Stock Market Forecasting from Multi-Source Data using Tolerance Based Multi-Agent Deep Reinforcement Learning

C. Bhuvaneshwari, R.Beena

Abstract: ***Analyzing and forecasting the future trends in stock market is challenging due to the ever increasing size of stock data. Modern techniques extract the stock indicators from the web data to forecast the stock movements. However, most previous studies were based on single source of data for extracting these indicators. This might not be effective in obtaining all the possible diverse factors that influence the market movements. Multi-source data has been rarely applied for stock prediction and even those techniques have limitations in handling larger data. In an attempt to utilize multi-source data more effectively for extracting stock indicators and improve the forecasting accuracy of stock movements, this paper developed a stock market forecasting model using Tolerance based Multi-Agent Deep Reinforcement Learning (TMA-DRL) model. The TMA-DRL model effectively combines the quantitative stock data with the indicators i.e. the events extracted from news data and sentiments extracted from tweets. This forecasting model utilizes Random forests to extract the twitter opinions and Restricted Boltzmann Machine (RBM) for event extraction from news data. Combining these indicators, the TMA-DRL model leads to improved data learning and provides highly accurate prediction of future stock trends. Datasets for evaluation were collected from three sources namely Twitter, Market News and Stock exchange, for 12 months period. Evaluation results illustrate the effectiveness of the proposed TMA-DRL stock market forecasting model which makes predictions with high accuracy and less time complexity.***

Keywords: ***Event extraction, Random forests, Restricted Boltzmann Machine, Sentiment analysis, Stock forecasting, Tolerance based multi-agent deep reinforcement learning.***

I. INTRODUCTION

Stock market is one of the most vital components of financial markets and active indicator of the nation's economy [1]. Forecasting the future stock trends using the historical data is tremendously essential for the market investors to understand the possible increase or decrease of the stock price for determining the investment plans. Investors employ many models to analyze the big market data to predict the price forecasting to minimize the investment risks [2]. Yet, the prediction of stock price trends is very challenging and complex due to the varying noise environment and is highly volatile to the daily market values.

The major complexities are relevant to the different factors like political policies, financial news, socio-economic events, yearly earnings reports, trade reports and conflict in the market indicators [3]. As the share traders mostly rely on the technical stock indicators which are poised on a day-to-day basis. Irrespective of the techniques and access to larger rich stock indicators, the forecasting of future stock trends is often more difficult [4], [5]. The unstructured quantitative data and the difficulty in extracting useful indicators from these data make the stock prediction more challenging.

The introduction of machine learning and deep learning algorithms in modern data processing techniques has greatly improved the stock market prediction strategies [6]. Many existing studies have been developed in the past years that aimed at improving the efficacy of stock prices forecasting [7]. However, most research studies were able to extract the indicators and features from only a single source of data. Extensive studies have found that exploiting the stock news from web to extract the events has greater influence the fluctuations in stock markets [8]. Further research also revealed that the sentiment analysis of stock related opinions and survey from people can also have significant impact on the stock price movements [9]. Along with the Natural Language Processing (NLP) techniques, the exploitation of the web news has provided efficient results. However, the techniques are more adaptable to the single source data. As the news events and user opinions can influence the stock price trends, the investigation of the effective approaches for fusing these data together become prominent. The correlations among different sources and the prediction of trends from the multi-source data can possibly outclass forecasting based on single source [10]. This study aims at developing such a technique to utilize the news event extraction and opinion mining from tweets to improve the stock market forecasting.

In this paper, the data is collected from three sources for 12 months period. The stock data is collected from the Stock Exchanges, Bombay Stock Exchange (BSE) and National Stock Exchange (NSE). The news data are collected from the web news portals and the social media data are collected from the Twitter.

The news and tweets are processed using RBM and Random forests, respectively to extract the events and sentiments related to the stock price movements. Then the newly developed TMA-DRL learning model is applied to fuse these indicators with the stock quantitative data to predict the future trends. The major contributions of this paper can be summarized as follows:

- Development of TMA-DRL model for providing accurate and robust forecasting of stock market future trends from the analysis of fused heterogeneous data of

Revised Manuscript Received on February 05, 2020.
***** Correspondence Author

C. Bhuvaneshwari*, Research Scholar, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore, India. Email: bhuvaneshwari.c2020@gmail.com.

Dr.R.Beena, Associate Professor, Department of Computer Science, Kongunadu Arts and Science College, Coimbatore, India Email: beenamridula@yahoo.co.in.

web news, tweets and quantitative data.

- An efficient framework of RBM to extract the events from the news data using the event representation concept.
- A conceptual model of sentiment analysis using Random forests to extract the sentiments from the tweets.
- Multi-source datasets of tweets, news and stock price data collected for 12 months period and applied for the evaluation of the proposed forecasting model.

The remainder of this paper is organized as: Section II presents a discussion on recent related works. Section III explains the feature extraction process from the tweets and news and illustrates the proposed TMA-DRL model. Section IV presents the evaluation results while section V makes a conclusion of this paper and also suggests directions for future works.

