American Journal of Computer Science and Technology
2024, Vol. 7, No. 3, pp. 51-61
https://doi.org/10.11648/j.ajcst.20240702.14

SciencePG
Science Publishing Group

Research Article

An Investigation of Predictability of Traders' Profitability Using Deep Learning

David Ademola Oyemade[*], **Eseoghene Ben-Iwhiwhu**

Department of Computer Science, Federal University of Petroleum Resources, Effurun, Nigeria

Abstract

Trading in the financial market is a daunting task in spite of the attracting increase of the daily turnover of the Forex financial market from 6.5 trillion USD in 2022 to approximately 7.5 trillion USD in 2024. About 80% of retail investors lose money. However, to minimize the risk of losses, investors explore the possibility of profitable trading by resorting to social trading. In social trading of the financial market, the performance statistics and performance charts of traders with diverse trading strategies, methods and characteristics are showcased by the financial market brokers to enable investors decide on which trader’s signal to adopt or copy for profitable investment. However, investors are often faced with the problem of choosing a set of profitable traders among thousands with different past hypothetical results, in spite of the provision of traders’ performance ranking, made available by the brokers. The investors have serious concern on the stability, sustainability and predictability of a trader’s future performance which will eventually determine the investors profit or loss if the trader’s signals are copied or followed. This paper applies three deep learning models: the multilayer perceptron, recurrent neural network and long short term memory for the prediction of traders’ profitability to provide the best model for investment in the financial market, and reports the experience. The results of the study show that recurrent neural network performs best, followed by long short term memory while multilayer perceptron yields the least results for the prediction. These three models yield a mean squared error of 0.5836, 0.7075 and 0.9285 respectively in a test scenario for a trader.

Keywords

Deep Learning, Traders, Financial Market, Performance, Prediction

1. Introduction

Retail trading in the financial market remains a daunting task. About 80% of retail traders in the Forex financial market lose money. However, the attracting increase in the daily turnover of the Forex financial market from 6.5 trillion USD in 2022 to about 7.5 trillion USD in 2024 still makes it a choice for investors [1]. In recent times, many investors are turning to social trading. Social trading platforms are invaluable in the financial market for investors who aspire to benefit from the experience and expertise of successful traders by simply copying the signals of the successful traders or following the exact trading strategies of the traders. The performance statistics and historical data of such traders are transparently provided by the financial market brokers such as forex brokers. Such social trading platforms are integrated with simplified tools that interested investors can apply for setting up invested cap-

[*]Corresponding author: oyeamde.david@fupre.edu.ng (David Ademola Oyemade)

Received: 4 June 2024; **Accepted:** 24 June 2024; **Published:** 8 July 2024

itals and for specifying the traders to follow. The investors make gains or losses depending on the performance of the traders. The trader gains commissions from the profit made by the investor as defined by the financial market broker. The broker in turn makes some gains from both the traders and the investors. However, the investor's decision in selecting the trader to follow is based on hypothetical results of traders which often may not be sufficient to guarantee the future performance of the traders who usually have different characteristics and who sometimes are influenced by emotions that may adversely affect their trading strategies and principles because of the complexity of trading in the financial market. Investors have serious concern on the stability, sustainability and predictability of a trader's future performance which will eventually determine the investors profit or loss if the trader's signals are copied or followed. It thus becomes necessary that intelligent systems be considered for the predictability of traders' performance in an effort to preventing avoidable losses by investors. Fxtm.com, OctaFx.com and ZuluTrade.com are examples of social trading platforms. This study investigates the application of deep learning models for the predictability of the financial traders' profitability. This paper is structured as follows: the introduction is followed by the technical background and this is followed by related works on the financial market and artificial intelligence models. The section on materials and methods follows the related work. The paper ends with results and discussion followed by the conclusion.

2. Technical Background

In this section, the technical background of the study is discussed.

2.1. Deep Learning

Deep Learning (DL) is a sub-field of machine learning methods [2]. It uses an artificial neural network that consists of multiple processing layers and models data with a high level of abstraction. Thus in DL models, the essential features of the input data can be automatically extracted through a general learning procedure [3]. The descriptions of multilayer perceptron (MLP), recurrent neural network (RNN) and long short-term memory (LSTM) are given in the next subsections.

2.2. MLP

MLP is classified among the first set of artificial neural network developed. Compared to shallow nets, MLP consists of more layers. However, deep multilayer perceptron models (DMLP) contains three layers: input, hidden, and output, although some variations can exist depending on the problem requirement. The number of neurons in each layer and the number of layers are the hyper-parameters of MLP. Each neuron in the hidden layers has input (*x*), weight (*w*), and bias (*b*) terms. Each neuron also has a nonlinear activation function, which produces a cumulative output of the previous neurons. Equation (1) illustrates the output of a single neuron in the Neural Network (NN). The equations for widely accepted nonlinear activation functions include sigmoid, hyperbolic tangent, Rectified Linear Unit (ReLU), leaky-ReLU, swish and softmiax. These are available in literature and they are well defined [4-8].

