OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

REPRESENTATION OF FINANCIAL SIGNAL AND TRADING USING DEEP NEURAL NETWORK

Ms. Gayatri Panchwagh[1], Dr. Deepak Gupta[2]

Department of Computer Engineering Siddhant College of Engineering Sadumbre, Savitribai Phule, Pune University, Pune[1]

Abstract: The stock markets all over the world have many uncertainties which contain a large amount of data with jumps, noises, and movements in leading to the highly non-stationary time series. Here, introduce real-time financial signal representation and trading techniques as RDNN (Recurrent Deep Neural Network) for the environment sensing and recurrent decision making for the online financial assert trading. In which the RDNN have two parts, first one is DNN (Deep neural network) function of this DNN is feature learning and the second one is RNN (Recurrent Neural Networks) to predict rapidly changing market condition. This proposed model based on two biological related learning concepts such as Reinforcement Learning (RL) and Deep Leaning and DL (Deep Learning) is used to represent financial signals and self-taught reinforcement trading. Here, RL is used to interact with deep representations and makes trading decisions to accumulate the ultimate rewards in an unknown environment. It improves the robustness of market summarization using fuzzy learning concepts it reduces the uncertainty of input data. Fuzzy MLP will be used to provide the prediction of the stock market.

Keywords: Deep Neural Network, Direct reinforcement Learning, Fuzzy logic, Multilayer Perceptron Model.

I INTRODUCTION

Portfolio management is the process of making decisions about the constant redistribution of the Fund amount in a number of different financial investment products in order to maximize profits while limiting risk. Traditional methods of portfolio management can be divided into four categories: "follow the winner", "follow the loser", "pattern matching" and "meta-learning" [1]. The first two categories are based on previously built financial models, although some machine learning techniques for parameter determination may also help. The effectiveness of these methods depends on the reliability of the models in different markets. The Pattern-Matching algorithms predict the next market distribution based on a sample of historical data and explicitly optimize the portfolio based on sample distribution. The last class, the method of "meta-learning", combines several strategies of other categories to achieve more consistent work [2].

There are deep machine learning approaches to trading financial markets. However, many of them try to predict price movements or trends [3]. A history of prices the neural network can output the predicted vector of asset prices from all assets as input data for the next period. Then the sales agent can act on this forecast. This idea easy to implement because it is supervised learning, or more specifically regression problem. However, the performance of these algorithms is based on price forecasting, strongly depends on the degree of accuracy of the forecast, but it turns out that the future market prices are difficult to predict. In addition, price forecasts are not market actions, converting them into actions requires an additional layer of logic. If this layer is a manual code, then the whole approach is not fully machine learning, and thus not very open and adapted. For example, for a network-based prediction, it is difficult to consider the transaction value as a risk factor.

Previous successful attempts in Model-free and complete machine learning schemes for algorithmic trading problems without predicting future prices include this recent deep use by Moody and Sachel, denpster and Rayman, caming, and RL Deng et al. (2017). These RL algorithms output a separate trading signal to the asset. Limited to single asset trade, they do not apply to the general problem of portfolio management; where distributors manage multiple assets.

In this paper study about the Literature Review done, in section II, the Proposed Approach Modules Description, Mathematical Modeling, Algorithm and Experimental setup in section III .and at finally provide a Conclusion in section IV.

II LITERATURE REVIEW

Here, present the literature review of existing techniques:

Deng Yue, Bao Feng, and Ren Zhiquan [1] presents the DL (Deep Learning) to financial signal processing and a typical DRL (Deep Recurrent Learning) framework for online trading. The contribution of the system is twofold. First, it is a trading system without technical indicators that greatly frees a person to choose features from a large number of candidates. This advantage is due to the automatic feature learning mechanism of DL. The power of the DDR system has been verified, but there are some promising future directions. First, all the methods proposed in this paper to handle one share of assets. In some large hedge funds, the trading system is required to be able to constantly manage a number of assets at the same time.

D. Wierstra, J. Hunt and D. Silver[2], used Deep Q-learning for the actor-critique model-free algorithm based on a deterministic policy gradient that can operate on a continuous action space. The same learning algorithm, network architecture hyperparameters, this algorithm has a powerful solution to more than 20 simulated physical tasks, including classic problems such as cart pole swing-up, manual dexterity operation, carrying legs, driving a car. In addition, many tasks demonstrate that the algorithm can learn the policy "end-to-end" directly from the raw pixel input.

