Karbala International Journal of Modern Science

Volume 10 | Issue 1 Article 6

Multi-Agent System For Portfolio Profit Optimization For Future Stock Trading

Usha Devi
Department of Computer Science and Engineering, National Institute of Technology Tiruchirappalli, Tuvakudi, India., 406915052@nitt.edu

Mohan R
Department of Computer Science and Engineering, National Institute of Technology Tiruchirappalli, Tuvakudi, India.

Multi-Agent System For Portfolio Profit Optimization For Future Stock Trading

Abstract

Stock trading highly contributes to the economic growth of the country. The stock trading objective is to earn profits with buy/sell/hold decisions on the set of stocks in the portfolio. The portfolio optimization problem is finding the decision sequence that leads to higher profit and lower risk. Portfolio optimization is challenging due to complex price history patterns and an uncertain environment. Incorrect decisions in stock trading lead to massive losses. The proposed Multi-Agent System for Portfolio Profit Optimization (MASPPO) aims to optimize trading profit and reduce risk with accurate predictions. The proposed model integrates the Fuzzy c-means with the Deep reinforcement learning model. The experimental datasets contain stock price history with 14,562 records. The MASPPO model maximizes the portfolio profit, intending to reduce the error. The proposed model, MASPPO, showed a Root mean squared error of 9.48 and a Mean absolute error of 2.63 and outpaced the recent models in the literature. The results proved that MASPPO maximizes the portfolio profit and is reliable.

Keywords

Bi-LSTM, Fuzzy c-means, Reinforcement Learning, Stock.

Cover Page Footnote

Not applicable

RESEARCH PAPER

Multi-Agent System for Portfolio Profit Optimization for Future Stock Trading

N.S.S.S.N. Usha Devi*, R. Mohan

Department of Computer Science and Engineering, National Institute of Technology Tiruchirappalli, Tuvakudi, India

Abstract

Stock trading highly contributes to the economic growth of the country. The stock trading objective is to earn profits with buy/sell/hold decisions on the set of stocks in the portfolio. The portfolio optimization problem is finding the decision sequence that leads to higher profit and lower risk. Portfolio optimization is challenging due to complex price history patterns and an uncertain environment. Incorrect decisions in stock trading lead to massive losses. The proposed Multi-Agent System for Portfolio Profit Optimization (MASPPO) aims to optimize trading profit and reduce risk with accurate predictions. The proposed model integrates the Fuzzy c-means with the Deep reinforcement learning model. The experimental datasets contain stock price history with 14,562 records. The MASPPO model maximizes the portfolio profit, intending to reduce the error. The proposed model, MASPPO, showed a Root mean square error of 9.48 and a Mean absolute error of 2.63 and outpaced the recent models in the literature. The results proved that MASPPO maximizes the portfolio profit and is reliable.

Keywords: Bi-LSTM, Fuzzy c-means, Reinforcement learning, Stock

1. Introduction

In recent years, portfolio profit optimization has captured attention as a means of earning high profits and escaping risk. Portfolio optimization involves predicting the decisions on the quantities of stocks to hold/buy/sell that depend on the stock price prediction models. Many researchers developed stock price prediction models by analyzing datasets captured from online price history, Twitter [1], news, and social groups. Among these, the most prominent approach is historical price analysis, which contains day-to-day trading transaction details. The stock historical price data is a non-linear time series with high variance, making forecasting a challenging problem [2].

Deep Neural Networks (DNN) [3] captured attention as dynamic computing models in stock value prediction. DNN is a simulation of a neural network in the human brain that makes a decision based on available knowledge. DNN applications are growing in a variety of domains. There are two popular basic DNN models: Convolutional Neural Network (CNN) [4] and Recurrent Neural Network (RNN) [5]. CNN has many applications in feature extraction, image processing, and text analytics. There are two subcategories of RNN: Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) [6]. These models are widely used in time series analysis and sequence modelling. Due to the simple architecture, the GRU is faster when compared with LSTM. Various hybrid DNNs [7] are emerging to solve problems in multiple fields. Creating a suitable DNN for the given problem requires great analysis and rigorous experimentation.

The stock market environment is uncertain [8]; hence, learning the environment with experience using Reinforcement learning (RL) [9] can frame the solution to the portfolio profit optimization problem. RL mimics human brain thinking in finding a

Received 17 June 2023; revised 30 November 2023; accepted 3 December 2023.
Available online 30 January 2024

* Corresponding author.
E-mail address: 406915052@nitt.edu (N.S.S.S.N. Usha Devi).

https://doi.org/10.33640/2405-609X.3337

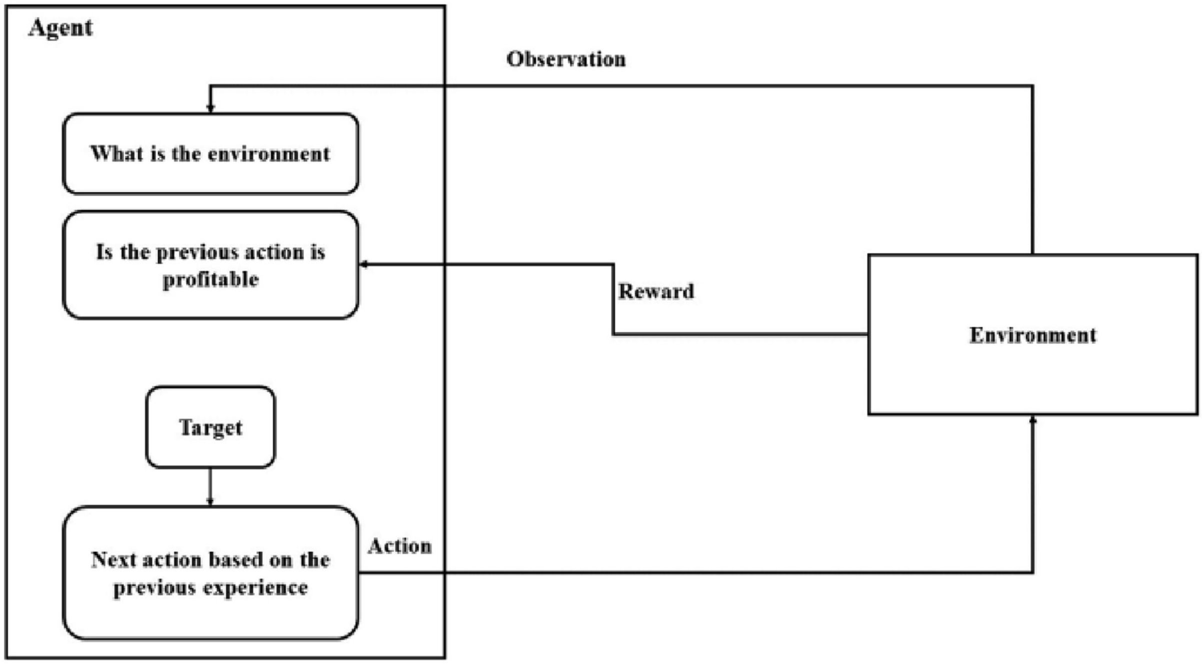

Fig. 1. Reinforcement learning approach.