$$y_i = \sigma(\textstyle\sum_i W_i\, x_i + b_i) \tag{1}$$

2.3. RNN

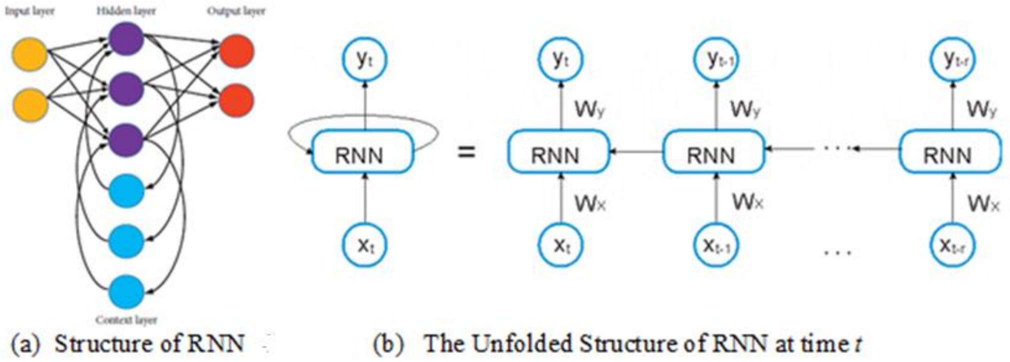
RNN receives information from the input layer and the activation from the previous time-step forward propagation to update its hidden layer. Unlike in feed-forward neural networks, temporal information embedded into the input data can be accommodated by RNN. The computation of each time steps of RNN is given in Equation (2).

$$h_t = \sigma(W_x.x_t + W_h.h_{t-1} + b_h) \tag{2}$$

where $W_x \in R^{m\times n}$ is the weight matrix connecting the input layer and the hidden layer, *m* is the size of the input, *n* is the size of the hidden layer, $W_h \in R^{n\times n}$ is the weight matrix connecting the neurons of the hidden layer to itself, and it is shared across hidden states h_t−1 and h_t through time, b_h is the bias vector of the hidden layer, and σ is the activation function to generate the hidden state. The network output is given as shown in Equation 3.

$$y_t = \sigma(W_y.h_t + + b_y) \tag{3}$$

where $W_y \in R^n$ is the weight connecting the hidden layer and the output layer, *by* is the bias vector of the output layer, and σ is the activation function of the output layer. Figure 1 shows the general structure of RNN and the unfolded structure of RNN at time *t*.



(a) Structure of RNN (b) The Unfolded Structure of RNN at time *t*

Figure 1. *The structures of recurrent neural network [12].*

2.4. LSTM

LSTM evolved from RNN. It is designed to address the problem of vanishing and exploding gradients associated with the training process of RNN [9, 10]. LSTM networks are made up of LSTM units called memory blocks and merged to form an LSTM layer. Each LSTM unit consists of cells with an input gate, output gate, and forget gate which regulate the information flow. This design enables each cell to remember necessary values over time intervals. The equations for the forward pass of the LSTM unit are available in literature with various symbols representing input vector to the LSTM unit, forget gate's activation vector, input gate's activation vector, output gate's activation vector, output vector of the LSTM unit, cell state vector, sigmoid function, hyperbolic tangent function, element-wise (Hadamard) product, weight matrices to be learned and bias vector parameters to be learned [11].

3. Related Works

Previous and related works are discussed in this section.

Kasongo and Sun [12] introduced an intrusion detection system which combined deep learning with filter based feature engineering. Tariq et al. [13] focused on the review of deep learning security and privacy techniques. Sehovac and Grolinger [14] proposed a deep learning method for the forecasting of electoral load. Joseph et al. [15] applied deep learning with Internet of Things for the automatic identification, classification and organization of waste as a solution to national waste management problem. Abdul Lateef et al. [2] focused on the taxonomy survey of deep architectures and the review of deep learning applications of intrusion detection system. Gayathri et al. [16] proposed some deep learning techniques for the tracking of the activities of unmanned aerial vehicles. Fang et al. [17] proposed a hybrid of bi-directional recurrent neural networks and long short-term memory for the prediction of cyber-attacks.

On the application of deep learning and other artificial intelligence models to the financial market, Sezer et al. [3] explored the application of deep learning to the financial time series in a literature review covering fifteen years. Vogl et al. [18] evaluated the performance of wavelet neural networks in comparison to others for the analysis of the financial market data. Schmidt-Kessen et al. [19] analyzed the inflexibility of algorithmic trading and its associated systemic risk as an automated financial contract. Zhang [20] proposed an optimization technique for the financial markets using particle swarm optimization. Dash [21] proposed the combination of a recurrent Legendre polynomial neural network with shuffled frog leaping based learning strategy as a hybrid performance analysis approach for forex prediction. Supsermpol [22] proposed the use of logistic regression and random forest algorithm for the prediction of the financial performance of some companies publicly listed on the stock market during transition period, focusing on Thailand. Ueda et al. [23] proposed a method which adopts paired information consisting of social media topic and its sentiment mood for financial market price forecasting. Oyemade and Ojugo [1] proposed the application of genetic algorithm for price prediction in the financial market. Oyemade et al. [24] proposed a fuzzy logic topology for profitable trading in the Forex financial market. Bevilacqua et al. [25] proposed a method for the measurement of systemic risk in the financial market by extracting and analyzing data from equity options prices. Bossaerts et al. [26] introduced the application of Kyle model for the identification of traders with positive and valuable informational price impact in the financial market.

Other related studies focused on option pricing for non-storable assets, cryptocurrency mining, oil price forecasting, financial market quality of service modeling and stock prediction [27-32]. Others include research efforts on financial market software life cycle model, hybrid greedy and dynamic programming model, property oriented trading model and scalability model [33-36] Other literatures on the applications of machine learning to the financial market centred on the analysis of the raw financial market time series data for the prediction of trends [37-40]. No study has been carried out on the application of deep learning for the prediction of traders' performance, according to investigations. This is the main contribution and novelty of this paper.

4. Materials and Methods

The fundamentals of the multilayer perceptron, recurrent neural network and long short term memory have been explained in section 2, under technical background, as the components of the methodology adopted for traders' performance prediction in this paper.

Two research questions are considered in this study. The first research question deals with the feasibility of method while the second research question deals with the efficiency of method.