David Lu [3], they present the possible concepts for realizing trading and present robot trading concept. In order to achieve a similar level of performance and generality, like human traders, agents are used to maturing learning model that leads to long-term rewards at the human level. The robustness and feasibility of the system implemented in the iterative structure of LSTM (Long Term Short Term Memory) with a reinforcement learning or evolution strategy that acts as an agent is comparable to that of GBPUSD.

James Cumming [4], develops system reinforcement learning of an algorithmic trading problem, in this, they define terms of the classical structure of the reinforcement learning problem. The aim of the reinforcement learning methods to optimize the agent's performance in an unknown environment are actively developed and regularly implemented and advanced solutions are improved.

Dongbin Zhao and Yuanheng Zhu [5], proposed PAC (Probably Approximately Correct) algorithm to solve online RL (Reinforcement Learning) of continuous state systems for this used deterministic transition functions. In this, they firstly used PAC for continuous deterministic systems without relying on any system dynamics. For analyzing high utilization of online observed samples they combine the efficient exploration principle with the state aggregation techniques. To save sample values they used grid portions it saves continues state space into the different cells. It is helpful to compare this algorithm with some former PAC algorithms.

Volodymyr Mnih, Daan Wierstra and Shane Legg [6], used the latest advances in deep neural network training to develop an artificial agent called deep Q-Network, which can learn successful policies directly from the multidimensional sensor inputs for this used comprehensive reinforcement learning techniques. Performance of system tested on classic Atari 2600 games. The deep Q-network techniques, for receiving only pixels and the game score as input, was able to surpass the performance of all previous algorithms and reach a level comparable to that of the professional human games tester in a set of 49 games.

Yue Deng, Risheng Liu, and Sanqing Hu [7] present a nonconvex framework for the learning of essential low-rank structure from corrupted data. In traditional approaches they directly utilize convex norms for measuring the sparseness this proposed method introduces more reasonable nonconvex measurements to enhance the sparsity in both the intrinsic low-rank structure and the sparse corruptions. They introduce techniques on how to combine the widely used $l_p norm (0 < p < 1)$ and log-sum term into the framework of low-rank structure learning and this is solved by MM (Majorization–Minimization) algorithm.

Alex Graves, Geoffrey Hinton and Abdel-rahman [8], Mohamed present RNNs (Recurrent Neural Network) by neural networks for progression-relapse. By end-to-end training methods such as connectionist time classification, RNNs can be trained for sequence tagging problems where input/output alignments are unknown. The combination of these methods with the long short-term memory RNN architecture provides state-of-the-art results in cursive handwriting recognition and has proved particularly fruitful. But the performance of RNNs is Speech recognition is a deep feedforward network that returns better results. The flexible use of long-distance contexts that empower RNNs and the ability to combine multiple levels of expression has proven to be highly effective in deep networks.

Yue Deng, Zengke Zhang, and Qionghai Dai [9] present a framework for the fusing noisy point groups from multi-view images of the same object. They solve the classical visual problem using a signal processing technique called Matrix complete. In this framework, they build the initial imperfect matrix from the observation point group of all cameras, which are invisible to the point where the other cameras are shown as unknown works. The observation points corresponding to the same object point are placed on the same line. When done properly, all columns describe the same object, so the restored matrix is Rank. Therefore, an intuitive approach to minimizing the Matrix due to the fact that the rank of the subject and the consistency of the observation by which the work was submitted.

George Dahl, Li Deng, and Dong Yu [10] present the Context-Dependent (CD) techniques for LVSR (Large Vocabulary Speech Recognition) which utilizes recent advances using deep belief networks for telephone recognition. They also pre-trained DNN-HMM (Deep Neural Network Hidden Markov Model) hybrid architecture that trains the DNN to produce a distribution over senones as its output. Experimental results show that business search dataset demonstrates that CD-DNN-HMMs it significantly outperforms the conventional context-dependent and accuracy of this improvement of 5.8% and 9.2%.