possible decision based on previous actions and individual results. Fig. 1 shows the RL method learned by an artificial agent. The agent environment is represented by a collection of states and corresponding actions. In RL, the artificial agent follows episode-wise learning of the environment. Each episode describes multiple steps taken by the agent. The agent's every move in the environment generates a positive or negative reward. Agents' good experiences, like huge profits, are noted as positive rewards, which is a preferable movement.

On the other hand, bitter experiences, like less profitability and more expenditure, are noted as negative rewards. The agent tries to avoid actions that lead to negative rewards to receive good rewards in future learning.

A Multi-Agent System (MAS) consists of multiple agents learning the environment with a policy. The design of the learning policy depends on the type of problem the MAS will solve. In this context, a portfolio profit maximization model [10] is required to optimize the profit and prediction error. The complexity of analyzing the stock price pattern and the dynamism in deep reinforcement learning motivates the proposed approach.

The proposed system combines Deep reinforcement learning with Fuzzy c-means to reduce the overfitting of

DNN. Fuzzy c-means a soft clustering [11] that learns through unlabeled data where each data point is recognized with a degree of membership with the cluster. The strength of the belongingness is in proportion with the amount of degree.

The following are the objectives of the present work.

1. Develop an accurate portfolio optimization model with real data sets.
2. Eliminate overfitting DL with Fuzzy c-means.
3. Identify a suitable method to adopt Reinforcement learning with MAS for trading decisions.
4. Evaluate and compare the models in the literature and the proposed models using the following metrics.
   (i) Domain-specific: Annual Volatility (AV) and Compound Annual Growth Rate (CAGR).
   (ii) General regression: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

The organization of the consequent sections is as follows. Section 2 describes the related work. Section 3 illustrates the theoretical background and experimentation. The experiment results and discussion are explained in section 4. Section 5 interprets the conclusion and future works.

2. Related work

The LSTM is proven as a real-time dynamic time series predictive model compared with other machine learning models like SVM, variations of the Regression model, KNN, and statistical models ARIMA and ARMA [12]. Bi-LSTM (Bidirectional

Long Short Term Memory) [13] is an improved version of LSTM with two-way processing capability. The Bi-LSTM learns the time series data accurately but is a little slower than LSTM due to the two-way learning. An adaptive Neuro-Fuzzy inference system [14] proved that integrating Fuzzy logic improves the neural network's learning. In the present work, accuracy is considered the critical metric. Hence, the Bi-LSTM was chosen as a time series predictive model. Fuzzy c-means clustering is used in data preparation to make the model learning easy to deal with uncertainty.

RL is a popular predictive model in which an agent learns in an environment and makes decisions through experience. The agent's goal is to optimize a specific objective. In MAS, the system has multiple objectives. NENDA3 is a portfolio selection model with Q-learning [15]. The model has shown considerable performance in maximizing the portfolio profit with one stock. A DELM model improves the portfolio by recommendations on the stocks selling [16]. DELM is an Extreme Machine Learning (EM) model developed based on Discrete Wavelet Transformation. This model uses K-means clustering to create distance-based stock groups to predict the stock's trend; however, K-means is unsuitable for capturing the uncertainty.

Another portfolio selection model is BPSM, which uses EM. BPSM follows behavior-based trend prediction using DL with text analytics. A multi-objective model, GAPS [17] maximizes the portfolio by computing each decision's risk and return values so that the portfolios selected have low risk and high return. Shruti Mittal et al. proposed a medium and long-term investment prediction model using Fuzzy sets [18], a rule-based stock recommendation model exploring various decisions. An EM model, Random forest [19], performs portfolio allocation, achieving considerable returns.

Integrating RL and deep neural networks is an appropriate model for forecasting time series [20]. The RPGA is a deep reinforcement learning model [21] that combines RL and LSTM for the forecasting of portfolio profits. Integrating stacked deep learning and reinforcement learning [22] is also popular as stock portfolio prediction models. A Multi DQN [23] model consists of a multiple agents learning environment combining deep and Q learning. The Q value produced by the deep learning model determines the agent's action. Bellman's equation [24] is a prominent optimization function for a deep reinforcement learning approach to optimize the benefit of the agent movement. The learning model must be evaluated with the three possible stock trends using both domain-specific and general regression metrics [25] for proper evaluation.

The proposed model, MASPPO, integrates the Fuzzy c-means with Bi-LSTM to deal with overfitting. The MAS learning was formulated to optimize the profit functions based on decision constraints. The MAS follows constraint−based optimization with Q learning. Evaluation metrics of the predictive models play a vital role in assessing real-time performance. The proposed research focused on finding suitable metrics for portfolio optimization models.

3. Theory and experimentation

The architecture of the MASPPO is shown in Fig. 2. The collected historical stock data is preprocessed and then clustered using fuzzy c-means. Bi-LSTM learns these clusters to generate an optimized Q value with the Bellman Equation (1). Multiple agents respond to these Q values to decide on buying, selling, or holding stocks.

3.1. The workflow of MASPPO

The workflow of MASPPO activities is presented in Fig. 3. The collected input datasets were preprocessed by removing null entries and then divided into training and testing datasets. The Fuzzy c-means cluster the dataset as a data preparation method to prepare the data for learning by the Bi-LSTM model. The deep reinforcement learning modules run with an initialization. The initialization includes the state and action entries with zeros. The default decision is *Hold*. Then, the Bi-LSTM is trained, tested, and ready for the prediction. For each forecast, compute the new state and the action using *NewQ(S,A).* This process continues till the end of the episodes. The model evaluation is done using domain-specific and general regression metrics.