Research Question 1: Can the performance and the behavior of a forex trader providing signals to others be predicted using deep learning?

Research Question 2: Which deep learning model is most efficient for the prediction of traders' performance, for traders providing signals that inexperienced investors can copy in social trading?

4.1. Materials

The software used for the experiments in this study include: Anaconda Navigator 1.9.7, Jupyter Notebook 6.1.6 installed with numpy, pandas, keras, matplotlib, datetime, sklearn, sklearn preprocessing, StadardScaler, sklearn neural network and sklearn model selection libraries. Windows 8.1 Pro 64 bit operating system was employed with Intel (R) Core (TM) i3-2330M central processing unit running at 2.20 GHz speed. The RAM size was 4.0 GB. ZuluTrade Forex broker traders' performance data was used.

4.2. Dataset

The dataset was obtained from the social trading platform of zulutrade.com. This is made freely available by ZuluTrade.com to all clients with registered trading accounts to enable the clients to monitor the performance statistics of various traders. ZuluTrade also provides the performance chats of various traders and ranks the traders using ZuluRank, a proprietary algorithm.

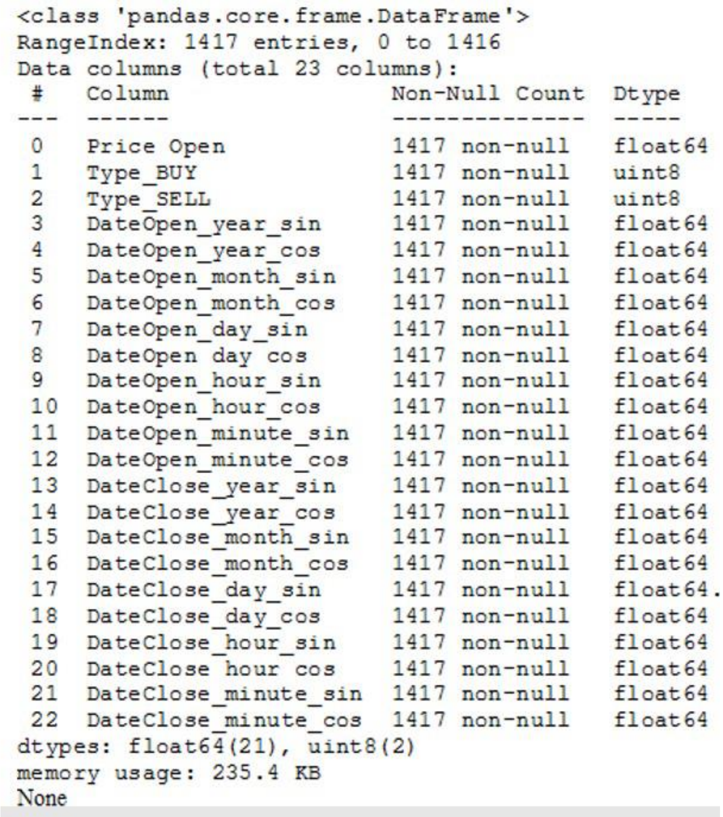
The performances of three different traders with different trading strategies and characteristics were captured over a period of 80 to 134 weeks using Microsoft Excel comma delimited flies. The three traders are referred to as Trader A, Trader B and Trader C in this study. Trader A traded on GBP/JPY currency pair, trader B traded on EUR/USD while trader C traded on different financial instruments which comprised GBP/CAD, GBP/CHF, EUR/CAD and EUR/AUD currency pairs. Table 1 gives an instance of the basic input data for Trader B. The input data include broker ticket, trade type, currency pair, date open, price open, price close and the cumulative profit.

Table 1. *Instance of the Basic Input Data of Trader B.*

Broker Ticket	Type	Currency	Date Open	Price Open	Price Close	Profit (pips)
65207563	SELL	EUR/USD	2019/05/28	2019/05/29	1.11623	1.11519
65252549	SELL	EUR/USD	2019/05/29	2019/05/29	1.11521	1.11521
65417368	SELL	EUR/USD	2019/05/29	2019/05/30	1.11332	1.11281
65739566	BUY	EUR/USD	2019/05/30	2019/05/31	1.11299	1.1135
66859276	SELL	EUR/USD	2019/06/13	2019/06/14	1.12765	1.12706
66958990	SELL	EUR/USD	2019/06/16	2019/06/17	1.12112	1.12054

4.3. Feature Encoding

In the dataset, there exist non-numeric features such as date/time (open date/time and close date/time). The date/time was split into separate components (year, month, day, hour, minute and second). Given the cyclical nature of the date/time components (excluding year), cyclical feature encoding was employed on such components. To encode year, a maximum year of 2025 is set and cyclical encoding is employed. The result of the feature encoding is feature vector of length 23, serving as input to the network. These features are represented in Figure 2 using a pandas dataframe (a popular Python library for loading and processing tabular wireframe data).