II. RELATED WORKS

Many studies have been developed in the recent years that focus on the improving the stock price or trend prediction. Machine learning algorithms were the mostly employed methods for achieving effective forecasting of stock market movements. Das & Padhy [11] proposed the use of Support vector machines (SVM) for prediction of Indian stock prices. Lin et al. [12] also utilized quasi-linear SVM for stock market trend prediction in which better prediction were obtained for single source data. However, the prediction is based on only single factor from the historical stock data. To improve the performance of SVM, particle swarm optimization with uncertain knowledge was employed by Xin et al. [13] for optimizing the SVM parameters. Similarly, Pan et al. [14] included the technical indicators with the SVM. These methods significantly improved the stock prediction accuracy. But these methods also suffered from poor handling of indicators and increased the time complexity along with degradation in prediction accuracy. Huang [15] developed a stock prediction model using Support vector regression (SVR) model in which the SVR parameters were optimized by the genetic algorithm. Similarly, Kazem et al. [16] presented SVR based stock prediction model with chaos-based firefly algorithm. This model employed a novel chaos-based firefly optimization for optimizing the SVR hyper-parameters when a delay coordinate is embedded. The optimized SVR analyses the stock indicators extracted from the historical stock values. These models provided better results and were simpler to implement than the neural networks. But the non-consideration of other stock indicators greatly reduces the overall effectiveness of these models.

Ticknor [17] proposed Bayesian regularized artificial neural network model for the stock market forecasting. This model assigns a probabilistic concept to the network and improves the automatic analysis of the complex structures. This process improved the overall accuracy and reduced the time. However, this model utilized only one source of data and does not include technical indicators for the forecasting. Masoud [18] presented a stock prediction model using artificial neural networks (ANN) on the Libyan stock market data. This model showed that the ANN has better performance than other model when only single source stock indicators are considered. Due to the inefficiency of the SVM and ANN based stock prediction models, Patel et al. [19] suggested prediction models using fusion or hybrid algorithms of SVM, ANN and Random forests for stock prediction. Similarly, Tang et al. [20] proposed a stock price forecasting model using a combination of piecewise linear representation (PLR) and weighted support vector machine (WSVM). This PLR-WSVM model uses a fitness function to select the automatic threshold of PLR and then employs the oversampling method for the stock price turning points. Finally the relative strength index (RSI) is utilized by the optimal WSVM to predict the stock prices. However, these models also does not include important indicators for accurate prediction due to the consideration of only single source data, thus leading to missing points and errors.

Extensive studies showed that the deep learning models and extreme learning machine (ELM) models can be the most appropriate solution to support stock market forecasting from larger stock market data. Hsieh et al. [21] presented stock prediction model using integrated system of wavelet transforms and recurrent neural networks (RNN) optimized by artificial bee colony (ABC) algorithm. This integrated prediction model employs Haar wavelet transform decompose the stock price data and minimizes the noise. Then the RNN algorithm is used to construct the features and the ABC optimizes the RNN parameters to improve the prediction. However, this model has limitations in handing complex time series stock data and the structure is also complex. Additionally, the important indicators are not included except the stock quantitative indicators. Pang et al. [22] proposed an innovative neural network based model for stock market prediction. Instead of the traditional neural networks, this model employed deep long short-term memory neural network (LSTM) with embedded layer and auto-encoder for predicting the stock prices. The model provided 53.2% accuracy in prediction. However, this model also uses only the historical data while ignores the other financial indicators. Another study by Jiang et al. [23] presented a hybrid model of stock prediction using tree-based ensemble models and deep learning algorithms. This model included Random forests, extreme boosting algorithm, RNN, LSTM and bidirectional RNN as the combination algorithms for predicting the stock price future trends. Although very efficient and provided highly accurate prediction, the exclusion of financial indicators from news and market factors negatively impacts the results.

Li et al. [24] developed a stock market prediction model using kernel based ELM. This model extracted the indicators from the raw datasets of market news data and stock price data. The ELM algorithm used the Radial basis function (RBF) kernel function to reduce the training time and improve the prediction accuracy. The comparison results proved that the ELM has better performance than SVM and Back-propagation neural networks (BP-NN). However, this RBF kernel based ELM increases the CPU resource more than the SVM model. Also, the trade-off between CPU scheduling and time for hardware resources is critically worse. Reducing the CPU resources will reduce the prediction speed of ELM and this is the major limitation of this model. In another work, Das et al. [25] proposed a stock market prediction using ELM, Online Sequential Extreme Learning Machine (OSELM) and Recurrent Back Propagation Neural Network (RBPNN) with Firefly algorithm based feature reduction method.

In this approach, the features are extracted and the best features are selected using the firefly evolutionary algorithm. These stock features are used by the classifiers to predict the stock trends. This approach was applied to the four stock market datasets from BSE and NSE and the results showed that the firefly algorithm improved the classifier performance with the OSELM outperforming the other two methods. However, this approach extracted only the stock market price indicators from only one source.