3.2. Bidirectional long short term memory

Bi-LSTM is a DL model shown in Fig. 3. This is an extension of LSTM with a two-way processing policy. In processing, it saves one step forward and one step backward information. The proposed system implements a Bi-LSTM structure consisting of four layers: one input layer, one output layer, and two hidden layers. The softmax was selected as the activation function to speed up the processing. Here $\overrightarrow{h_{t-1}}$ indicates the output obtained in a forward pass in the learning and $\overleftarrow{h_{t+1}}$ indicates the output in

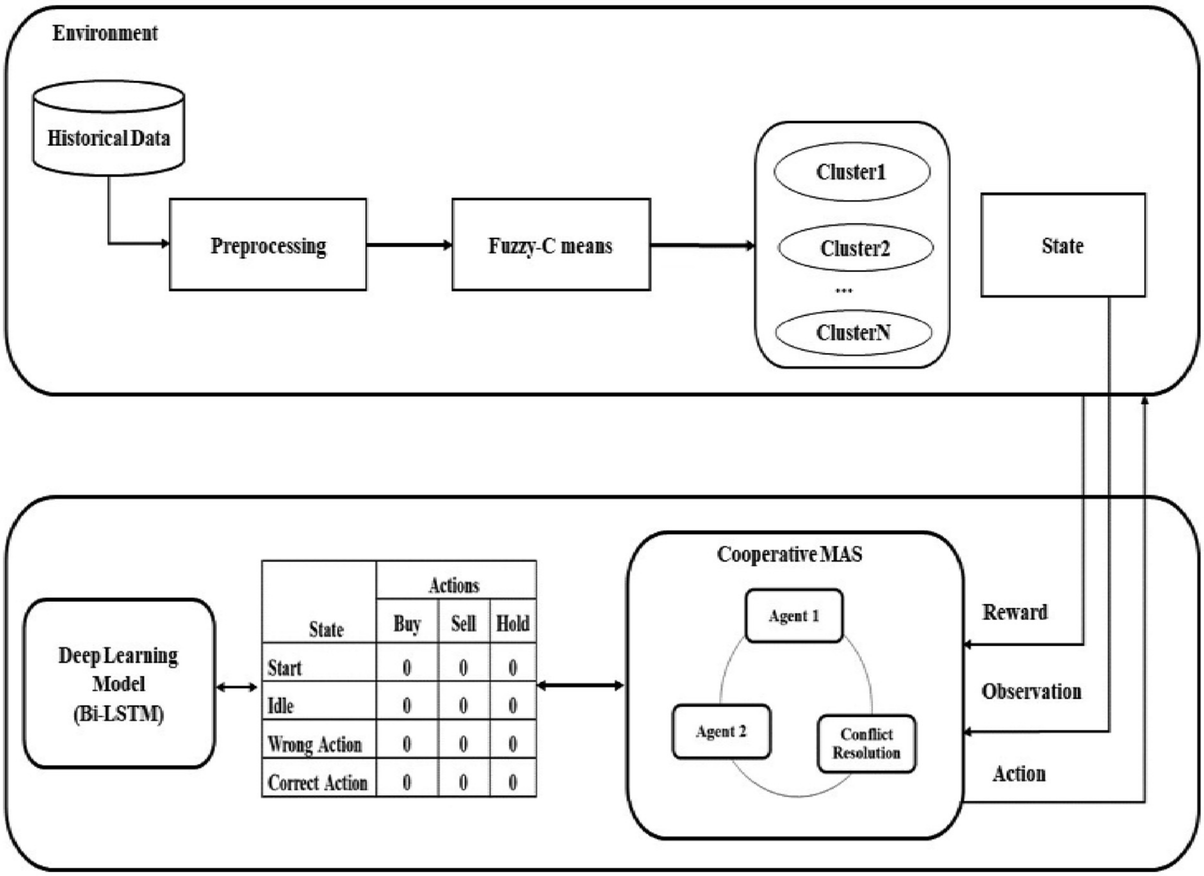

Fig. 2. The architecture of the Multi-agent system for portfolio profit optimization.

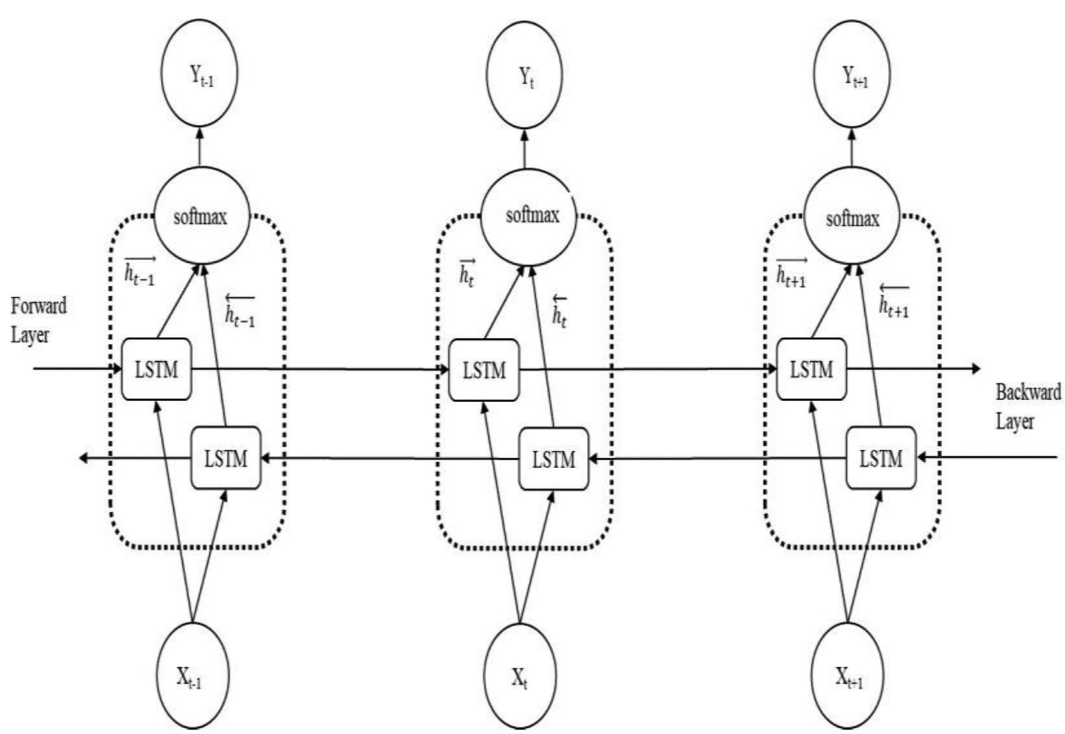

Fig. 3. The layered architecture of Bi-directional Long Short Term Memory.

the backward pass. The popular function Adam is chosen as optimization. Now, the model is ready for Deep reinforcement learning.

