Analysis and implementation of realtime stock prediction using reinforcement frameworks

DrYVS Sai Pragathi¹, M V S Phani Narasimham² and Dr B V Ramana Murthy³
¹CSE Dept, Stanley College of Engineering & Technology for Women, Hyderabad, India
²Senior Architect, Wipro Technologies, Hyderabad, India

E-mail: pragathiyeellani@gmail.com

Abstract. Real time stock prediction is interesting research topic due to the risk involved with volatile scenarios. Modelling of the stocks by reducing the overestimation in ANN model, due to rapid fluctuations in the market guide fund managers risky decisions while building stock portfolio. This paper builds real time framework for stock prediction using deep reinforcement learning to buy, sell or hold the stocks. This paper models the transformed stock tick data and technical indicators using Transformed Deep-Q Learning. Our framework is cost reduced and transaction time optimized to get real time stock prediction using GPU and Memory containers. Stock predictor is architected using GRPC based clean architecture which has the benefits of easy updates, addition of new services with reduced integration costs. Data archive features of the cloud will give benefit of reduced cost of the new stock predictor framework.

Keywords: Stock, KPCA, ANN, ANFIS, Microsoft Azure, AWS.

1. Introduction
Reference stock trading platform key modules are shown in figure 1. It is used by bank securities, stock investors for stock booking and reducing risk of daily and long term trading. Stock viewer module contains reading of real time stock tick data inputs like market-capitalization, daily-high, low, range, ask, bid, last-trade-amount, ebitda, pe-ratio, volume, etc and display the data in user customized grid. Order Transaction module consists of account details (id, balance, daily-profit-loss, gross-profit-Loss), current price per share, transaction type for taking and placing the order. Stock analyser module consists of different charts of stock price, sales volume and financial data of the stock market-cap, net-profit-margin, pe, roe, price-to-free-cashflow, div-yield, price-to-book, etc to display the data in user customized charts. Stock analyser module also consists of heat maps of different industry sectors – technology, financial, healthcare, telecom services and consumer goods. Sentiment Analyzer does cognitive search with a scoring criteria of the stock news and integrates the effect of Facebook and twitter into stock model., AI Filters and AI Modelers will filter the data and will have real time AI models using logistic regression, boosted Bayesian trees, support vector machine (SVM), artificial neural network (ANN), ANFIS and deep learning techniques.
There is very high demand for real time stock predictors especially when investing money by fund managers. Real time stock predictor service models the stock reducing local stock fluctuations, news fluxes and sentimental dynamic markes. The paper consists of the following topics, topics related to the paper are discussed in section 2, cloud reference architecture using GPU and memory intensive containers in section 3, section 4 describes the ANN task model using container orchestration is presented, Deep-Q reinforcement model using real time cloud architecture is summarized in section 5, modelling parameters and evaluation methods are discussed in section 6, results are discussed in section 7, section 8 concludes with enhancement to stock framework and integration of GPU, FPGA containers for real time performance.

![Typical 3-layer architecture](image)

Figure 1. Typical 3-layer architecture

2. Literature Review of Related Work
L. P. Kaelbling in their paper, “Reinforcement Learning: A Survey” have analyzed different reinforcement learning and proposed improvements like shaping, local reinforcement and problem decomposition[1]. K. Arulkumaran et.al in their paper, “Deep Reinforcement Learning. A brief survey” in their paper discussed value based, policy-based agents. Deep Q-network, trust region policy optimization, hybrid actor – critic based neural nets for training the policy models are discussed [2]. Vincent F.L in their paper, “An Introduction to Deep Reinforcement Learning”, discussed the concepts of RL like Value-Based Methods, Policy gradient methods, Model based methods and deep learning for partially observable markov decision process environments [3].

Aparna N. et. al in their paper, “Prediction Models for Indian Stock Market”, have combined prediction models with sentiment analysis. Sentiment of the company using twitter and news combined with open price, close price and volume traded per day are used in the predicted model. Decision boosted tree was found to give better performance than and Logistic Regression [4]. Kunal P. et. al in their paper, “Stock Market Analysis using Supervised Machine Learning” have used SVM to predict the future stock price [5].