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1417 entries, 0 to 1416
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Price Open            1417 non-null   float64
 1   Type_BUY              1417 non-null   uint8
 2   Type_SELL             1417 non-null   uint8
 3   DateOpen_year_sin     1417 non-null   float64
 4   DateOpen_year_cos     1417 non-null   float64
 5   DateOpen_month_sin    1417 non-null   float64
 6   DateOpen_month_cos    1417 non-null   float64
 7   DateOpen_day_sin      1417 non-null   float64
 8   DateOpen day cos      1417 non-null   float64
 9   DateOpen_hour_sin     1417 non-null   float64
 10  DateOpen_hour_cos     1417 non-null   float64
 11  DateOpen_minute_sin   1417 non-null   float64
 12  DateOpen_minute_cos   1417 non-null   float64
 13  DateClose_year_sin    1417 non-null   float64
 14  DateClose_year_cos    1417 non-null   float64
 15  DateClose_month_sin   1417 non-null   float64
 16  DateClose_month_cos   1417 non-null   float64
 17  DateClose_day_sin     1417 non-null   float64.
 18  DateClose_day_cos     1417 non-null   float64
 19  DateClose_hour_sin    1417 non-null   float64
 20  DateClose hour cos    1417 non-null   float64
 21  DateClose_minute_sin  1417 non-null   float64
 22  DateClose_minute_cos  1417 non-null   float64
dtypes: float64(21), uint8(2)
memory usage: 235.4 KB
None
```

Figure 2*. Feature encoding using a pandas dataframe.*

4.4. Model Design

The architectural specification of the three models (MLP, RNN, and LSTM) are presented in Figure 3. Each model is a four-layer network containing two hidden layers. The input layer consists of the encoded and processed features of the data, while the output of the model is the prediction of the profit (pip) given the input. The first hidden layer of the RNN model is a RNN layer, and likewise an LSTM layer for the LSTM model. Given that the signal being predicted is future profit (pips) of a trader, the problem is cast in machine learning as a regression task (having continuous output values) and not a classification task (which contains discrete classes or output values). Therefore, the hyperbolic tangent (i.e. tanh) activation function was employed because of its suitability for the regression task.

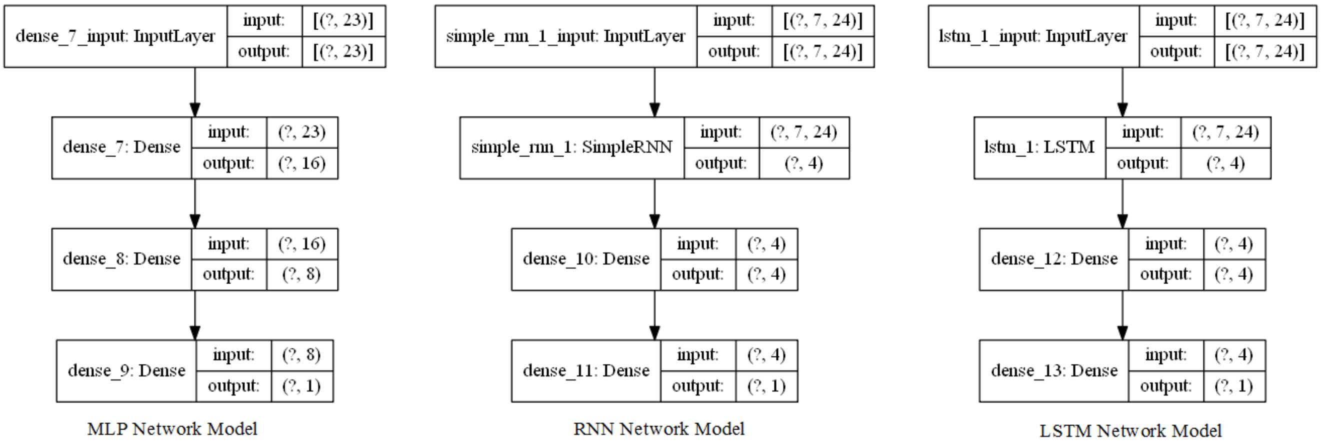


Figure 3*. Graphical representations of the deep learning models.*

4.5. Overfitting Resolution, Training Procedure and Testing

To combat overfitting, L2 regularization technique penalizing large weight values is employed. Also, given the limited size data (number of data samples), the size of the network models was reduced.

Mean squared error loss function was employed. Each model was trained for 1000 epochs, using Adam optimizer [41] with learning rate of 0.001. The data was split into train and test data sets using 80:20 ratio. The training data was pre-processed and the features were standardized to mean 0 and standard deviation of 1. The training data statistics were also used to standardize the test data. For the RNN and LSTM models training setup, the data was further pre-processed into sequences, where each sample in such train data is a sequence of chronological samples with sequence length 7. Each sample in the sequence was concatenated with the label (profit/pips) of the sample from the previous time step. Concretely, the sample sequence of a single RNN/LSTM sample is given below. The objective is to predict the label (profit/pips) of the seventh data in the sequence. (x1, y0), (x2, y1),... (x7, y6). This concatenation makes each element in the sequence input to the RNN and LSTM models become a feature vector of length 24 (i.e. the original 23 feature vectors plus an additional label from the previous time step).

For the testing of the models, 80% data was used as train data while 20% data was used as test data.

5. Results and Discussion

MLP, RNN and LSTM deep learning models were applied to three different traders with different trading characteristics. The traders are tagged, Trader A, Trader B and Trader C.

