	2.61	4.37	1.55	3.689	2.77
MDD	0.226	0.159	0.126	0.172	0.214	0.144	0.162	0.154	0.203	0.274	0.184
Calmar	0.863	1.031	1.635	1.372	0.869	1.16	1.146	1.63	0.828	1.031	1.156
Random Forests (RAF)											
Metric	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
SD	0.161	0.125	0.073	0.141	0.096	0.101	0.094	0.059	0.137	0.099	0.1086
AR	14.6%	12.2%	16.2%	18.8%	16.12%	16.65%	15.82%	19.9%	17.51%	15.3%	16.31%
SR	0.596	0.576	1.534	0.979	1.158	1.64	1.151	2.525	0.913	1.044	1.041
MDD	0.388	0.349	0.238	0.412	0.313	0.309	0.309	0.339	0.321	0.215	0.319
Calmar	0.376	0.35	0.681	0.456	0.515	0.539	0.512	0.587	0.545	0.713	0.524
Support Vector Regression (SVR)											
Metric	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
SD	0.169	0.195	0.155	0.175	0.161	0.139	0.125	0.105	0.196	0.142	0.156
AR	10.8%	9.12%	10.35%	13.9%	12.68%	11.7%	13.4%	14.6%	8.88%	12.88%	11.83%
SR	0.343	0.211	0.345	0.509	0.477	0.482	0.672	0.914	0.198	0.555	0.437
MDD	0.452	0.388	0.289	0.427	0.429	0.462	0.463	0.434	0.352	0.453	0.415
Calmar	0.239	0.235	0.358	0.326	0.296	0.253	0.289	0.336	0.252	0.284	0.287
Buy and Hold											
Metric	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
SD	0.058	0.039	0.026	0.035	0.026	0.045	0.043	0.019	0.04	0.029	0.036
AR	9.8%	5.53%	7.26%	11.2%	8.4%	2.3%	7.34%	9.3%	–5.63%	8.2%	6.37%
SR	0.828	0.136	0.869	1.771	1.308	–0.6	0.544	2.263	–2.66	1.103	0.381
MDD	0.67	0.56	0.594	0.593	0.522	0.544	0.62	0.52	0.64	0.622	0.589
Calmar	0.146	0.099	0.122	0.189	0.161	0.042	0.118	0.179	–0.088	0.132	0.11

5 Conclusion

This research proposes a novel framework to construct a financial stocks' portfolio, to be used as a quantitative trading system, that repeatedly adjust the number of stocks and their percentages based on a deep learning prediction module. The proposed module has the ability to extract useful knowledge from the input dynamic financial information using a DBN network, that is also used as a dimension reduction tool. A powerful prediction neural network (LSTM) is used to forecast the stocks' performance. The predicted stocks' performance affects the buy/sell/hold decisions taken by the model and the corresponding fund allocated to each stock.

References

1. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting stock market movement direction with support vector machine. Comput. Oper. Rese. **32**(10), 2513–2522 (2005)

2. Kazem, A., Sharifi, E., Hussian, F.K.: Support vector regression with chaos-based firefly algorithm for stock market price forecasting. Appl. Soft Comput. **13**(2), 947–958 (2013)
3. Cervelló-Royo, R., Guijarro, F., Michniuk, K.: Stock market trading rule based on pattern recognition and technical analysis: forecasting the DJIA index with intraday data. Expert Syst. Appl. **42**(14), 5963–5975 (2015)
4. Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E.W.T., Liu, M.: Application of evolutionary computation for rule discovery in stock algorithmic trading: a literature review. Appl. Soft Comput. **36**, 534–551 (2015)
5. Aguilar-Rivera, R., Valenzuela-Rend-on, M., Rodr-guez-Ortiz, J.: Genetic algorithms and Darwinian approaches in financial applications: a survey. Expert Syst. Appl. **42**, 7684–7697 (2015)
6. Raffinot, T.: Hierarchical clustering-based asset allocation. J. Portfolio Manage. Multi-Asset Special Issue **44**(2), 89–99 (2018)
7. Gonzalvez, J., Lezmi E., Roncalli, T., Xu J.: Financial Applications of Gaussian Processes and Bayesian Optimization. arXiv:1903.04841 (2019)
8. Thakkar, A., Chaudhari, K.: A comprehensive survey on portfolio optimization, stock price and trend prediction using particle swarm optimization. Arch. Comput. Meth. Eng. **28**(4), 2133–2164 (2020). https://doi.org/10.1007/s11831-020-09448-8
9. Hinton, G., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 527–554 (2006)
10. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed on Nov 2020
11. Bao, W., Yue, J., Rao, Y.: A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE **12**, e0180944 (2017)
12. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. Euro. J. Oper. Res. **270**, 654–669 (2018)
13. Ta, V.-D., Liu, C.-M., Tadesse, D.A.: Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. Appl. Sci. **10**(2), 437 (2020)
14. Ribeiro, B., Lopes, N.: Deep Belief Networks for Financial Prediction. Lecture Notes in Computer Science, vol. 7064. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24965-5_86
15. Shen, F., Chao, J., Zhao, J.: Forecasting exchange rate using deep belief networks and conjugate gradient method. Neurocomputing **167**, 243–253 (2015)
16. Assis, C.A.S., Pereira, A.C.M., Carrano, E.G., Ramos, R., Dias, W.: Restricted boltzmann machines for the prediction of trends in financial time series. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–18. Rio de Janeiro (2018)
17. AbdelKawy, R., Abdelmoez, W.M., Shoukry, A.: A synchronous deep reinforcement learning model for automated multi-stock trading. Progress Artif. Intell. **10**(1), 83–97 (2021). https://doi.org/10.1007/s13748-020-00225-z
18. Ho, T.K.: Random decision forests. In: Proceedings of the Third International Conference on Document Analysis and Recognition, pp. 278–282. IEEE (1995)
19. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
20. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**(1), 199–222 (2004)
21. Lu, C.J., Lee, T.S., Chiu, C.C.: Financial time series forecasting using independent component analysis and support vector regression. Decis. Support Syst. **47**(2), 115–125 (2009)