Xi Zhang et.al in their paper, “Stock Market Prediction via Multi-Source Multiple Instance Learning”, have used stock data, social media data, web news data and developed multi source multiple instance model (M-MI). SVM is applied on the M-MI model to predict the stock price [6]. K. Hiba
Sadia et al. in their paper, “Stock Market Prediction using Machine Learning Algorithms”, have preprocessed the stock dataset and applied random forest and SVM for predicting of stock price [7].

Osman H et al. in their paper “A machine learning model for stock market prediction”, integrates particle swarm optimization (PSO) algorithm and Least Square(LS)-SVM for stock price prediction using financial technical indicators relative strength index, money flow index, exponential moving average, stochastic oscillator and moving average convergence/divergence[8]. Ekram G et al. in their paper, “Using artificial neural network models in stock market index prediction” have analyzed different ANN models like multi-layer perceptron(MLP), dynamic artificial neural network (DAN2) and hybrid neural networks which use generalized autoregressive conditional heterodasticity (GARCH). They found MLP model outperforms over DAN2 and GARCH in terms of stock market index prediction [9]. Amin H et al. in their paper, “Stock market index prediction using artificial neural network” have applied OSS training method and TANGSIG transfer function in a network with 20-40-20 neurons in hidden layers resulted in an R² value of 0.9408 for validation of NASDAQ dataset[10].

Weiwei Jiang in their paper, “Application of deep learning in stock market prediction: recent progress” have analyzed different linear and machine learning tools that can be used for stock prediction. Different deep learning models and their reproducibility to baseline stock prediction is accomplished in this paper [11]. Joanthan L. T in his paper, “A Bayesian regularized artificial neural network for stock market forecasting” have used Bayesian regularized artificial neural network for predicting the future closing price of individual stocks. The probabilistic nature of the network weights allows the investors to safely expand the network without increasing the risk of overfitting with the use of effective number of parameters [12]. Dharmaraja S et al. in their paper, “Indian stock market prediction using artificial neural networks on tick data” have compared different neural networks learning algorithms - Levenberg-Marquardt, Scaled Conjugate Gradient and Bayesian Regularization for stock market prediction based on tick data and found Bayesian regularization gives best results for stock prediction[13].

Nirbhay N in his paper, “Stock market prediction using Machine Learning and Cloud Computing”, have proposed using cloud computing and machine learning algorithms to perform stock prediction. In this paper SVM, Linear Regression, random Forest, XG Boost, LSTM for deep learning are used to perform stock prediction [14]. Jyh-Shing, Roger Jang in their paper “ANFIS: Adaptive-Network-Based Fuzzy Inference System” presented the architecture and learning procedure underlying ANFIS, a fuzzy inference system implemented in the framework of adaptive networks [14]. ANFIS modeling is used to solve the non-linear problems. An adaptive neuro-fuzzy inference system (ANFIS) integrates both neural networks and fuzzy logic principles in a single framework [15]. Ghofrane Rehaiem, Hamza Gharsellouai, Samir Ben Ahmed in their paper "A Neural Networks Based Approach for the Real-Time Scheduling of Reconfigurable Embedded Systems with Minimization of Power Consumption" have used ANN based back propagation technique to model the real time scheduling of tasks in embedded systems to achieve minimum cost [16].

ANN with proper number of input, output and hidden neurons is able to identify and classify complex regions . S.Agatanovic-kustrin, R. Beresford, “Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research” has applied ANN to find optimal dosage of the drug and analyzing chromatographic data[17].

M.V.S Phani Narasimham et al. in their paper, “Development of realistic models of oil well by modeling porosity using modified ANFIS technique” have used ANN based neuron porosity model will give reliable static reservoir models for oil well simulation frameworks [18]. Kishore Kumar et al. in their paper, “Comparative Study on Internet of Things: Enablers and Constraints”, have discussed applying cloud technologies in IOT space [19].

Alberto Nunez et al. in their paper “iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator” propose cloud simulator tomodel cloud based large simulations by varying number and type of virtual machines [20].