5.1. Training Prediction for the Three Traders

After the training was completed, the models were used to predict the profit (pips) of the training data itself and the result are shown in Figure 4 for trader B. The plots show that all the three models were able to produce good predictions, approximately matching the actual trader data (label or profits) for each trader (A, B and C) on which they were trained. Therefore, the results demonstrated that the models were able to capture the underlying statistical distribution of the training data. However, a major goal of machine learning is to test the generalization capabilities (of the trained models) to data unseen during training (i.e., the test data) [42]. The results for the test data evaluation are discussed in the subsequent sections.

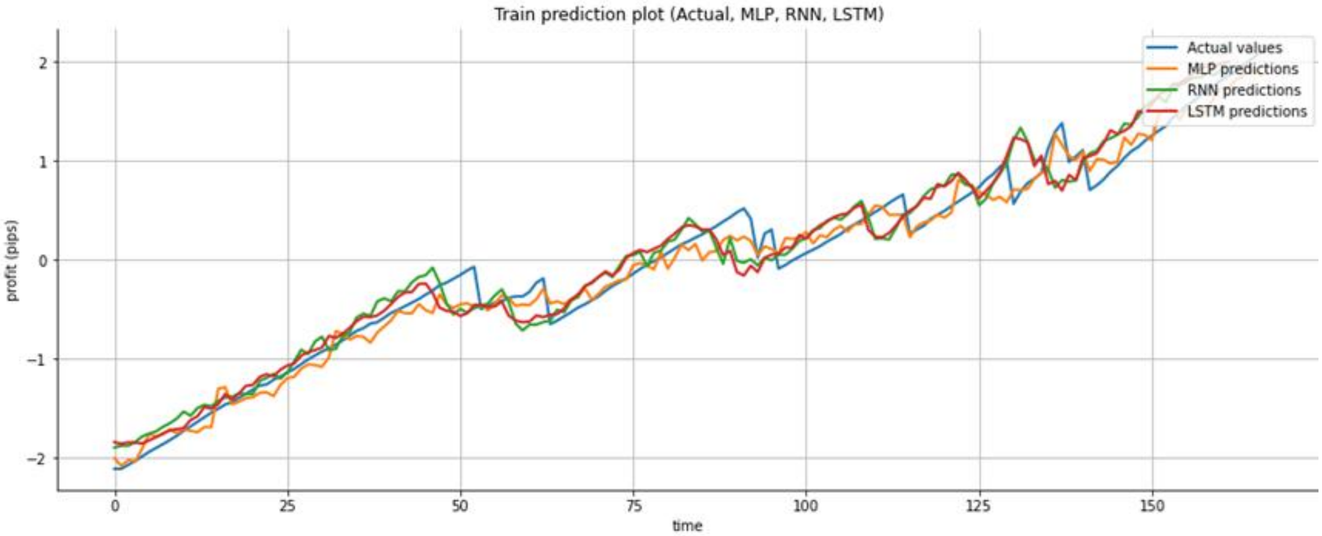


Figure 4. Actual values and trained prediction plots for Trader B.

5.2. Test Data Prediction for the Tree Traders

In order to determine how well the deep learning models generalize from training data to test data, the trained models were used to predict the output of both the train and test data for each trader. Figure 5, Figure 6 and Figure 7 show the trained and test data prediction plots for trader A, trader B and trader C.

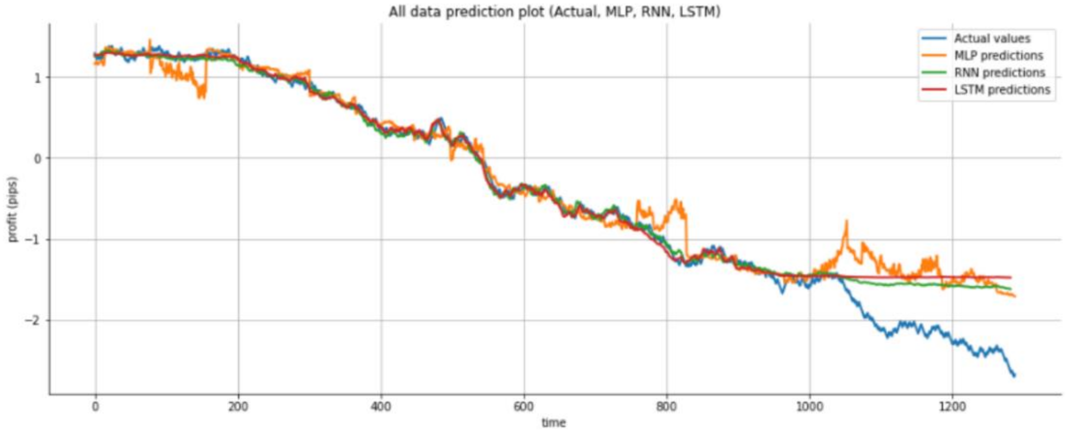


Figure 5. Trained and test data prediction plots for trader A.

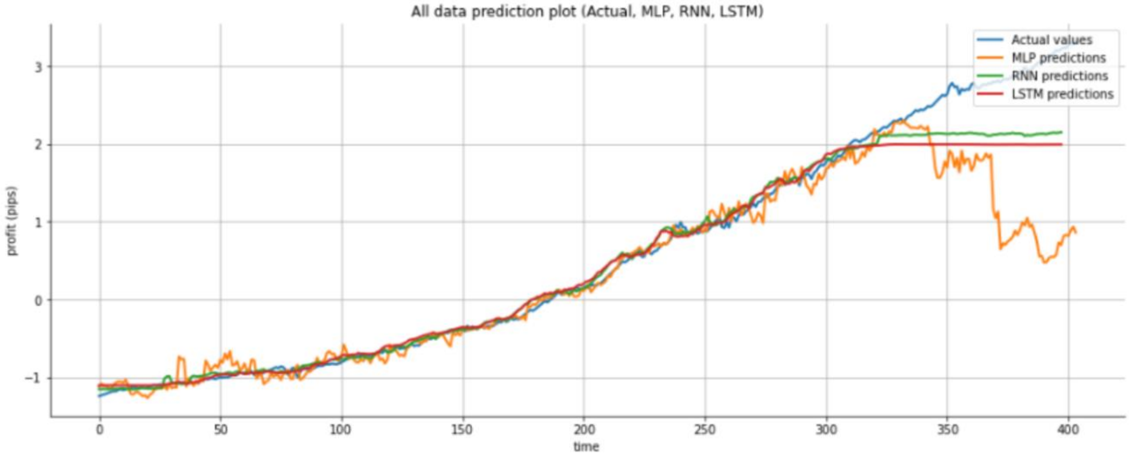


Figure 6. Trained and test data prediction plots for trader B.

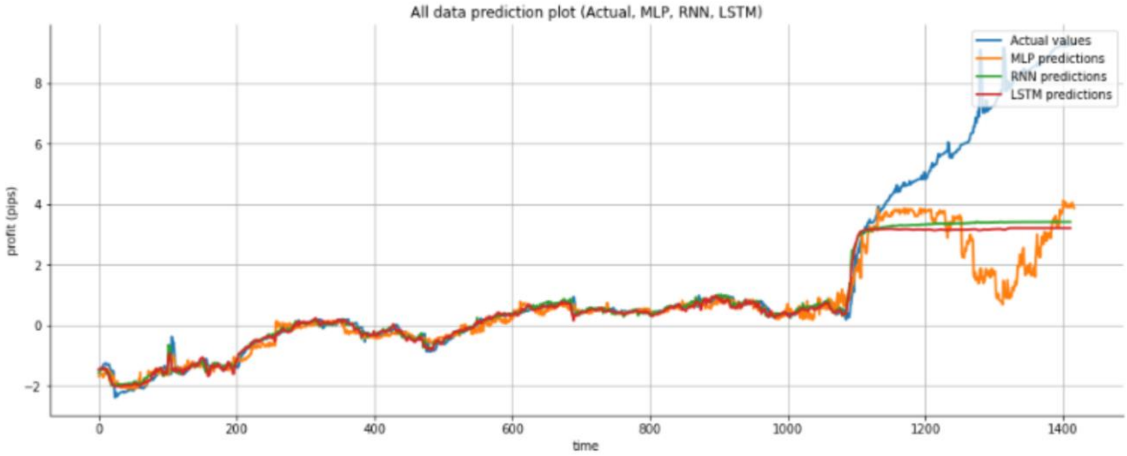


Figure 7. Trained and test data prediction plots for trader C.

Comparing the curves for the predicted and actual data, it can be observed from Figure 5, Figure 6 and Figure 7 that the predictions of the model starts veering off from the actual label towards the end of the plots, the region where the model encounters the test data. This shows that the three deep learning models demonstrate various degrees of generalization from the training data (the earlier regions of the plots) to the test data (the later regions of the plot).

Table 2. Performance statistics of the MLP, RNN and LSTM for the three traders.

	Trained Data			Test Data		
Trader/Model	**RMSE**	**MAPE**	R^2	**RMSE**	**MAPE**	R^2
A/MLP	0.1532	0.2114	0.9765	0.9285	0.3935	-10.0325
A/RNN	0.0500	0.0674	0.9975	0.5836	0.2407	-3.4326
A/LSTM	0.5050	0.0902	0.9974	0.7075	0.2949	-5.5135
B/MLP	0.0987	0.1270	0.9903	1.5025	0.4254	-23.2741
B/RNN	0.0444	0.0701	0.9980	0.6565	0.2099	-3.7806
B/LSTM	0.0400	0.0562	0.9984	0.7890	0.2591	-5.9056
C/MLP	0.1898	0.6375	0.9640	4.5808	0.5211	-5.6889
C/RNN	0.1014	0.1940	0.9897	3.7826	0.4646	-3.6175
C/LSTM	0.1067	0.3320	0.9886	3.9640	0.4928	-4.0711

Three performance metrics: root mean squared error (RMSE), mean absolute percentile error (MAPE) and R-squared were applied to ascertain the various degrees of generalization [43-45]. Table 2 shows the performance metrics for MLP, RNN and LSTM for trained data and test data for the three traders.

6. Conclusions

Many previous applications of deep learning to the forecasting of the prices in the financial market focused on the employment of time series historical data for trend predictions. As a deviation from the previous approaches, this study takes the results of the performance of traders with diverse trading strategies and conditions as the input data set. These new constraints make this study unique. Recurrent neural network, long short-term memory and multilayer perceptron deep learning models were applied for the prediction of three traders' performance. Comparing the three deep learning models, RNN produced results that are closest to the actual data. LSTM follows RNN in the accuracy of results while MLP performed least for the prediction of traders' profitability. Future work shall focus on investigating the effect of various modifications of RNN and LSTM models on traders' performance.

Abbreviations

MC	Machine Learning
NN	Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
DMLP	Deep Multilayer Perceptron Models
DL	Deep Learning