3.3. Multi-agent system (MAS)

The MAS objective is to predict a stock value and achieve maximum portfolio profit. Fig. 4 illustrates the learning environment of the MAS. The objectives of Agent 1 (A1) and Agent 2 (A2) are different. However, they cooperatively work to achieve a global optimization function. A1 handles the trading decision Buy as based on the trader's investment. A2 makes decisions to Sell and Hold. In case of overlapping decisions, the conflict resolution module prioritizes the decision; the order of priority is Buy, Sell, and Hold. For example, suppose A1 and A2 make conflicting decisions on a stock. In that case, the conflict resolution module prioritizes the decision and returns Buy. The advantage of this module is that it allows you to run MAS without ambiguous decisions. So, at every moment, MAS can make a unique decision on a stock.

The environment of the MAS can be seen as order pairs of state and action. Agents learn this environment learns episode-wise. The agents produce a sequence of steps for each episode run. Agent learning starts with observation. It receives a positive/negative reward for every action. Agent must move towards positive rewards to optimize the target. The mathematical model for agent learning is given in equation (1).

$$NewQ(S,A) = Q(S,A) + \alpha\left[R(S,A) + \gamma\, MaxQ^{l}\left(S^{l},A^{l}\right) - Q(S,A)\right] \tag{1}$$

α: Determines the agent's learning rate.

Q(S, A): The Q value at state, *S* for a specific action, *A*

γ: Immediate reward representing the discount rate.

R (S, A): Reward obtained by action *A* at state *S*.

$MaxQ^{l}(S^{l},A^{l})$: Highest expected future reward by estimated Action A^{l} at estimated state S^{l}

3.4. Deep reinforcement learning

In episode-wise agent learning, the deep reinforcement learning objective is to optimize the Q-value to make a decision. In each episode, the agent may decide to stay in the same state or transition to the next state based on the Q-table. The following procedure demonstrates the deep reinforcement learning approach to obtaining maximum trading profit with an optimized Q-value.

1. Assign zeros to all the (State, Action) entries.
2. Train the Bi-LSTM.

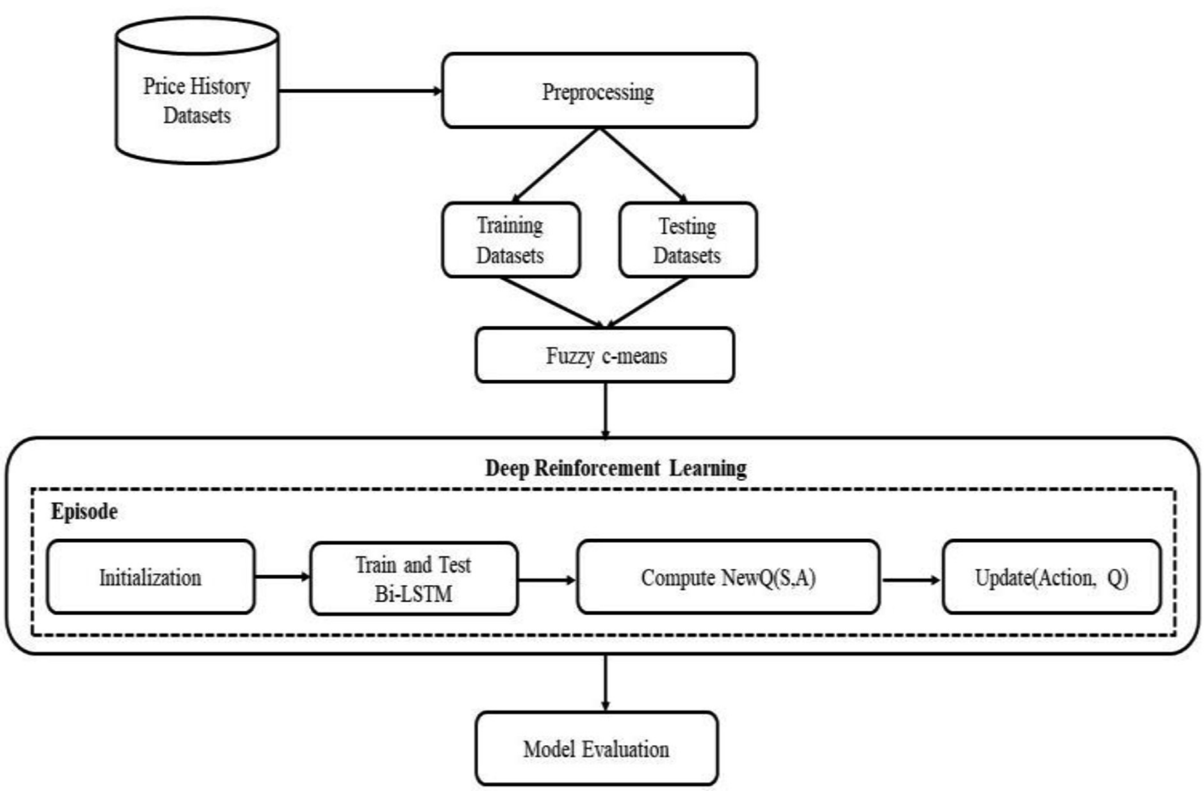

Fig. 4. The workflow of the activities in the MASPPO.

3. Compute Bellman's equation using (1).
4. Obtain Q value and update the entries (Action, Q).
5. Until all entries in the table are filled in, or all episodes are completed, repeat step 3.