The most feasible and effective approach is the inclusion of most financial factors. Unfortunately only few studies have undertaken this solution. Vargas et al. [26] employed Deep learning algorithm for stock market prediction with the help of news articles. This model utilizes Convolutional Neural Networks (CNN) and RNN deep learning algorithms to improve the stock prediction. This model provided highly improved prediction results largely due to the use of news data along with stock data. Yet, this model requires further fine-tuning as the sentence embedding is average and results in less correlation between the two sources of data. Zhang et al. [27] utilized multi-source multiple instance learning (MIL) for effective stock market prediction. This model utilized news, tweets and stock data for extracting the indicators and employed them to achieve high accuracy in prediction results. However, the multiple instance learning is less accurate and does not perform efficiently when precise classes are not defined.

From the literature studies, it is learnt that the stock price forecasting can be efficient only when multiple financial factors are extracted as indicators. Similarly, the use of complex techniques must also be avoided and suitable techniques must be selected so that multiple source data are processed proficiently. Considering these points, the proposed study has developed an efficient stock market forecasting model using TMA-DRL which fuses the indicators from news, tweets and stock data to provide accurate prediction.

III. METHODOLOGY

The stock market future trends can be predicted by employing the proposed methodology consisting of TMA-DRL, RBM and Random forests. The major events that influence the stock price movements are the trade report, market news and the sentiments of the investing person. For enabling the prediction process, the key factors are collected from the data sources and based on their impact, they are assigned as indicators. These indicators may be influential news events, stock index values or collective sentiments. In order to extract these indicators, the data are formatted into the supportable forms particularly as vectors. The temporal arranged group of news articles, opinions and stock indices are denoted as a multi-source group G. This group of data is processed by the proposed techniques as in the framework given in Fig. 1. The data are represented as vectors for k days from the starting day t. This representation enables the system to factor the processes based on the day-to-day analysis.

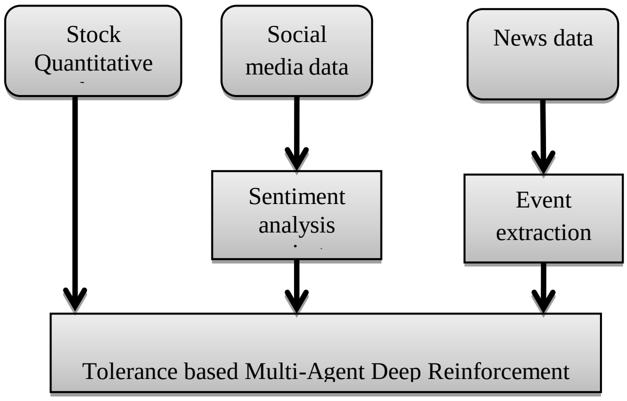

Fig.1. Proposed TMA-DRL model based stock forecasting framework

A. Data Collection and Pre-processing

Data for evaluation was collected at three sources related to Indian Stock markets. The time period fixed is as 12 months starting from October 1, 2018 to September 30, 2019. September 30 is the last day of third quarter of financial year in India. So the data were collected from the first day of fourth quarter of 2018 to the last day of third quarter of 2019 financial year.

Stock market data: The quantitative stock price data were collected directly from BSE and NSE websites for all market days in the specified duration. The BSE 500 and NSE 500 price indices were collected as they provide the stock price of top 500 companies. The data contains the opening, closing, high and low prices of stocks, traded shares and turnover rate of top 500 companies registered in NSE and BSE.

News data: The market and finance news headlines and news prelude are collected from popular financial news website https://www.moneycontrol.com. Initially 47,652 news articles were collected from October 1, 2018 to September 30, 2019 among which the news articles containing non-English sentences are omitted. Finally, a total of 10,026 English language news data were gathered. The extraction of the events requires only the titles as they sum all the essential contents and hence the whole articles are not considered.

Social media data: The social media posts from Twitter are collected for sentiment extraction. The tweets are crawled based on keywords related to shares, stock market and financial tweets using the Twitter developer tool available from the Twitter itself. A total of 176248 tweets were collected initially and after removing the non-English tweets, a total of 66,542 tweets were finalized for evaluation.

The pre-processing is performed for all the three data sources. For the quantitative data, the missing values are checked and the data cleaning is applied to remove if any irrelevant data is present in the collected dataset. For the news data, the non-English news is removed and the data cleaning and case normalization are performed. For the tweets, the non-English tweets are removed. Then the data cleaning, URL removing, punctuation and special symbol removal and case normalization are performed. Stop words removal, stemming and part-of-speech (PoS) tagging are also performed. For each of the trading days, the sentiments from tweets and the news events are used to determine if the stock is increased or decreased.

The training data is employed for learning process and based on these training, the prediction is performed. For each dataset, the data from the first 11 months are used for training and the last month data is used for testing.

B. Feature Extraction

The features are extracted from quantitative stock data, tweets and news datasets. The features from the quantitative stock data are easier to extract as they are often given as the column headings. The news events and tweet sentiments are extracted using the proposed techniques to provide them as input to the TMA-DRL algorithm.

C. News Event Extraction

The news event features are extracted using the RBM technique. First the syntactic analysis method extracts the main structure details of the news sentences. Then these details are employed as the input to the RBM. A pre-trained vector would be the output of the RBM such that they are used as input the sentence2vec model to extract the event representations.