Richard S. Sutton et al in their paper, “Policy gradient methods for reinforcement learning with functionapproximation”, have proposed actor critic method for function approximation [21]. Scott F et.
al in their paper, “Addressing Function Approximation error in Actor-Critic Methods” have used Double-Q learning to reduce function approximation errors and to find optimal policy agents [22]. Rajkumar B et.al in their paper, “Cost-Efficient Orchestration of Containers in Clouds: A Vision, Architectural Elements, and Future Directions” have designed methodology for efficient orchestration of containers in cloud environments[23]. Isam Mashhour et. al in their paper, “Container Orchestration Engines: A Thorough Functional and performance comparison” have studied Kubernetes, Apache Mesos, Docker Swarm and concluded Kubernetes is better choice for complex deployments, other engines give better performance for simple deployments [24].

Hritwik B et.al in their paper, “A Survey on Efficient Container Orchestration Tools and Techniques in Cloud Environment” have reviewed analysis of container orchestration methodologies using Genetic algorithm, ant colony optimizations, Markov prediction models [25].

Danilo De et. al in their paper, “Introduction to GPU Computing and CUDA Programming: A Case Study on FDTD” has shown 42.5% improvement in peak speed for Finite difference time domain applications [26]. Jonathan T et. al in their paper, “An introduction to the OpenCL Programming Model”, has shown using opencl on GPU for large matrix computations has significant performance benefit compared to CPU [27]. Kamra K et.al in their paper, “A Performance comparison of CUDA and OpenCL” have shown CUDA performs well than OpenCL with trade-off on portability [28].

3. CPU, GPU, Memory Intensive PODS
Existing stock predictor application is architected to use GRPC based micro services are containerized as docker containers. AWS Fargate and Azure Kubenetes with custom scheduling on spot instances is used for testing the reference architecture.

The optimal number of containers pods to achieve real time response will be determined based on optimized energy and end to end delay schemes. This approach has added advantage as it replaces the heavy weight VMs with cost and time optimized docker pods giving more granularity to cloud scheduling.

Stock Viewer and Chart Visualizer pods are implemented using Angular JS and communicate with GRPC APIs. AI Filters and AI models are implemented using C# and Flask Restful Python frameworks. Intermediate stock data required for AI analysis is saved in mongo db for quick and easy access. GPU containers use OpenCl and CUDA libraries for the iterations and Deep-Q episodes.

3.1. ANN Based Real Time Cloud Architecture
ANN Supervised learning model with various algorithms backpropagation, hybrid algorithms are explored to develop optimal ANN based cloud reference model. ANN Perceptron network with input layer, hidden layer and output layer classifies input stock predictor tasks.

Input batch of simulator tasks of a real time scenario are classified based on the task features obtained from the training batches. Stock predictor characteristics in json file are fed to the ANN modeler.

Training batch measures the following VM container characteristics CPU Speed, GPU, Memory Intensive and Generic. The ANN classifier will identify the most suitable containers for execution

Class 1 container - are CPU intensive with threads,

Class 2 containers - are GPU intensive using CUDA and Open Cl.

Class 3 containers - are memory intensive.

Class 4 Generic containers.

Ant colony optimizer will find the optimal schedule for execution of the tasks on the containers that will have the minimum timespan and cost for execution. Reducing the cost reduces the power usage and this will be determined by the energy consumptions of the Green cloud simulator.
4. ANN Task Model

Input task characteristic are input to 4 input Perceptron layers followed by two output neurons. The perceptron network weights are tuned using back propagation on the cost from the training set.

\[ t = \{ t_1, t_2, t_3, t_4 \} \] is the expected target value set.

\[ z_j^L = w_{j0}^L o_1^L + w_{j1}^L o_2^L + w_{j2}^L o_3^L + w_{j3}^L o_4^L + b_j^L \] (2)

\[ o_j^L = f(z_j^L) \] (3)

Equation (1) gives the cost of the network model using mean square error of output and expected target.

\[ C = \sum_{j=0}^{K=n} (o_j^L - t_j^L)^2 \] (1)

(2) Calculates the input to the output neuron of the final layer. Eq.3 is the activation function output from input z.

\[ \frac{\partial c_0}{\partial w_{jk}^L} = \frac{\partial z_j^L}{\partial w_{jk}^L} \times \frac{\partial o_j^L}{\partial z_j^L} \times \frac{\partial c_0}{\partial o_j^L} \] (4)

Figure 2. ContainerImages in Stock Predictor Framework

Equation (1) gives the cost of the network model using mean square error of output and expected target.
4.1. ACO Mathematical Model

In ACO the pheromone component $\eta$ is computed using $1 \div (ETC_{ij} + Pod_{j} + Cost_{j})$, where $ETC_{ij}$ is the expected time to compute task on the Pod, $Pod_{j}$ running in the VM and cost is cost per hour charged for the containers.