3.5. Problem formulation

The portfolio consists of set of stocks $\{s_1, s_2, s_3 \ldots s_k\}$, with costs $\{c_1, c_2, \ldots c_k\}$, each stock might have different quantities say $\{n_1, n_2, \ldots n_k\}$. Consider the following linear equations (2) and (3) for trading profit at time $t + 1$.

MASPPO aims to maximize the $P(t + 1)$ by taking one of three decisions: *Hold*, *Buy*, and *Sell* as per the investor amount with a target ≥ 5 % profit. Here, the investors can make the *Sell* and *Hold* decisions. But to *buy* stocks, the investment needs to be considered. Agent 1 (A1) optimizes the H_{it} and S_{it} in the cooperative MAS environment, and Agent 2 (A2) optimizes the B_{it}. The artificial agent learning will proceed with the following assumptions.

Assumption 1. The agents are finding the solution in a cooperative environment that is fully observable, sequential, and stochastic.

Assumption 2. The agent's decisions, Buy, Hold, and Sell, are discrete.

Assumption 3. The order of priority for the decisions as *Buy*, *Sell*, and Hold to deal with conflict issues.

Remarks 1. System equation (2) and equation (3) are different from equation (1), and they solve another optimization problem.

$$P(t+1) = I(t) + H_{it} + S_{it} \tag{2}$$

$$I(t) = \sum B_{it}(n_{it}, c_{it}) = n_{1t}c_{2t} + n_{2t}c_{2t} + n_{3t}c_{3t} + \ldots + n_{kt}c_{kt} \tag{3}$$

4. Results and discussion

In this work, we have experimented with portfolio optimization using proposed and existing models. The steps in the experimentation and the discussion on the performance of the models are as follows.

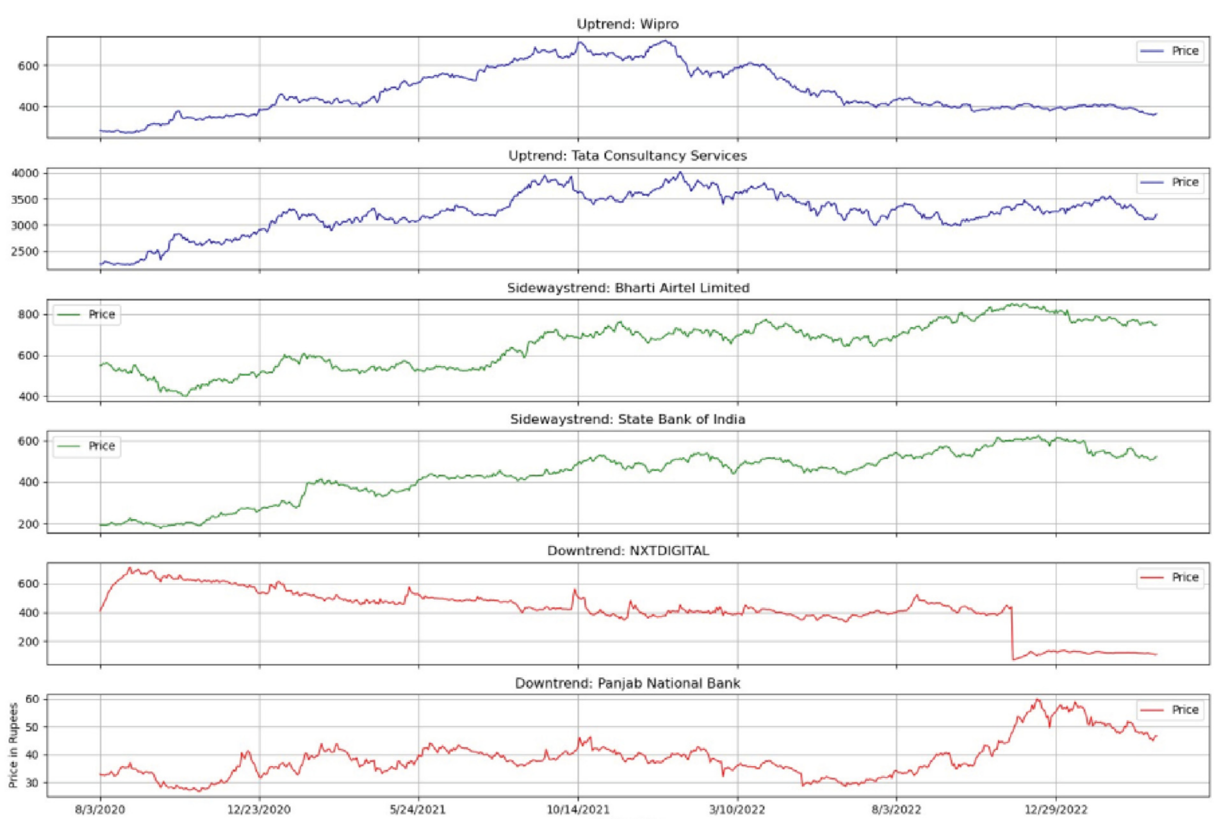

Fig. 5. Day-wise stock price in rupees for the six input stocks.