The syntactic structures are obtained by the use of a text parser model. The text parser named HanLP is used for this purpose whose tree based modeling provides sentence structures using the object and subjects of the verbs in the sentence. The core words extracted are connected together to represent the event details. Then the event structures are mapped into a vector as mentioned in the group vector G. The pre-trained model of RBM is used to reconstruct the original events from the vectors. The two layers of the RBM are modeled by the vectors. The input layer is mapped with the encoded m-dimensional vector and is used to estimate the n-dimensional vector of the hidden layer. This output obtained from the RBM is fed as input to the sentence2vec model. The sentence2vec model is trained using the sentence id and then the sentence vector is obtained. These vectors will be used as the features in the proposed TMA-DRL model. For example, the news “Reliance Infra shares loss 14% despite Anil Ambani's assurance on reducing debt”, is applied to the event extraction model. First, the core words are obtained i.e. Reliance Infra, shares, loss, Anil Ambani's, assurance, reducing and debt. Then the encoding is applied to obtain the one vector which is pre-processed by RBM to give 100-dimensional vector. This is finally processed by sentence2vec and is taken as features to the TMA-DRL model.

D. Tweet Sentiment Extraction

Random forests algorithm combines the concepts of random subspaces and bagging. The improvised decision tree algorithm is known as the random forest algorithm that trains on multiple decision trees on subsets of data. Algorithm 1 summarizes the steps involved in Random forests for tweet sentiment extraction.

Algorithm 1: Random forest for tweet sentiment extraction

```
Input: DT= Number of trees, N= Training tweets,
       F= Total features, f= feature subsets
Output: Sentiment class label for tweets
For each tree in forest DT
        Choose bootstrap sample S of size N from
         training tweets
        Generate the recursive repeated tree for each
         node
            Select random f from F set
            Optimally selected best f
            Divide the node
End for
Select created DT trees
Pass test tweets to each tree
Estimate majority votes to assign class labels
Return class labeled tweets
End
```

The tweets dataset consisting of Indian stock market related tweets are used in this process. The tweets are mapped into the random forests and many trees are generated from smaller subsets of the input data. Then the individual tree results are aggregated based on a voting mechanism to generate the tweet sentiments. The use of PoS tagging enables the system to determine the topic words (noun) and background words (other than noun). The major challenge in sentiment extraction is the determination of positivity or negativity. For finding the exact emotions in a tweet, the sentiment words list [28] is used which contains 4818 negative words and 2841 positive words. When a tweet word is not in any of the two words list, they are marked as neutral. In this basis, the tweet words are labeled as either positive or negative.

Initially, the decision trees are generated and the training data is passed into the trees by selecting a bootstrap sample. Then by recursive repeated process for each node, the features are selected from the feature subset to divide the nodes. The divided nodes are used for test data labeling into positive and negative labels.

E. Tolerance based Multi-Agent Deep Reinforcement Learning (TMA-DRL) model

The proposed framework for stock forecasting combines the features extracted from the three data sources and uses the TMA-DRL model to determine the increasing or decreasing stock labels. In the primary step, the features extracted using RBM and Random forests are fed as input to the TMA-DRL along with the quantitative features of stock data. The development of TMA-DRL is based on the application of leniency or tolerant learners on the multi-agent model of deep reinforcement learning. Although, similar models have been developed in recent years like the lenient multi-agent reinforcement learning [29] and lenient multi-agent deep reinforcement learning [30], this proposed model aims at applying the leniency policy in a different manner.

In general, the single-agent DRL models have generally has limited parallel and distributed functioning. Hence the multi-agent DRL has become increasing utilized. However, the experience replay memory (ERM) of the DRL in even two-agent systems fails to converge efficiently due to the moving agent problem. To achieve efficiency in parallel reinforcement learning, the Q-learning and tolerance concept is applied. Fig. 2 shows the application of deep Q-networks and tolerance concepts for the development of Tolerant Deep Q-networks i.e. the working model of TMA-DRL.

Q-networks are modelled based on the Q-learning using the temporal difference learning that yield stochastic outcomes. The function approximation in the state-action space is learned by the experience gained by the agents through training data.

Let θ be the function approximation parameter of Q-networks, a_t is the action from action set A and s_t is the state of function based on bootstrapping the immediate outcomes of the updated Q–function. The immediate outcome r_{t+1} obtained in immediate state s_{t+1} from the next state outcome is given by the Q-function for the bootstrap target Y_t^Q summing the immediate outcome next state.

$$\theta_{t+1} = \theta_t + a(Y_t^Q - Q(s_t, a_t; \theta_t)\nabla_{\theta_t} Q(s_t, a_t; \theta_t) \quad (1)$$

Here $\nabla_{\theta_t} Q(s_t, a_t; \theta_t)$ is the gradient scalar of Q-value and the Q-value moves towards it. The bootstrap target Y_t^Q can be reduced by $\gamma \in [0,1]$ as given below:

$$Y_t^Q = r_{t+1} + \gamma \max_{a\in A} Q(s_{t+1}, a; \theta_t) \quad (2)$$

For deep Q-networks, bootstrap target Y is given by the double Q-learning where the target action chosen using the weight parameter.