The pheromone of ant $k$ mapping task $i$ to container $j$ is given by

$$\tau_{i,j} = (1 - p) \tau_{i,j} + \sum_{k=1}^{m} \Delta \tau_{i,j}^{k}$$

(5)

probability of mapping task $i$ to container $j$ is given by

$$P_{i,j}^{ant,k} = \frac{(\tau_{i,j})^{\alpha} (\eta_{i,j})^{\beta}}{\sum((\tau_{i,j})^{\alpha} (\eta_{i,j})^{\beta})}$$

(6)

$\beta$ represent the influence factor of heuristic value. Cumulative probability of task $i$ mapping to the container is calculated. $\{ P_{i}, P_{i+1}, P_{i+2}+3, \ldots \}$.

5. Deep-Q Reinforcement Stock Model

Figure 3. 4-input and 4-hidden neurons perceptron network.

Figure 4. Architecture of Deep Q learning for stock market.
5.1. Actor-Critic Model

Deep-Q reinforcement learning consists of training an agent that interacts with environment. At each time step agent will execute action \(a_i\) and will have different states \(s_i\) and reward. Best agent will optimize the reward for an episode by selecting the optimal action.

Actor critic model of [20] is used in the policy agent to reduce the variance of the policy gradient. The actor policy is updated by performing gradient ascent of Q value of the critic model.

![Figure 5. Actor Policy of the Deep Q learning for stock market.](image)

6. Modelling Parameters

Stock tick data parameters open, high, low, close and volume are transformed and used as observations for the policy. Estimated moving average (EMA10), Relative strength index (RSI), EMA20, Accumulation Distribution indicator (ADO) are added into the actor-critic deep-q model of section 5. model to determine the stock closing price and stock current price in real time. Accumulation Distribution indicator (ADO) catches difference between stock price and volume flow. If price is
rising but ADO is falling indicates that price decline is going to occur. Real time optimized reinforcement model is developed for each stock in real time to determine the accurate prediction for the day. Estimated moving average helps the model to detect the peaks and dips of the markets. Relative strength index (RSI) helps the model to capture the pattern of overbought when it is over 70 and oversold when it is below 30.

![Stack Data Transformation for Deep Q learning for stock market](image)

**Figure 7.** Stack Data Transformation for Deep Q learning for stock market

Behaviour of the Reinforcement Model on KPCA transformed using Gaussian kernel parameters is also analysed for prediction.

### 7. Results

Training batch of resource tasks is used to measure the makespan of the tasks on CPU intensive, GPU Intensive, Memory Intensive and Generic container VMs.

Relative time span characteristic of each task is measured using the below formulas.

- \( T_{\text{CPU}} \) = Time Span of the task on CPU intensive VM.
- \( T_{\text{GPU}} \) = Time Span of the task on GPU intensive VM.
- \( T_{\text{G}} \) = Time Span of the task on Generic VM.
- \( T_{\text{MI}} \) = Time Span of the task on Memory Intensive VM.

Relative Time Span of CPU

\[
R_{T_{\text{CPU}}} = \frac{100 \times T_{\text{CPU}}}{\text{Max}(T_{\text{CPU}}, T_{\text{GPU}}, T_{\text{G}}, T_{\text{MI}})}
\]

Relative time span parameters of the tasks are used as inputs to the ANN models of Perceptron networks trained using back propagation.
Figure 8. 4-Input, 4-Output ANN Perceptron Network.

Input parameters of the task characteristic needed for the real time stock predictor service are given as input json file. Input stock predictor training data is used to model the ANN classifier at desired rate of $1e^{-4}$ and learning rate of 0.1.

Figure 9. MSE error of output in the final iteration of ANN.