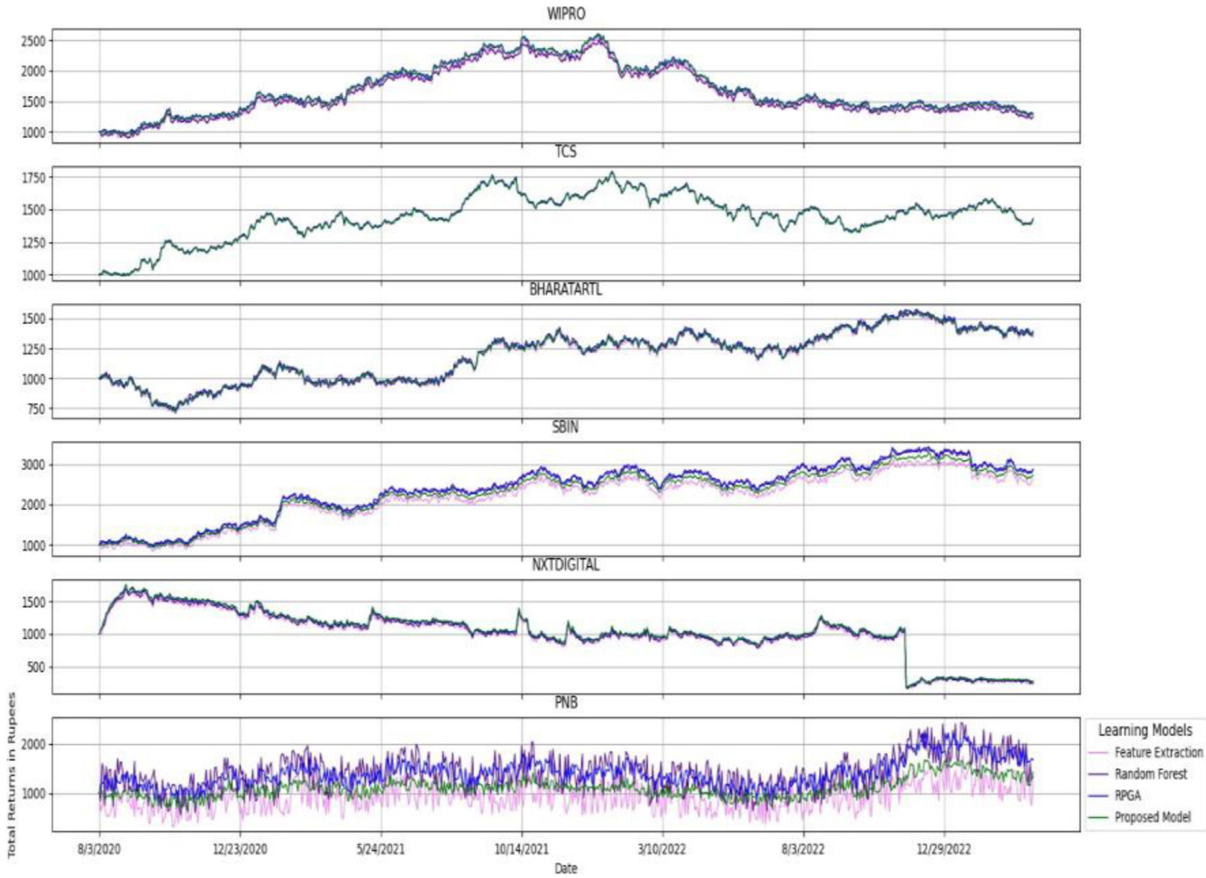

Fig. 6. The prediction of total returns by learning models for the stock datasets.

4.1. Data framing for the experimentation

The experimental data consists of day-to-day stock trading history collected from 8th August 2016 to 31st March 2023. These datasets were real, and they were collected from online portals. The stocks WIPRO, TCS, BHARTARTL, SBIN, NXTDIGITAL, and PNB demonstrate sideways, upward, and downward trends. The experimental datasets were generated in three steps. Firstly, time order summarization of the data collected from the Bombay Stock Exchange (BSE), Yahoo Finance, and National Stock Exchange (NSE) websites. The dataset size for each stock is 2427 for the trading days from 8th August 2016 to 31st March 2023. Fig. 5 Illustrate the input stock datasets. The total size of the price history data is 14,562 for the six stocks.

In the second step, these datasets were preprocessed by eliminating the entities with null values. Then, the prominent features of the dataset were extracted, such as date and close price, which play a crucial role in predicting the stock value, among the other features: Volume, high, low, adjacent close, and open price. In the final step, each dataset was separated into training and testing with 70 % and 30 %, respectively. Then, testing data consists of 728 days of history records from 4th March 2021 to 31st March 2023. The training data

Table 1. The RMSE and MAE of the proposed and existing models for the stock datasets.

Model/Stock Dataset	Proposed Model						RPGA [21]		Random Forest [19]		Feature Extraction [18]	
	c = 3		c = 6		c = 9							
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
WIPRO	61.29	6.65	11.03	2.77	116.32	9.37	64.88	6.66	51.64	6.27	84.06	8.00
TCS	117.69	9.38	37.22	5.22	113.78	9.18	60.67	6.59	80.24	7.76	80.24	7.73
BHARATARTL	68.20	7.12	17.70	3.56	106.67	8.69	55.93	6.10	115.39	9.05	82.05	7.88
SBIN	63.67	6.77	9.48	2.63	107.72	8.99	10.28	2.77	51.42	6.24	79.17	7.66
NXTDIGITAL	63.36	6.75	10.35	2.80	115.44	9.40	18.56	3.52	50.28	6.15	83.28	8.00
PNB	59.42	6.52	10.01	2.73	122.19	9.65	37.01	5.19	54.76	6.41	80.69	7.78

Table 2. Compound annual growth rate (%) computation results and Annualized volatility.

Stock	Metrics/Models	Proposed Model			RPGA [21]	Random Forest [19]	Feature Extraction [18]
		$c = 3$	$c = 6$	$c = 9$			
WIPRO	CAGR	85	81	117	84	104	103
	AV	0.76	0.82	0.86	0.76	0.74	0.69
TCS	CAGR	156	165	155	159	162	160
	AV	0.66	0.66	0.66	0.66	0.66	0.65
BHARATARTL	CAGR	149	122	127	137	145	129
	AV	0.73	0.76	0.7	0.75	0.71	0.68
SBIN	CAGR	1411	1276	1701	1630	1638	1403
	AV	0.86	0.89	0.92	0.80	0.71	0.71
NXTDIGITAL	CAGR	−98	−98	−97	−98	−98	−97
	AV	1.14	1.23	1.31	1.13	0.97	0.99
PNB	CAGR	681	85	282	124	299	157
	AV	5.42	5.89	8.33	5.00	2.09	1.86

consists of 1699 days of history records from 8th August 2016 to 3rd March 2021. Now, the datasets were ready to input to the existing and proposed models.

4.2. Evaluation metrics

The proposed and existing models were evaluated using the following metrics for comparison analysis. AV and CAGR are the domain−specific and were computed using equations (4) and (5). The annualized volatility represents the fluctuations of investment throughout a particular period. The CAGR (%) means the stock's total returns aggregate.

MAE and RMSE are essential metrics to evaluate regression models computed using Equation (6) and Equation (7). The MAE gives the absolute difference between the predicted and actual values.