$$Y_t = r_{t+1} + \gamma Q\left(s_{t+1}, \operatorname*{argmax}_{a\in A} Q(s_{t+1}, a; \theta_t); \theta_t'\right) \quad (3)$$

Here $\operatorname*{argmax}_{a\in A} Q()$ represents the argument maxima of the Q-function. In this model, the loss function $L(\theta_i)$ due to the tolerance in Q-networks are also formulated by the target

$$L(\theta_i) = E_{s,p}\left|\left(Y_t - Q(s, a; \theta_t)\right)^2\right| \quad (4)$$

Where p is the probability distribution of ERM transitions and $E_{s,p}$ is the time based transition function. In order to provide decentralized learning of Q-networks, the learning rates α and β are introduced such that β < α. This is performed to provide maximized learning without negative outcomes.

The tolerance strategy is applied to the multiple agents to avoid the relative overgeneralization. The tolerance also improves the probability of convergence to obtain global optimal solution for the agents. The training of the tolerant DRL, the agents updates the outcomes by reducing the Q-value of state-action pair (s,a). Then the tolerance $tol(s_t, a_t)$ and temperature function $Temp(s_t, a_t)$ determine the Q-value of (s,a). The tolerance function is given as

$$tol(s_t, a_t) = 1 - e^{-K*Temp(s_t, a_t)} \quad (5)$$

Here K is the tolerance control factor that controls impact of the temperature function on the tolerance level. After the update, the temperature value will be reduced by a minimization factor $\beta \in [0,1]$ such that $Temp_{t+1}(s_t, a_t) = \beta Temp_t(s_t, a_t)$. When a temperature decay error function δ is introduced, the tolerance will be applied to DRL Q-value update function as

$$Q(s_t, a_t) = \begin{cases} Q(s_t, a_t) + \alpha\delta & when\,\delta > 0 \\ Q(s_t, a_t) & when\,\delta \le 0 \end{cases} \quad (6)$$

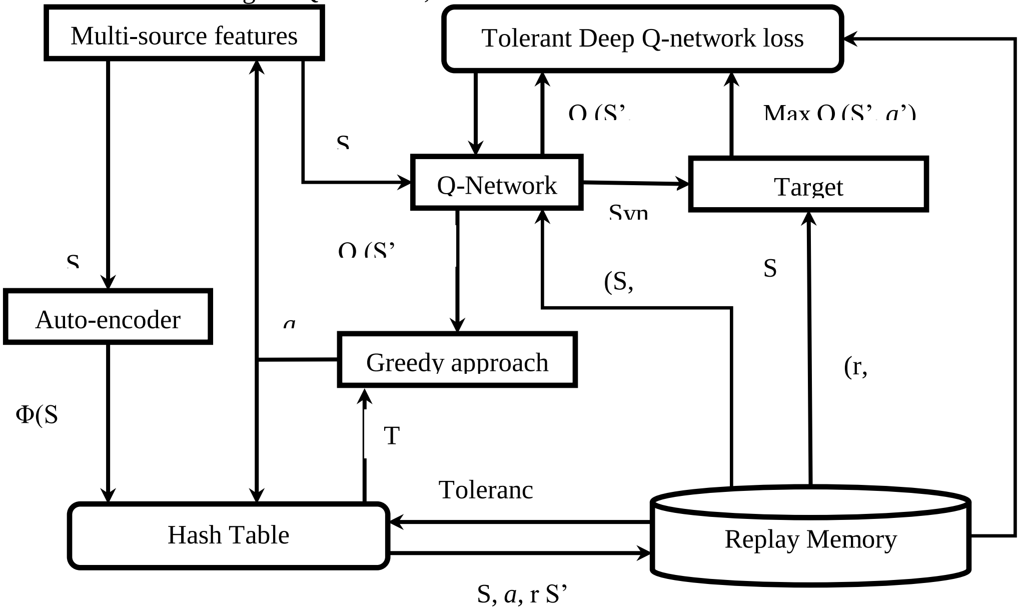

Fig.2. Architecture of TMA-DRL

By combining the tolerance and deep Q-networks, the proposed TMA-DRL is achieved by the utilization of temperature and tolerance function on a standard Q-network. Applying the tolerance strategy to DRL, a bias might be developed in the initial iterations. This bias will result in poor outcomes and hence must be resolved using alternate methods. The tolerance amount is estimated at time t within a specified tuple $\left(s_{t-1}, a_{t-1}, r_t\ , s_t, tol(s_t, a_t)_t\right)$

of ERM.

It can be estimated by using the temperature value $Temp$ and the associated hash-key $\phi(s)$ for the state-action pair (s,a). It is given by

$$tol(s,t)=1-e^{-K*Temp(\phi(s),a)} \tag{7}$$

Here $(\phi(s),a)$ is mapped to the temperature value using a lexicon and the hash-keys are estimated. In case, the temperature value for a particular pair of $(\phi(s),a)$ is not present in the lexicon, then the temperature value is set as Maximum value for that pair. The loss function for each sample is also estimated and the samples that do not meet the tolerance condition in Eq. (6) are discounted. After mapping the temperature values, the scheduled temperature decay is performed to ensure the agent's tolerance to be adaptable to semantic similarity in any future state.