Deng Yue, Kong Youyong, Dai Qionghai, and Bao Feng [11], present the real-time high-frequency financial signal representation and an optimal trading (Scot) system with sparse coding for trading. The SCOT (sparse coding-inspired optimal trading) simultaneously learns trading strategies in dictionaries, sparse features, and co-optimization to obtain optimal feature representations for specific trading objectives. The learning process is modeled as a bi-level optimization and solved by an online gradient descent method of fast convergence. In this dynamic context, this approach will be tested in real financial markets to trade index futures in the Shanghai exchange center.

Lee Honglak, Grosse Roger, and Ranganath Rajesh [12] present a convolution-deep belief network, which is a scalable generation model for learning hierarchical representations from unlabeled images, and show that the model works well in various visual recognition tasks. This approach is a promising and scalable learning algorithm of hierarchical representation from high-dimensional complex system data.

Deng Yue, Dai Qionghai, Qian Yanjun and Li Yipeng [13], propose a code word assignment method to generate statistical histograms for image classification. The technical contribution of this paper is mainly concerned with the information theoretic manifold embedding model for codeword allocation. Presenting use of a graph structure to clarify the nonlinearity between large-scale image features and to regularize the discrimination of data label pairs in the embedded space by maximizing mutual information. Further, since most of the algorithm still enters the typical similarity-based allocation framework, other coding methods, for example, to improve the classification accuracy in sparse coding.

Deng Yue, Dai Qionghai, and Zhang Zengke [14] propose a spectral-graph-based algorithm for face image repairing, which can improve the recognition performance on occluded faces. The face completion algorithm proposed in this paper includes three main procedures: 1) sparse representation for partially occluded face classification; 2) image-based data mining, and 3) graph Laplace (GL) for face image completion. The novel part of the proposed framework is GL, as named from graphical models and the Laplace equation, and can achieve a high-quality repairing of damaged or occluded faces. The relationship between the GL and the traditional Poisson equation is proven.

III PROPOSED APPROACH

A. Problem Statement

Propose recurrent deep neural learning techniques for the real-time financial signal representation and trading and this technique based on the two biological related learning concepts Deep Learning (DL) and Reinforcement Learning (RL) it automatically senses dynamic marketing condition and provides a prediction for the market conditions.

B. Proposed System Overview

In this present techniques RDNN (Recurrent Deep Neural Network) Structure for the environment sensing and recurrent decision making for the online financial assert trading. In which the RDNA using a two parts DNN (Deep neural network) it is used for the feature learning and the second one is RNN (Recurrent Neural Networks) for the RL

(Reinforce Learning). Improve the robustness of market summarization using fuzzy learning concepts it reduces the uncertainty of input data.

The DDR trading system is used data of the real financial market for future contracts trading. In detail, it accumulates the historic data of Yahoo Finance NSE, BSE, and Bajaj data. This real market data will be directly used for performance verifications. This system analyzes 30 days of data and system performance compared with existing system. The comparisons show that the DDR system and its fuzzy extension are much robust to different market conditions and could make reliable profits on various future markets.

i. Read Dataset:

In this module, it read financial data. It used real time financial historical data such as Yahoo Finance NSE, BSE, and Bajaj data.

ii. Fuzzy Extensions:

In this module for data, uncertainty problem is considered. Financial sequences contain a high amount of unpredictable uncertainty due to the random gambling behind trading. Besides, a number of other factors, e.g., global economic atmosphere and some company rumors may also affect the direction of the financial signal in real time. Therefore, reducing the uncertainties in raw data is an important approach to increase the robustness for financial signal mining.

Membership functions of fuzzy sets are never unique. Different individuals might define various F A ' S for the same fuzzy set. For example, the membership function of a fuzzy TALL defined by an American for Americans would probably be different from that defined by an Oriental for Orientals. Measures of fuzziness estimate the average ambiguity in fuzzy sets in some well-defined sense. Begin by considering properties that seem plausible for such a measure.

The fuzziness of a crisp set using any measure should be zero, as there is no ambiguity about whether an element belongs to the set or not. If a set is maximally ambiguous(p ~ (z =) 0.5 V x), then its fuzziness should be maximum. HzE(A, P) is a measure associated with fuzzy set A and when a membership value approaches either 0 or 1, the ambiguity about belongingness of the argument in the fuzzy set decreases.