Docker container images of stock predictor architecture described in section 3 are saved in secure container private registry. VMs orchestrated using the task model in section 4. Table 1 gives the results of the actor-critic deep-Q model used for taking the decision of buy, sell, hold of the stock. Real time stocks tick data for 10 years is used to train the model and tested on 1 years of taking right decision. Table 2 gives the results of the transformed actor critic deep-Q model for 3 configurations.

| Model                   | Profit % for Stock1 | Profit % for Stock2 | Profit % for Stock3 |
|-------------------------|---------------------|---------------------|---------------------|
| Actor-Critic Deep-Q     | 35.33               | 31.05               | 41.03               |
| Transformed Actor Critic Deep-Q | 45.66               | 45.02               | 51.03               |

Table 1. Profit percentage for 1 years for three different stocks with Actor-Critic Deep-Q model and Transformed Actor-Critic Deep-Q model.
Deep-Q Model Configuration 1 Configuration 2 Configuration 3
Number of Nodes 300 350 400
Transformed Actor Critic Deep-Q 40.23 45.66 45.61

Table 2. Profit percentage for 1 years for varying nodes for Stock 1 for Transformed Actor-Critic Deep-Q model.

8. Conclusion
Stock real time application framework is proposed using reinforcement model and integrated with actor critic deep-Q model which gives maximum profits to daily trader. Integrating sentiment analysis into the reinforcement model using cognitive services from Azure or AWS clouds will give more accurate results for the fund managers and security investors reducing the risk of bad investments. From the results transformed, Deep-Q learning with real time reference cloud architecture has the benefit of reduced cost and reduced delay. Further real time performance needs to be explored using FPGA containers while optimizing the Deep-Q episodes and stock data transformations. In the next research paper sentiment analysis will be integrated into the framework giving more profits to daily traders.

Acknowledgements
Thankful to Stanley College of Engineering & Technology for women, Principal Dr Satya Prasad Lanka and College management for encouraging to pursue new research areas.
Thankful to Wipro management, Ramakrishna Kashikar, Ganeshan Kailasam in encouraging and helping with relevant environment for working on this article.

References
[1] L P Kaelbling, ML Littman, AW Moore, Reinforcement learning: A Survey, Journal of Artificial Intelligence Research 4 (1996) 237-285.
[2] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage and Anil Anthony Bharath, “Deep Reinforcement Learning, A brief survey”, IEEE Signal Processing Magazine, 13 Nov 2017, DOI 10.1109/MSP.2017.2743240.
[3] Vincent F L, Peter H, Riashat I, Marc G B and Joelle P, “An introduction to Deep Reinforcement Learning”, Foundation and Trends in Machine Learning: Vol. 11, No. 3-4, pp 219-354. DOI: 10.1561/2200000071.
[4] Aparna N, M.M. Manohara Pai and Radhika M. Pai, “Prediction Models of Indian Stock Market”, 216 international multi conference on information processing – 2016, Procedia Computer Science 89 (2016) 441-449.
[5] Kunal P, Neha A, “Stock Market Analysis using Supervise Machine Learning”, 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, India 14th-16th Feb 2019.
[6] Xi Zhang, Si Qu, Jieyun H, Binxing F and Philip Y, “Stock Market Prediction via Multi-Source Multiple Instance Learning”, IEEE Access, Aug 27, Oct 8 2018, DOI 10.1109/ACCESS.2018.2869735.
[7] K.Hiba Sadia, Adita S, Adaarrsh P, Sarmistha P, Saurav S, “Stock Market Prediction Using Machine Learning Algorithms", IEEE, ISSN: 2249-8958, Vol-8, Issue -4, April 2019.
[8] Osman H, Omar S. S, and Mustafa A S, “A Machine learning model for stock market prediction”, International Journal of Computer Science and Telecommunications, Vol-4, Issue -12, Dec 2013, pp 17-23.
[9] Erkam G, Gulgan K and Tugrul U.D, “Using artificial neural network models in stock market index prediction”, Expert systems with Applications, DOI 10.1016/j.eswa.2011.02.068.
[10] Amin H. M, Moein H M, Morteza E, “Stock market index prediction using artificial neural network”, Journal of Economica, Finance and Administrative Science, doi 10.1016/j.jefas.2016.07.002.

[11] Weiwei Jiang, “Application of deep learning in stock market prediction: recent progress”, Elsevier Journal, Mar 5 2020.