The temperature decay must not be rapid and some schedule must be followed such that it avoids premature convergence or cooling. The decaying maximum temperature is denoted as $Temp_{max}$. It is actually pre-estimated as a schedule $(\beta_0,\beta_1,\ldots,\beta_n)$ where n is the step unit. The values of β are estimated using a decay exponent ρ and decay rate d such that the value β < α learning rate in deep networks.

$$\beta_n = e^{\rho\times d^t} \qquad 0\le t<n \tag{8}$$

This process continues until the terminal state is reached. The main objective in applying this process is to avoid premature cooling and the slowed rate of cooling can improve the performance. The sampling process is majorly performed to extract the increasing and decreasing stock forecasting for future trends. Algorithm 2 summarizes the processes in TMA-DRL.

Algorithm 2: TMA-DRL model

Input: Quantitative, events and sentiment features, learning rate α and β, number of agents, weight θ, Maximum iterations, Initial Temp, t

```
$t \leftarrow 0$
Initialize all agents
  For iteration $iter=0, iter\_max$ do
    Reset gradients $\nabla_{\theta_t}$
    Synchronize weights
    Sample tuple from ERM for all agents
    Repeat
  For each agent
  Estimate the loss function, data training rate
  Sample action $a_t$ at state $s_t$
  Map each state-action pair to network agents
   Build communication map
  End for
  Take action $a_{t+1}$ for all agents
  Estimate tolerance amount for each state
  Model outcomes r
  $t \leftarrow t+1$
  Until terminal state for all agents
  $n \leftarrow 0$, steps←max steps
  For $i = steps\ to\ 0$ do
If $\beta_n Temp_t(\phi(s)_i, a_i) < Temp_{max}$ then
 $Temp_{t+1}(\phi(s)_i, a_i) \leftarrow \beta_n Temp_t(\phi(s)_i, a_i)$
Else
 $Temp_{t+1}(\phi(s)_i, a_i) \leftarrow Temp_{max}$
End if
 $n \leftarrow n+1$
End for
Update weights for all agents
Return gradients and loss
End for
```

IV. PERFORMANCE EVALUATION

The performance of the proposed stock forecasting framework using the TMA-DRL model is evaluated and compared with that of the existing stock prediction models. The performance metrics considered are the accuracy, precision, recall, F1- score and execution time. The existing methods considered for comparison are: SVM [11], ANN [18], ELM [24] and MIL [27]. These methods are selected in the sense that the SVM and ANN are the baseline methods most commonly used. The hybrid approach is the ELM while MIL is the most recent model utilizing multi-source data for evaluation. Table.1 shows the forecasting results obtained for the proposed TMA-DRL and the existing models.

Table- I: Stock forecasting results

Metrics	**SVM [11]**	**ANN [18]**	**ELM [24]**	**MIL [27]**	**TMA-DRL**
Accuracy (%)	59.45	62.28	68.546	72.55	79.778
Precision (%)	67.35	69.08	74.446	78.00	80.04
Recall (%)	60.54	62.80	70.32	72.67	80.60
F1-score (%)	71.10	72.93	80.19	82.73	89.50
Time (s)	378.686	357.93	323.05	319.58	151.95

From table I, it can be seen that the performance metrics of the proposed TMA-DRL has higher values of accuracy, precision, recall and F1- score. Also, the time for execution is also less for the proposed model. From the comparison analysis, it is learnt that the best prediction models can be ordered as TMA-DRL, MIL, ELM, ANN and SVM. The proposed TMA-DRL model has produced accuracy of 79.778% which is 7.228% greater than the second best MIL. Similarly, TMA-DRL has 2.04% higher precision, 7.93% greater recall and 6.77% greater F1-score than the MIL model. The proposed model also consumes less time of 151.95 seconds which is half the time consumed by MIL.

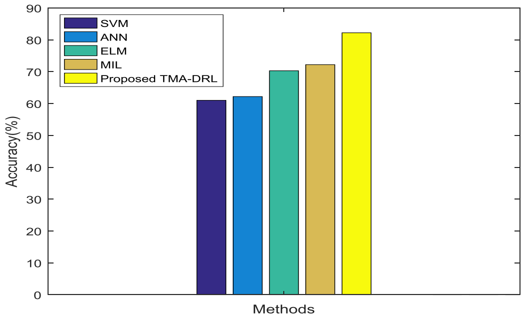

Fig.3. Accuracy Comparison

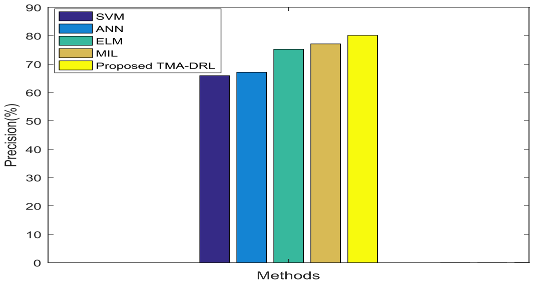

Fig.4. Precision Comparison

Fig. 3 shows the accuracy comparison and Fig. 4 shows the precision comparison of the stock prediction models. It can be seen that the performance of the proposed TMA-DRL model based stock forecasting has higher accuracy and precision values compared to their existing models. This is mainly due to the bias-free performance of the learning process in tolerance based DRL. The tolerance concept has greatly reduced the temperature decay and hence the accuracy and precision are improved.