Extensions to Membership Functions on Real Intervals of the multiplicative and additive classes are easily extended to fuzzy sets on any real interval. Let A be a fuzzy subset of X c R. Then define a measure of fuzziness (under the multiplicative class) for A as:

$$H_{*c}(A) = K \int_x g(\mu A(x))dx$$

Where K, is a constant and g is a continuous function whose form is defined in H, reduces to H, at whenever X is discrete.

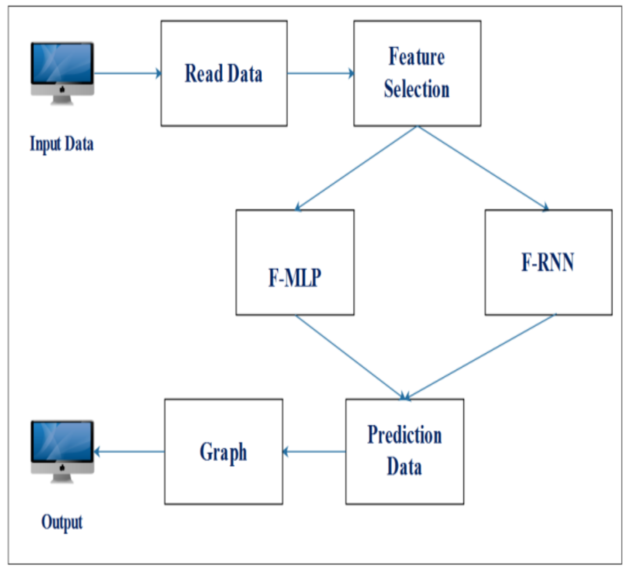

Figure 1. Proposed System Architecture

For the reducing uncertainty of data mostly used the artificial intelligent techniques fuzzy learning. In which is used the input as linguist values. In it compare a real word data with the number of fuzzy-rough sets and then deriving corresponding fuzzy membership degrees. Fuzzy-rough sets are naturally defining according to basic movements of the stock price. Fuzzy sets are defined by increasing decreasing and no trend groups. The parameters in the fuzzy membership function are predefined.

iii. Direct Reinforcement Learning(DRL):

DRL (Direct Reinforcement Learning) is the review of the Moodys DRL frameworks and typical DRL is the one-layer RNN. In this consider the P1, P2… Pn is the sequences released from exchange center. The return at time point t is determined by the Z = Pt - Pt-1 his value is based on the current market condition real-time trading decision policy.

The profit Rt made by the trading model is obtained by:

$R_t = \delta_t - Z_t - C|\delta_t - \delta_{t-1}|$

iv. Recurrent Deep Neural Network (RDNN):

In this module, using practical learning strategy to train the DNN using two steps of system initialization and fine-tuning. For the learning of the DNN use three learning parts, First one is fuzzy representation part is it the easy process of initialization. In this only parameter are specified in the fuzzy center. The parameter k is fixed as 3 because each input node is connected with three membership functions. The second one is RNN contains an input layer, a dense layer (128 hidden neurons), a recurrent layer, and a soft-max layer for classification. In the third part, it compares the DDR framework with other prediction-based DNNs.

v. Fuzzy Multi-Level Perceptron (FMLP):

A three-layered fuzzy MLP is considered. The input is modeled in terms of the 3-dimensional linguistic feature space while the output consists of class membership values. The feature space gives us the condition attributes and the output classes the decision attributes, so as to result in a decision table. This table, however, may be transformed, keeping the complexity of the network to be constructed in mind. Rules are then generated from the (transformed) table by computing relative educts. The dependency factors of these rules are encoded as the initial connection weights of the fuzzy MLP. The network is next trained to refine its weight values.

C. Algorithm

- Algorithm 1: RNN Algorithm
  Step1-Initialise input dataset
  Step2-Remove entry which contains missing values
  Step3- Repeat step-2 until reach end of file
  Step4-Create new .csv file removes missing values from the dataset
  Step5-Generate training and testing file from.csv
  Step6-Train RNN on Training file
  Step7-Prediction of next Working-day
- Algorithm 2: Fuzzy-Multi-Layer perceptron Model:

In Fuzzy MLP Algorithm, The concept of fuzziness is inherited from FRNN algorithm. In this algorithm, it uses fuzzy logic to handle uncertainties. Clustering stock data into 3 clusters like Losser, Gainer, and Neutral. So it makes easy for MLP to handle data to recall error back propagation (EBP) training.