RMSE	Root Mean Squared Error
MAPE	Mean Absolute Percentile Error

Acknowledgments

The good effort of Mrs. Stella Oyemade who proofread the script is acknowledged.

Author Contributions

David Ademola Oyemade: Conceptualization, Data curation, Project administration, Resources, Writing – original draft, Writing – review & editing

Eseoghene Ben-Iwhiwhu: Methodology, Formal Analysis, Writing – review & editing, Investigation, Visualization

Funding

Only the personal funds of the authors were used for this work.

Data Availability Statement

The data supporting the outcome of this research work has been reported in this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Oyemade, D. A., Ojugo, A. A. An optimized input genetic algorithm model for the financial market. International Journal of Innovative Science, Engineering & Technology. 2021, 8 (2), 408-419.

[2] Abdul Lateef A. A., Sufyan, T., Al-Janabi, S. T. F., Al-Khateeb, B. Survey on intrusion detection systems based on deep learning. Periodicals of Engineering and Natural Sciences 2019. 7 (3), pp. 1074-1095.

[3] Sezer, O. B., Gudelek, M. U., Ozbayoglu, A. M. Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. Applied Soft Computing Journal. 2020, 90 (106181), pp. 1-27. https://doi.org/10.1016/j.asoc.2020.106181

[4] Goodfellow, I., Bengio, Y., Courville, A. Deep Learning; 2016, MIT Press.

[5] Cybenko, G. Approximation by superpositions of a sigmoidal function. Math. Control Signals Systems. 1989, 2 (4), pp. 303–314.

[6] Kalman, B. L., Kwasny, S. C. Why tanh: choosing a sigmoidal function. In Proceedings of IJCNN International Joint Conference on Neural Networks, Baltimore, MD, USA. 1992; pp. 578–581.

[7] Nair, V., Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning. 2010; pp. 807–814.

[8] Maas, A. L., Hannun, A. Y., Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning, Atlanta. 2013; pp. 3.

[9] Bengio Y., Simard, P., Frasconi, P. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks. 1994, 5(2), pp. 157–166.

[10] Hochreiter, S., Schmidhuber, J., Long short-term memory. Neural Computation. 1997, 9(8), pp. 1735–1780.

[11] Greff, K., Srivastava, R. K., Jan, K. J., Steunebrink, B. R., Schmidhuber, J. LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems. 2016, 28 (10), pp. 2222–2232.

[12] Kasongo, S. M., Sun, Y. A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access. 2019, 7 (2019), pp. 38597–38607.

[13] Tariq, M. I., Memon, N. A., Ahmed, S., Tayyaba, S., Mushtaq, M. T. et al. A review of deep learning security and privacy defensive techniques. Hindawi Mobile Information Systems, 2020, 2020, pp. 18.

[14] Sehovac, L., Grolinger, K. Deep learning for load forecasting: sequence to sequence recurrent neural networks with attention. IEEE Access. 2020, 8 (2020), pp. 36411-36426.

[15] Joseph, J., Aswin, R., James, A., Johny, A., Jose, P. V. Smart waste management using deep learning with IoT. International Journal of Networks and Systems. 2019, 8 (3), pp. 37–40.

[16] Gayathri, M., Meghana, M., Trivedh, M., Manju, D. Suspicious activity detection and tracking through unmanned aerial vehicle using deep learning techniques. International Journal of Advanced Trends in Computer Science and Engineering. 2020, 9 (3), pp. 2812-2816. https://doi.org/10.30534/ijatcse/2020/51932020

[17] Fang, X., Xu, M., Shouhuai, Xu, S., Zhao, P. A deep learning framework for predicting cyber attacks rates. EURASIP Journal on Information Security. 2019, 5, pp. 11. https://doi.org/10.1186/s13635-019-0090-6

[18] Vogl, M., R ötzel, P. R., Homes, S. Forecasting performance of wavelet neural networks and other neural network topologies: A comparative study based on financial market data sets. Machine Learning with Applications. 2022, 8 (100302), 1-13. https://doi.org/10.1016/j.mlwa.2022.100302

[19] Schmidt-Kessen M. J., Eenmaa, H, Mitre, M. Machines that make and keep promises - Lessons for contract automation from algorithmic trading on financial markets. Computer Law and Security Review. 2022, 46 (105717), 1-17. https://doi.org/10.1016/j.clsr.2022.105717