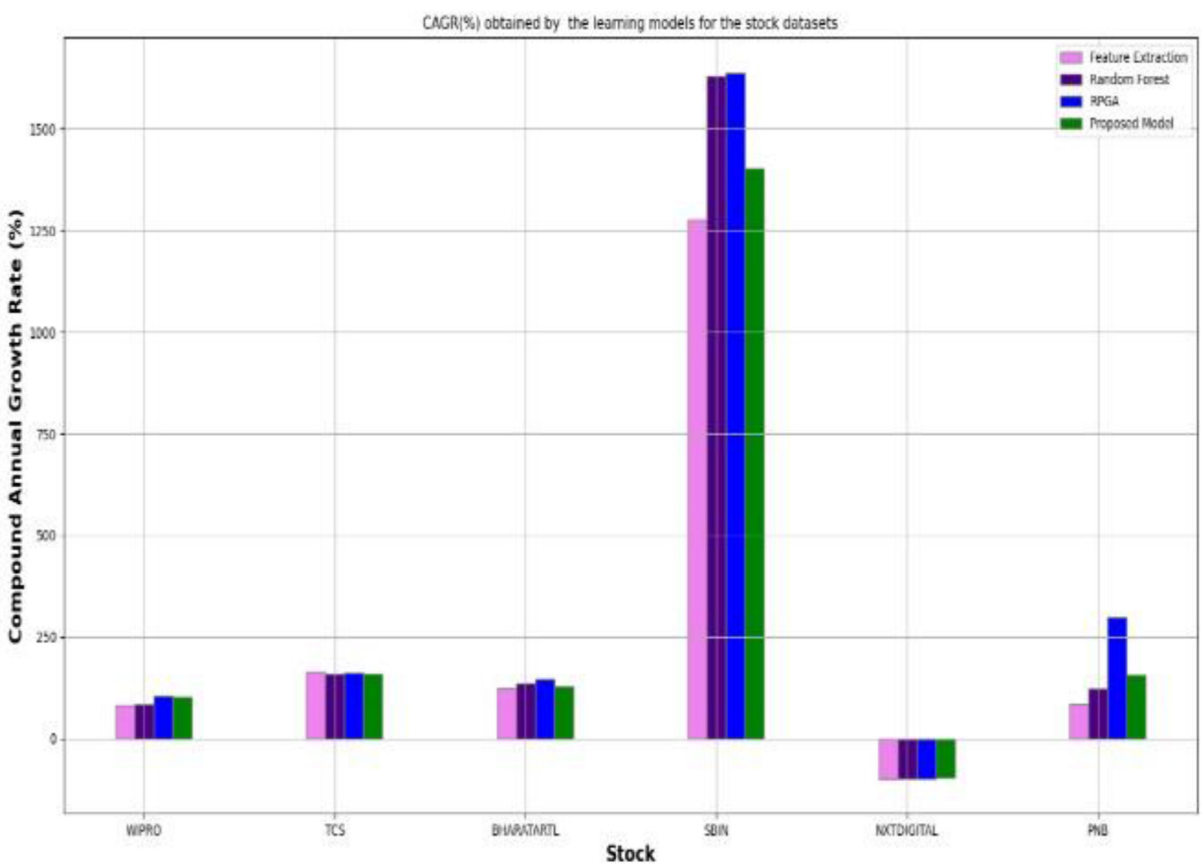

Fig. 7. The compound annual growth rate (%) obtained for the stock datasets.

RMSE gives the square root of the average squared error computed for 250 average trading days per year.

$$CAGR(\%) = \left(\frac{Final\ Price}{Initial\ Price}\right)^{\frac{1}{(Number\ of\ Years)}} - 1 \tag{4}$$

$$Annulized\ Voltility = Standard\ deviation\ of\ Daily\ Returns * Square\ root\ (Average\ trading\ days) \tag{5}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Actual_i - Predicted_i)^2}{N}} \tag{6}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|Actual_i - Predicted_i| \tag{7}$$

4.3. Model parameters

Fuzzy c-means: The Fuzzy c-means performs the soft clustering on the input datasets. The c value is chosen as multiples of 3, i.e., 3, 6, and 9, as the clusters should cover the possible up, down, and sideways trends.

Bi-LSTM: Configuration of Bi-LSTM consists of two hidden, one input and one output layer. The optimizer is Adam, with $batch\ size = 32$.

Reinforcement Learning: The Initial state of the decision is Hold, and the remaining parameters of the Q table are zeros.

Model Evaluation: The computation of CAGR considers the risk factor to be 2 %, and the testing period is 2.7 years. The average trading days are equal to 250 for the calculation of AV.

4.4. The existing and proposed models' evaluation

Existing and proposed systems computed the total returns for the input stock datasets, as shown in Fig. 6. These models were evaluated using 1. General regression metrics RMSE and MAE and 2. domain-specific using metrics CAGR (%) and AV.

The summaries of the results are shown in Tables 1 and 2. Fig. 7 illustrates the CAGR (%) obtained by the stocks. The major observation is that for SBIN stock, the CAGR is high, i.e., 1630 % (Table 2) for the RPGA model when compared with other models. However, the accuracy metrics, the RMSE and MAE, are 10.28 and 2.77, respectively (Table 1), greater than the metrics of the proposed system with $c = 6$. The stock recommendation based on the CAGR without considering general regression metrics leads to overfitting or underfitting conditions. The proposed model with Fuzzy c-means with $c = 6$ achieved higher accuracy with an RMSE of 9.48 and an MAE of 2.63.

5. Conclusion and future scope

This research work develops the integration of Fuzzy c-means with a Multi-agent Deep reinforcement learning model for portfolio profit maximization. The experiment was carried out on six stocks that covered upward, downward, and sideways trends for 2427 days with 14,562 records.

The overfitting of the Bi-LSTM was overcome by Fuzzy-c means. The experiment results showed that both domain–specific and general regression model evaluation metrics must be computed to predict the optimized portfolio profit.

The results showed that MASPPO achieved an RMSE of 9.48 and MAE of 2.63 and outpaced the recent models in the literature. This model is easily applied in real-time scenarios and is simple to implement. However, the present simple MAS conflict resolution with prioritization may not always be efficient. In the future, this work can be enhanced by defining an intelligent MAS conflict resolution model for decision-making with intelligence. The other enhancement can be to adopt explainable Artificial Intelligence (AI) to meet the investor's satisfaction and to improve the model utilization.

Data availability

Datasets related to this article can be found at https://data.mendeley.com/datasets/njp5565zzm.