If a submitted pattern provides an output far from the desired value, the weights and thresholds are adjusted s. t. the current mean square classification error is reduced. The training is continued/repeated for all patterns until the training set provides an acceptable overall error. Usually, the mapping error is computed over the full training set. EBP training is working in two stages:

1. The trained network operates feed-forward to obtain an output of the network
2. The weight adjustment propagates backward from the output layer through a hidden layer toward the input layer.
3. Error Back-Propagation Training Algorithm
   i. Initialization of k means algorithm.
   ii. Use Fuzzy membership function to calculate
   iii. Values of each record in a dataset.
   iv. Given P training pairs $\{Z_1, \quad D1, Z_2, D_2, \dots, Z_p, D_p\}$ where Zi is (l * 1), Di is (K*1), i=1,……….,P
      a. The lth component of each Zi is of the value -1 since the input vector is augmented.
   v. Size J-1 of the hidden layer having output y is selected.
      a. Jth component of y is -1 since hidden layer have also been augmented.
      b. Y is (J * 1) and o is (k * 1)
   vi. In the following, q is a training step and p is stepped counter within the training cycle.
      a. Choose $\eta > 0, E_{max} > 0$
      b. Initialized weight at small random values, W is (k * j), V is (j* l),
      c. Initialize counters and error: $q \leftarrow 1, p \leftarrow 1, E \leftarrow 0$
      d. Training cycle begins here:
         Set $\leftarrow Z_p, D \rightarrow D_p$;
         $y_i \leftarrow f(V_j^t Z), j = 1, \dots \dots, j$
         $(V_j \ a\ column\ vector, jth\ row\ of\ V)$
         $o \leftarrow f(w_k^t\, y), K$
         $(w_k \ a\ column\ vector, Kth\ row\ of\ w)$
         $(f(net)\ is\ sigmoid\ function)$

vii. Find error : $E \leftarrow \frac{1}{2}(d-o)^2 + E$

$for\ k = 1, \dots \dots, k$

viii. Error signal vectors of both layers are computed.

$\delta_0\ (output\ layer\ error)\ is\ k * 1,$

$\delta_y (hidden\ layer\ error)\ is\ j * 1$

$$\delta_{ok} = \frac{1}{2}(d_k - o_k)(1 - o_k^2),$$

$for\ k = 1, \dots \dots, k$

$$\delta_{yj} = \frac{1}{2}(1 - y_j^2) \sum_{k=1}^{k} \delta_{ok} w_{kj},$$

$for\ j = 1, \dots \dots, j$

ix. Update weight:

a. Output:

$w_{kj} = \leftarrow w_{kj} + \eta\, \delta_{ok} y_i,$

$k = 1, \dots, k, \quad j = 1, \dots, j$

b. Hidden layer:

$V_{ij} \leftarrow V_{ij} \eta\, \delta_{yj} z_j$

$jk = 1, \dots, j, \quad i = 1, \dots, l$

x. Update weight:

If p<P then p←p+1, q→q+1

Go to step iv, otherwise, go to step ix.

xi. If $E < E_{max}$ the training is terminated, otherwise $E \leftarrow 0, p \leftarrow 1$ go to step iv for a new training cycle.

D. Mathematical Mode:

S= {I, P1, P2, …..,Pn}

Process P1: For RNN;

- P1= {I};
  Where;
  I = input dataset;
  Z (i) = W (i) X + b (i);
  Where; w (i) = weight;
  b (i) = bias;
  x = input;
  P1=z (i);
- Process P2
  P2 = {P1};
  P2 = Neuron Activation
  Activation function
  ht = tanh(Whh*(ht−1)+Wxh*xt);
  Where;
  ht is the hidden state;
  Whh is the parameter for the recurrent (hidden) input;
  Wxh is the parameter for the current input X
  ht−1 is the recurrent input from time t−1
  xt is the current input X at time t
- Process P3
  P3={P2};
  P3 = Error calculation;
  e_j(n)=d_j(n)-y_j(n);
  Where d_j is the target value and
  y_j is the value produced by the perceptron.
  n is the nth neuron
- Process P4
  P4= {P3};
  P4o = Output value
- Process P5
  P5=Repeat Processes P2, P3, P4 until end of file
  Process P6 apply RNN to predict next working day value
- Process P7 for MLP
  P7= {I}
  z(i) = W(i)X + b(i)
  Where;
  w(i) = Weight;
  b(i) = bias;
  x = Input;
  P7=z(i);
- Process P8
  P8={P7}
  P8 = Neuron Activation
  Activation function
  y(v_i) = tanh(v_i) y(v_i) = (1+e^{-v_i})^{-1}
- Process P9
  P9 = {P8}
  Error calculation:
  e_j(n)=d_j(n)-y_j(n),
  Where d_j is the target value and
  y_j is the value produced by the perceptron.
  n is the nth neuron
- Process P10
  P10 = {P9}
  P10o = Output Value
- Process P11
  P11=Repeat Processes P8, P9, P10 until end of file Process P12 apply MLP to predict next working day value.