[20] Zhang, H. Optimization of risk control in financial markets based on particle swarm optimization algorithm. Journal of Computational and Applied Mathematics. 2020, 368 (112530), 1-12. https://doi.org/10.1016/j.cam.2019.112530

[21] Dash, R. Performance analysis of an evolutionary recurrent legendre polynomial neural network in application to forex prediction. Journal of King Saud University – Computer and Information Sciences. 2020, 32, 1000-1011. https://doi.org/10.1016/j.jksuci.2017.12.010

[22] Supsermpol, P., Huynh, V, N., Thajchayapong, S., Chiadamrong, N. Predicting financial performance for listed companies in Thailand during the transition period: A class-based approach using logistic regression and random forest algorithm. Journal of Open Innovation: Technology, Market, and Complexity. 2023, 9 (100130), 1-16. https://doi.org/10.1016/j.joitmc.2023.100130

[23] Ueda, K., Suwa, H., Yamada, M., Ogawa, Y., Umehara, E. et al. SSCDV: Social media document embedding with sentiment and topics for financial market forecasting. Expert Systems with Applications. 2024, 245 (122988), 1-16. https://doi.org/10.1016/j.eswa.2023.122988

[24] Oyemade, D. A., Ekuobase, G. O., Chete, F. O. Fuzzy logic expert advisor topology for foreign exchange market. In Proceedings of the International Conference on Software Engineering and Intelligent Systems, Covenant University, Otta, Nigeria. 2010; pp. 215–227.

[25] Bevilacqua, M., Tunaru, R., Vioto, D. Options-based systemic risk, financial distress, and macroeconomic downturns. Journal of Financial Markets. 2023, 65 (100834), pp. 1-35. https://doi.org/10.1016/j.finmar.2023.100834

[26] Bossaerts, F., Yadav, N., Bossaerts, P., Nash, C., Todd, T. et al. Price formation in field prediction markets: The wisdom in the crowd. Journal of Financial Markets. 2024, 68 (100881), pp. 1-16. https://doi.org/10.1016/j.finmar.2023.100881

[27] Allenotor, D., Oyemade, D. A. A price-based grid resources pricing approach for non-storable real assets. Journal of Advances in Mathematical & Computational Science. 2022, 10 (2), 1-18. https://dx.doi.org/10.22624/AIMS/MATHS/V10N2P1

[28] Allenotor, D., Oyemade, D. A. An optimized parallel hybrid architecture for cryptocurrency mining. Computing, Information Systems, Development Informatics & Allied Research Journal. 2022, 12 (1), 94-104. https://doi.org/10.22624/AIMS/CISDI/V12N1P10

[29] Oyemade, D. A., Enebeli D. A dynamic level technical indicator model for oil price forecasting. Global Journal of Computer Science and Technology, 2021, 21 (1), 5-14. https://doi.org/10.34257/GJCSTGVOL21IS1PG5

[30] Oyemade DA, Allenotor D. A quality of service (QoS) model for forex brokers' platforms. International Journal of Innovative Science, Engineering & Technology, 2022, 9 (6), 123-132.

[31] Gündüz, H., Zehra, C. ¸ Çataltepe, Z., Yaslan Z. Stock daily return prediction using expanded features and feature selection. Turkish Journal of Electrical Engineering and Computer Sciences. 2017, 25 (6), 4829-4840. https://doi.org/10.3906/elk-1704-256

[32] Altuner, A. B., Kilimci, Z. H. A novel deep reinforcement learning based stock price prediction using knowledge graph and community aware sentiments. Turkish Journal of Electrical Engineering and Computer Sciences 2022; 30 (4), pp. 1506 – 1524. https://doi.org/10.55730/1300-0632.3862

[33] Oyemade, D. A., Allenotor, D. FAITH software life cycle model for forex expert advisors. Journal of Advances in mathematical and Computational Sciences 2021; 9 (1): 1-12.

[34] Oyemade, D. A. A typified greedy dynamic programming model for the metatrader platform. Journal of Advances in Mathematical and Computational Sciences. 2020, 8 (3): 49-60.

[35] Oyemade, D. A., Ojugo, A. A. A property oriented pandemic surviving trading model. International Journal of Advanced Trends in Computer Science and Engineering. 2029, 9 (5), 7397-7404. https://doi.org/10.30534/ijatcse/2020/71952020