Acknowledgement

This work didn't receive any external funding.

References

[1] H. Xu, D. Cao, S. Li, A self-regulated generative adversarial network for stock price movement prediction based on the historical price and tweets, Knowl Base Syst 247 (2022) 108712, https://doi.org/10.1016/j.knosys.2022.108712.

[2] S. Banik, N. Sharma, M. Mangla, S.N. Mohanty, S. Shitharth, LSTM based decision support system for swing trading in stock market, Knowl Base Syst 239 (2022) 107994, https://doi.org/10.1016/j.knosys.2021.107994.

[3] D. Wu, X. Wang, S. Wu, Jointly modeling transfer learning of industrial chain information and deep learning for stock prediction, Expert Syst Appl 191 (2022) 116257, https://doi.org/10.1016/j.eswa.2021.116257.

[4] T. Iqball, M.A. Wani, Weighted ensemble model for image classification, Int J Inf Technol 15 (2023) 557–564, https://doi.org/10.1007/s41870-022-01149-8.

[5] K. Radha, M. Bansal, Closed-set automatic speaker identification using multi-scale recurrent networks in non-native children, Int J Inf Technol 15 (2023) 1375–1385, https://doi.org/10.1007/s41870-023-01224-8.

[6] K. Dhrisya, G. Remya, A. Mohan, Fine-grained entity type classification using GRU with self-attention, Int J Inf Technol 12 (2020) 869−878, https://doi.org/10.1007/s41870-020-00499-5.
[7] A. Kanwal, M.F. Lau, S.P.H. Ng, K.Y. Sim, S. Chandrasekaran, BiCuDNNLSTM-1dCNN — a hybrid deep learning-based predictive model for stock price prediction, Expert Syst Appl 202 (2022) 117123, https://doi.org/10.1016/j.eswa.2022.117123.
[8] H.A.E.W. Khalifa, P. Kumar, Solving fully neutrosophic linear programming problem with application to stock portfolio selection, Croat Oper Res Rev 11 (2020) 165−176, https://doi.org/10.17535/CRORR.2020.0014.
[9] M. Schneckenreither, S. Haeussler, J. Peiró, Average reward adjusted deep reinforcement learning for order release planning in manufacturing, Knowl Base Syst 247 (2022) 108765, https://doi.org/10.1016/j.knosys.2022.108765.
[10] P. Kumar, Solution of extended multi-objective portfolio selection problem in uncertain environment using weighted tchebycheff method, Computers 11 (2022), https://doi.org/10.3390/computers11100144.
[11] T. Sadad, A. Munir, T. Saba, A. Hussain, Fuzzy C-means and region growing based classification of tumor from mammograms using hybrid texture feature, J Comput Sci 29 (2018) 34−45, https://doi.org/10.1016/j.jocs.2018.09.015.
[12] Q. Li, Y. Chen, J. Wang, Y. Chen, H. Chen, Web media and stock markets : a survey and future directions from a big data perspective, IEEE Trans Knowl Data Eng 30 (2018) 381−399, https://doi.org/10.1109/TKDE.2017.2763144.
[13] Z. Cui, R. Ke, Z. Pu, Y. Wang, Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values, Transport Res C Emerg Technol 118 (2020) 102674, https://doi.org/10.1016/j.trc.2020.102674.
[14] B. Amar Bensaber, C.G. Pereira Diaz, Y. Lahrouni, Design and modeling an adaptive neuro-fuzzy inference system (ANFIS) for the prediction of a security index in VANET, J Comput Sci 47 (2020), https://doi.org/10.1016/j.jocs.2020.101234.
[15] G. Jeong, H.Y. Kim, Improving financial trading decisions using deep Q-learning: predicting the number of Shares, action Strategies, and transfer learning, Expert Syst Appl 117 (2019) 125−138, https://doi.org/10.1016/j.eswa.2018.09.036.
[16] D. Wu, X. Wang, S. Wu, Construction of stock portfolios based on k-means clustering of continuous trend features, Knowledge-Based Syst 252 (2022) 109358, https://doi.org/10.1016/j.knosys.2022.109358.
[17] A. Corberán-Vallet, E. Vercher, J.V. Segura, J.D. Bermúdez, A new approach to portfolio selection based on forecasting, Expert Syst Appl 215 (2023) 119370, https://doi.org/10.1016/j.eswa.2022.119370.
[18] S. Mittal, C.K. Nagpal, Predicting a reliable stock for mid and long term investment, J King Saud Univ Comput Inf Sci 34 (2022) 8440−8448, https://doi.org/10.1016/j.jksuci.2021.08.022.
[19] M. Pinelis, D. Ruppert, Machine learning portfolio allocation, J Finan Data Sci 8 (2022) 35−54, https://doi.org/10.1016/j.jfds.2021.12.001.
[20] H. Wang, J. Li, Q. Yu, T. Hong, J. Yan, W. Zhao, Integrating recurrent neural networks and reinforcement learning for dynamic service composition, Future Generat Comput Syst 107 (2020) 551−563, https://doi.org/10.1016/j.future.2020.02.030.
[21] G. Mattera, R. Mattera, Shrinkage estimation with reinforcement learning of large variance matrices for portfolio selection, Intell Syst Appl 17 (2023) 200181, https://doi.org/10.1016/j.iswa.2023.200181.
[22] A.M. Aboussalah, C.G. Lee, Continuous control with stacked deep dynamic recurrent reinforcement learning for portfolio optimization, Expert Syst Appl 140 (2020) 112891, https://doi.org/10.1016/j.eswa.2019.112891.
[23] S. Carta, A. Ferreira, A.S. Podda, D. Reforgiato Recupero, A. Sanna, Multi-DQN: an ensemble of Deep Q-learning agents for stock market forecasting, Expert Syst Appl 164 (2021) 113820, https://doi.org/10.1016/j.eswa.2020.113820.
[24] E.F. Morales, J.H. Zaragoza, An introduction to reinforcement learning, Dec Theo Model Appl Artif Intell Conc Solut (2011) 63−80, https://doi.org/10.4018/978-1-60960-165-2.ch004.
[25] N. SSSN Usha Devi, R. Mohan, IASMFT: Intelligent agent simulation model for future trading, Int J Inf Technol (2023), https://doi.org/10.1007/s41870-023-01425-1.