IV. RESULTS AND DISCUSSION

A. Experimental Setup

Hardware and software of proposed system given below:

- Software Technology:
  1. Technology: Core Java
  2. Tools: JDK 1.8, Netbeans 8.0.2
  3. Operating System: Windows 7
- Hardware Technology
  1. Processor: 1.0 GHz
  2. RAM: 3 GB
  3. Hard Disk: 730 GB

B. Dataset

- Dataset 1: BSE SENSEX dataset from Yahoo finance.
- Dataset 2: The National Stock Exchange (NSE) of India Limited (NSE) is the leading stock exchange of India, located in Mumbai.
- Dataset 3: Bajaj auto ltd dataset.
- For the experiment, recent 1 year of data are considered.

C. Experimental Result

Accuracy calculation of MLP:

Change = Actual Value – Predicted Value

Error (%) = (Change/Actual Value) * 100

Accuracy (%) = 100% - Error

Table II shows that comparison between existing system F-RNN algorithm and proposed system F-MLP algorithm. A proposed technique is more accurate than the existing techniques.

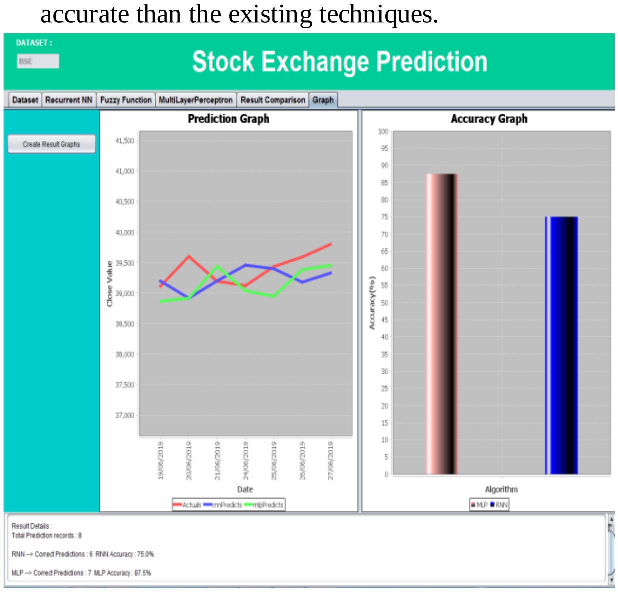

Figure 2: Accuracy Comparison Graph of F-RNN and F-MLP Algorithm

TABLE II: ACCURACY COMPARISON OF F-RNN AND F-MLP ALGORITHM

Algorithm	Accuracy in %
F-RNN	75.0
F-MLP	87.5

Figure 2 shows that the graph of accuracy comparison between existing system F-RNN algorithm and proposed system F-MLP algorithm. Result graph shows that the proposed system is more accurate than the existing system.

V CONCLUSION AND FUTURE SCOPE

A RDNN techniques used the recurrent decision making for the online financial assert trading and environment sensing. This technique is made up two parts DNN and second one is RNN. It improves the robustness of market summarization by the use of fuzzy learning concepts, and it reduces the uncertainty of input data. The deep RL system will be compared with other trading systems under diverse testing conditions. The comparisons show that the DDR system and its fuzzy extension are much robust to different market conditions and could make reliable profits on various future markets. MLP algorithm is used to predict the value of the next day. Fuzzy MLP is will be using to the predicted result. FRNN has some limitations like it takes long computation time because of complex calculations. So FRNN also provides lower accuracy while handling Uncertainties. FMLP overcomes both limitations of FRNN in computation time and accuracy of stock market prediction. It improves the robustness of market summarization by the use of fuzzy learning concepts, and it reduces the uncertainty of input data. The comparisons show that the DDR system and its fuzzy extension are much robust to different market conditions and could make reliable profits on various future markets.