[36] Oyemade, D. A., Allenotor, D. A Trade gap scalability model for the forex market. In IEEE 11th International Conference on Ubiquitous Intelligence & Computing and IEEE 11th International Conference on Autonomic & Trusted Computing; Washington, DC, USA. 2014; pp. 867 – 873.

[37] Dezsi, E., Nistor, I. A. Can deep machine learning outsmart the market? A comparison between econometric modelling and long- short term memory. Romanian Economic and Business Review. 2016, 11 (41), pp. 54–73.

[38] Almeida, B. J., Neves, R. F., Horta, N. Combining support vector machine with genetic algorithms to optimize investments in forex markets with high leverage. Applied Soft Computing. 2018, 64 (2018), pp. 596–613.

[39] Nia, L., Li, Y., Wang, X., Zhaing, J., Yud, J. Forecasting of forex time series data based on deep learning. Procedia Computer Science. 2019, 147 (2019), pp. 647–652.

[40] Carapuço, J., Neves, R., Horta, N. Reinforcement learning applied to forex trading. Applied Soft Computing Journal. 2018, 73 (2018), pp. 783–794.

[41] Kingman, D. B., Ba, L. J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations; San Diego, CA, USA. 2015; pp. 13.

[42] Ballester, R., Clemente, X. A., Casacuberta, C., Madadi, M., Corneanu, C. A., Sergio Escalera, S. Predicting the generalization gap in neural networks using topological data analysis, Neurocomputing. 2024, 596(127787), pp. 1-14. https://doi.org/10.1016/j.neucom.2024.127787

[43] Jia, D., Liwei Yang, L., Lv, T., Weiping Liu, W., Gao, X., Zhou, J. Evaluation of machine learning models for predicting daily global and diffuse solar radiation under different weather/pollution conditions. Renewable Energy. 2022, 187, pp. 896-906. https://doi.org/10.1016/j.renene.2022.02.002

[44] McHale, I. G., Holmes, B. Estimating transfer fees of professional footballers using advanced performance metrics and machine learning. European Journal of Operational Research. 2023, 306(1), pp. 389-399. https://doi.org/10.1016/j.ejor.2022.06.033

[45] Jain, P., Islam, M. T., Ahmed S. Alshammari, A. S. Comparative analysis of machine learning techniques for metamaterial absorber performance in terahertz applications. Alexandria Engineering Journal. 2024, 103, pp. 51-59. https://doi.org/10.1016/j.aej.2024.05.111

Biography

David Ademola Oyemade is an Associate Professor of Computer Science at the Federal University of Petroleum Resources, Effurun, Delta State, Nigeria. He holds a PhD degree in Computer Science obtained from the University of Benin, Benin City, Nigeria in 2014. He also holds M.Sc. degree in Computer Science obtained from the University of Benin, Benin City, Nigeria in 2007 and a postgraduate diploma in Computer Science obtained from the University of Benin, Benin City, in 2004. He is a life member of Nigeria Computer Society (NCS) and a professional member of Association for Computing Machinery (ACM). He has served as a lecturer in the Department of Computer Science, Federal University of Petroleum Resources, Effurun and rose through various ranks. He has supervised several students at undergraduate and postgraduate levels at the department of Computer Science, Federal University of Petroleum Resources, Effurun. He has many articles in international and local journals. His research area is Software Engineering, Software Architecture, Intelligent Systems and financial market algorithms and modelling.

Eseoghene Ben-Iwhiwhu holds a PhD degree from Loughborough University, United Kingdom (UK). He also holds an M.Sc. and B.Sc. degrees from Coventry University, UK and the Federal University of Petroleum Resources Effurun. He is an assistant lecturer in the department of Computer Science, Federal University of Petroleum Resources Effurun. His research interest includes machine learning, deep learning, reinforcement learning, Hebbian learning and neuroevolution.

Research Field

David Ademola Oyemade: Software Engineering, Software Architecture, Intelligent Software Systems, Financial Market Algorithms, Deep Learning.

Eseoghene Ben-Iwhiwhu: Machine Learning, Deep Learning, Reinforcement Learning, Hebbian Learning and Neuroevolution.