[12] Jonathan L. T, “A Bayesian regularized artificial neural network for stock market forecasting”, Elsevier, Expert systems with Applications, doi 10.1016/j.eswa.2013.04.013.

[13] Dharmaraj S, Venket K and Abbishke M, “ Indian stock market prediction using artificial neural networks on tick data” Financial Evolution, doi 10.1186/s40854-019-0131-7.

[14] Nirbay N, “Stock Market Prediction using Machine Learning and Cloud Computing”, International Journal of Engineering And Computer Science, Vol-8 Issue 9 Sept 2019, pp No. 24847-24850.

[15] Yih-Shing Roger Jang, “ANFIS:Adaptive-Network-Based Fuzzy Inference System”, IEEE Transactions on Systems, Man and Cybernetics, June 1993.

[16] Ghofrane Rehaiem, Hanza Gharsellaoui, Samir Ben Ahmed, "A Neural Networks Based Approach for the Real-Time Scheduling of Reconfigurable Embedded Systems with Minimization of Power Consumption.", ICIS Conference, At Okayama, Japan, June 2016, 10.1109/ICIS.2016.7550777.

[17] S Agantonioc-kuskin, R Beresford, “Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research”, Journal of Pharmaceutical and Biomedical Analysis”, 22(2000) 717-727.

[18] MVS Phani Narasimham, Dr YYV Sai Pragathi, “Development of realistic models of oil well by modeling porosity using modified ANFIS technique"International Journal on Computer Science and Engineering, Vol.11, No.07, July 2019.

[19] Kishor Kumar Reddy C, Anisha P R, Shastry R, Ramana Murthy B V, “Comparative Study on Internet of Things: Enablers and Constraints”, Advances in Intelligent Systems and Computing, 2021.

[20] Alberto Nunez,Jose L.Vázquez-Poleti,Agustin C Caminero,Gabriel G Castañé,Jesus Carretero,Ignacio M Llorente, “CanCloud: A Flexible and Scalable Cloud Infrastructure Simulator”, J Grid Computing (2012) 10:185–209.

[21] Richard S Sutton, David McAllester, Satinder Singh, Yishay Mansour, “ Policy gradient methods for reinforcement learning with function approximation”, Proceedings of the 12th internation conference on Neural Information Processing, November 1999, PP 1057-1063.

[22] Scott F, Herke V Hoof, David M, “Addressing Function Approximation Error in Actor-Critic Methods”, Proceedings of the 35th internaton conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[23] Rajkumar B, Maria A. Rodriguez, Adel Nadjaran Toosi, Jeeman Park, “Cost-Efficient Orchestration of Containers in Clouds: A Vision, Architecture Elements, and Future Directions”, Proceedings of the Mathematics, Informatics, Science, and Education International Conference (MISEIC 2018), Surabaya, Indonesia, July 21, 2018, DOI 10.1088/1742-6596/1108/1/012001

[24] Isam Mashhour Al Jawarneh, Paolo B, Fillipi B, Lucas F, Giuseppe M, Rebecca M, Amedeo P, “Container Orchestration Engines: A Thorough Functional and Performance Comparison”, ICC 2019-2019, IEEE International Conference on Communications, DOI 10.1109/ICC.2019.8762053

[25] Hritwik Bairagi, Uday Chourasiya, Sanjay Silakari, Priyanka Dixit, Smita Sharma, “A Survey On Efficient Container Orchestration Tools and Techniques In Cloud Environment”, IJSTR, Vol. 9, Issue 1, Jan 2020. ISSN 2277-8616.

[26] Danilo De Donno, Alessandra Esposito, Luciano Tarricone and Luca Catarinucci, “Introduction to GPU Computing and CUDA Programming: A Case Study of FDTD”, IEEE Antennas and Propagation Magazine, Vol. 52, No. 3, June 2010.

[27] Jonathan T, Kristofer S, “An Introduction to the OpenCL Programming Model”, Pearson Education, 2012/1/25, Vol 49, Page 31.

[28] Kamran K, Neil G. D, Firas H, “A Performance Comparison of CUDA and OpenCL”, Computing Research Repository, 2010/05/14, arXiv:1005.2581.