Fig. 5 shows the recall comparison and Fig. 6 shows the F1-score comparison of the stock prediction models. Similar to accuracy and precision, the TIMA-DRL model has increased the values of performance metrics mainly depending upon the effective feature extraction from the news and tweets. In addition, the consideration of tolerance factor has minimized the errors in prediction which is evident from the better performance results.

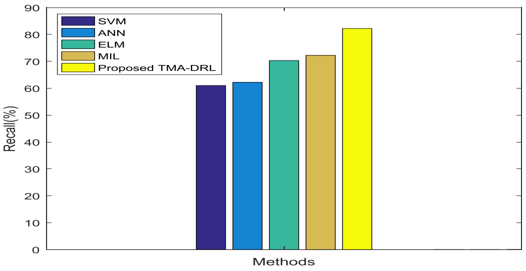

Fig.5. Recall Comparison

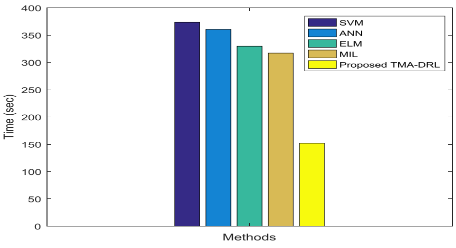

Fig.6. F1-score Comparison

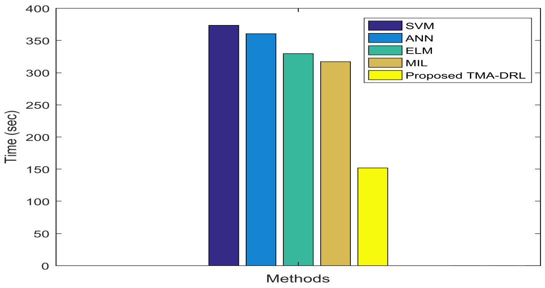

Fig.7. Time Comparison

Fig. 7 shows the execution time comparison of the stock prediction models. Similar to other metrics, the proposed TMA-DRL model has shown better performance with less time. As the proposed TMA-DRL employs the deep learning, the distributed and parallel computation property is ensured. This has resulted in the greater reduction of time complexity which is reflected in the lesser execution time. Thus from the performance evaluation results, it can be concluded that the proposed model of TMA-DRL based stock forecasting using the multi-source data has performed efficiently and best suited for real-time applications.

V. CONCLUSION

This paper aimed at developing an efficient stock market price movements forecasting model to help the investors and economists in determining the future trends in stock markets. The proposed stock forecasting model utilized indicators extracted from multiple sources of news, social media posts and quantitative stock data. The data were collected for a period of 12 months. These indicators were extracted using efficient techniques and were fed as input to the TMA-DRL model to learn the trends of stock price movements. Based on the learned trends, the future trends were predicted by the TMA-DRL model with 79.778% accuracy. The prediction time is also very less compared to the existing stock prediction models. Thus the proposed study efficiently performed the multi-source data based stock market forecasting with high efficiency. In the future, the impact of considering additional indicators will be investigated. Also, the possibility of combining other efficient machine learning and deep learning algorithms without increasing the complexity will be examined.

REFERENCES

1. A. Antonios,”Stock market and economic growth: an empirical analysis for Germany,” Business and Economics Journal, vol. 2010, pp. 1-12, 2010.
2. B. Ake, “The role of stock market development in economic growth: evidence from some Euronext countries,” International Journal of Financial Research, vol. 1, no. 1, pp. 14-20, 2010.
3. G. S. Atsalakis and K.P. Valavanis,“Surveying stock market forecasting techniques–Part II: Soft computing methods,” Expert Systems with Applications, vol. 36, no. 3, pp. 5932-5941, 2009.
4. D. Enke and S. Thawornwong, “The use of data mining and neural networks for forecasting stock market returns,” Expert Systems with applications, vol. 29, no. 4, pp. 927-940, 2005.
5. Z. Q. Jiang, W. X. Zhou, D. Sornette, R. Woodard, K. Bastiaensen and P. Cauwels, “Bubble diagnosis and prediction of the 2005–2007 and 2008–2009 Chinese stock market bubbles,” Journal of Economic Behavior and Organization, vol. 74, no. 3, pp. 149-162, 2010