In future scope, incorporating impact of specific events:

- Domestic monitory policy changes
- Impact of other global indices
- Sector specific trends

REFERENCES

[1] Deng Yue, Bao Feng, and Ren Zhiquan, "DEEP DIRECT REINFORCEMENT LEARNING FOR FINANCIAL SIGNAL REPRESENTATION AND TRADING"; IEEE Transactions on Neural Networks and Learning Systems, [2016].

[2] D. Wierstra, J. Hunt and D. Silver "CONTINUOUS CONTROL WITH DEEP REINFORCEMENT LEARNING", arXiv- 1509.02971v5, [2016].

[3] David Lu, "AGENT INSPIRED TRADING USING RECURRENT REINFORCEMENT LEARNING AND LSTM NEURAL NETWORKS", arXiv: 1707.07338 [2016].

[4] James Cumming, "AN INVESTIGATION INTO THE USE OF REINFORCEMENT LEARNING TECHNIQUE WITHIN THE ALGORITHMIC TRADING DOMAIN"; Knowledge Information System, Master's thesis, Imperial College London, United Kiongdom, [2015].

[5] Dongbin Zhao and Yuanheng Zhu, "MEC: A NEAR-OPTIMAL ONLINE REINFORCEMENT LEARNING ALGORITHM FOR CONTINUOUS DETERMINISTIC SYSTEMS", IEEE Transaction on Neural Networks and Learning Sys. Vol-26, no-2, [2015].

[6] Volodymyr Mnih, Daan Wierstra, and Shane Legg, "HUMAN LEVEL CONTROL THROUGH DEEP REINFORCEMENT LEARNING"; Macmillan Publishers Limited, Vol- 5 1 8; Nature, [2015].

[7] Yue Deng, Risheng Liu, and Sanqing Hu, "LOW-RANK STRUCTURE LEARNING VIA NONCONVEX HEURISTIC RECOVERY"; IEEE Transaction on Neural Network and Learning System, Vol-24, no.-3, [2013].

[8] Alex Graves, Geoffrey Hinton, and Abdel-rahman Mohamed, "SPEECH RECOGNITION WITH DEEP RECURRENT NEURAL NETWORK"; IEEE International Conference on Acoustics, Speech and Signal Processing, [2013].

[9] Yue Deng, Zengke Zhang, and Qionghai Dai, "NOISY DEPTH MAP FUSION FOR MULTIVIEW STEREO VIA MATRIX COMPLETION"; IEEE Journal of Selected Topics in Signal Processing, Vol-6, no.-5, [2012].

[10] George Dahl, Li Deng, and Dong Yu, "CONTEXT-DEPENDENT PRE-TRAINED DEEP NEURAL NETWORKS FOR LARGE-VOCABULARY SPEECH RECOGNITION"; IEEE Transaction on Audio Speech and Language Processing, Vol: 20, no.:1, [2012].

[11] Deng Yue, Kong Youyong, Dai Qionghai, and Bao Feng, "SPARSE CODING-INSPIRED OPTIMAL TRADING SYSTEM FOR HFT INDUSTRY"; IEEE Transactions on Industrial Informatics, VOL. 11, NO. 2, [2015].

[12] Lee Honglak, Grosse Roger, and Ranganath Rajesh, "CONVOLUTIONAL DEEP BELIEF NETWORKS FOR SCALABLE UNSUPERVISED LEARNING OF HIERARCHICAL REPRESENTATIONS"; International Conference on Machine Learning, Montreal, Canada, [2009].

[13] Deng Yue, Dai Qionghai, Qian Yanjun, and Li Yipeng, "VISUAL WORDS ASSIGNMENT VIA INFORMATION-THEORETIC MANIFOLD EMBEDDING"; IEEE Transactions on Cybernetics, VOL. 44, NO. 10, [2014].

[14] Deng Yue, Dai Qionghai, and Zhang Zengke, "GRAPH LAPLACE FOR OCCLUDED FACE COMPLETION AND RECOGNITION"; IEEE Transactions on Image Processing, VOL. 20, NO. 8, [2011].