6. V. H. Shah, "Machine learning techniques for stock prediction," Foundations of Machine Learning| Spring, vol. 1, no. 1, pp. 6-12, 2007.
7. J. Patel, S. Shah, P. Thakkar and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," Expert systems with applications, vol. 42, no. 1, pp. 259-268, 2015.
8. R. P. Schumaker and H. Chen, "A quantitative stock prediction system based on financial news," Information Processing and Management, vol. 45, no. 5, pp. 571-583, 2009.
9. T. H. Nguyen, K. Shirai and J. Velcin, "Sentiment analysis on social media for stock movement prediction," Expert Systems with Applications, vol. 42, no. 24, pp. 9603-9611, 2015.
10. X. Zhang, Y. Zhang, S. Wang, Y. Yao, S. Y. Fang Band Philip, "Improving stock market prediction via heterogeneous information fusion," Knowledge-Based Systems, vol. 143, pp. 236-247, 2018.
11. S. P. Das and S. Padhy, "Support vector machines for prediction of futures prices in Indian stock market," International Journal of Computer Applications, vol. 41, no. 3, 2012.
12. Y. Lin, H. Guo and J. Hu, "An SVM-based approach for stock market trend prediction," In The 2013 international joint conference on neural networks (IJCNN), IEEE, pp. 1-7, 2013.
13. J. Xin, K. Yuhong and Z. Keyi, "Stock price predicting using SVM optimized by particle swarm optimization based on uncertain knowledge," International Journal of Digital Content Technology and its Applications, vol. 6, no. 23, p. 216, 2012.
14. J. Pan, Y. Zhuang and S. Fong, "The impact of data normalization on stock market prediction: using SVM and technical indicators," In International Conference on Soft Computing in Data Science, Springer, Singapore, pp. 72-88, 2016.
15. C. F. Huang, "A hybrid stock selection model using genetic algorithms and support vector regression," Applied Soft Computing, vol. 12, no. 2, pp. 807-818, 2012.
16. A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi and O. K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," Applied soft computing, vol. 13, no. 2, pp. 947-958, 2013.
17. J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," Expert Systems with Applications, vol. 40, no. 14, pp. 5501-5506, 2013.
18. N. Masoud, "Predicting direction of stock prices index movement using artificial neural networks: The case of Libyan financial market," Journal of Economics, Management and Trade, pp. 597-619, 2014.
19. J. Patel, S. Shah, P. Thakkar and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," Expert Systems with Applications, vol. 42, no. 4, pp. 2162-2172, 2015.
20. H. Tang, P. Dong and Y. Shi, "A new approach of integrating piecewise linear representation and weighted support vector machine for forecasting stock turning points," Applied Soft Computing, vol. 78, pp. 685-696, 2019.
21. T. J. Hsieh, H. F. Hsiao and W. C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm," Applied soft computing, vol. 11, no. 2, pp. 2510-2525, 2011.
22. X. Pang, Y. Zhou, P. Wang, W. Lin and V. Chang, "An innovative neural network approach for stock market prediction," The Journal of Supercomputing, pp. 1-21, 2018.
23. M. Jiang, J. Liu, L. Zhang and C. Liu, "An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms," Physics A: Statistical Mechanics and its Applications, vol. 541, p. 122272, 2020.
24. X. Li, H. Xie, R. Wang, Y. Cai, J. Cao, F. Wang and X. Deng, "Empirical analysis: stock market prediction via extreme learning machine," Neural Computing and Applications, vol. 27, no. 1, pp. 67-78, 2016.
25. S. R. Das, D. Mishra and M. Rout, "Stock market prediction using Firefly algorithm with evolutionary framework optimized feature reduction for OSELM method," Expert Systems with Applications: vol. X, no. 4, p. 100016, 2019.
26. M. R. Vargas, B. S. De Lima and A. G. Evsukoff, " Deep learning for stock market prediction from financial news articles," In 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), IEEE, pp. 60-65, 2017.
27. X. Zhang, S. Qu, J. Huang, B. Fang and P. Yu, "Stock market prediction via multi-source multiple instance learning," IEEE Access, vol. 6, pp. 50720-50728, 2018.
28. "Sentiment words list Opinion Mining, Sentiment Analysis and Opinion Spam Detection", http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
29. E. Wei and S. Luke, "Lenient learning in independent-learner stochastic cooperative games," The Journal of Machine Learning Research, vol. 17, no. 1, pp. 2914-2955, 2016.
30. G. Palmer, K. Tuyls, D. Bloembergen and R. Savani, "Lenient multi-agent deep reinforcement learning", In Proceedings of the 17th International Conference on Autonomous Agents and Multi Agent Systems," International Foundation for Autonomous Agents and Multi-agent Systems, pp. 443-451, 2018.

AUTHORS PROFILE

C. Bhuvaneshwari is working as an Assistant Professor in the Department of Computer Science, Dr.G.R.Damodaran College of Science. She has obtained her M. Phil degree from Bharathiar University, Coimbatore. Her areas of interests are Data Mining, Big Data, and Machine Learning. Email: bhuvaneshwari.c2020@gmail.com

Dr. R. Beena is working as an Associate Professor and Head of the Department of Computer Science, Kongunadu Arts and Science College. She has received her PhD degree from Bharathiar University, Coimbatore. Her research interest includes software engineering, data mining and networking concepts. She has 20 years of Teaching Experience and published many articles in international journals and conferences. Email: beenamridula@yahoo.co.in