Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-020-04782-2

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Multi model-Based Hybrid Prediction Algorithm (MM-HPA) for Stock Market Prices Prediction Framework (SMPPF)

Subba Rao Polamuri[1] · Kudipudi Srinivas[2] · A. Krishna Mohan[3]

Received: 3 April 2020 / Accepted: 5 July 2020

Abstract

In financial arena, stock markets have influence on the performance of organizations and investors. Stock markets are highly dynamic in nature, and predicting the stock prices is a challenging task. Reliable prediction needs a superior method for drawing inferences from the data available. In machine learning domain, artificial neural networks are considered as accurate prediction models but using these techniques to forecast the stock market price up to some extent. So, there is a need to improve the accuracy of the model. Towards this end a framework known as Stock Market Prices Prediction Framework (SMPPF) is proposed. This framework has the underlying algorithm known as Multi-Model based Hybrid Prediction Algorithm (MM-HPA) that is the combination of linear and non-linear models including Genetic Algorithm (GA). It is the combination of linear and nonlinear models including genetic algorithm (GA). In linear model, use autoregressive moving average model and in nonlinear model, use the deep learning model known as recurring neural network. Finally apply the GA for finding optimal parameters for the hybrid model. The proposed framework is evaluated with a prototype built on Python data science platform. The empirical results revealed that the difficulty of traditional models in capturing patterns from nonlinear data is overcome by the proposed hybrid model.

Keywords Deep learning · Stock returns prediction · Linear model · nonlinear model · Genetic Algorithm · Artificial Neural Network and Recurrent Neural Network

1 Introduction

Stock markets play vital role in financial domain of any nation. They have importance both nationally and internationally. Enterprises in the real world need to be aware of their stocks and growth of them. At the same time investors and other stakeholders wanted to know the trends and patterns found in the stock markets [1]. Stock market prices prediction influences all the stakeholders. Therefore, accuracy in prediction is crucial for wellbeing of the stakeholders. Both industries and academia are interested in the research of stock markets. In the presence of noisy and dynamic nature of stock prices, prediction of stock prices in future is challenging problem on which researchers are striving to improve prediction models [2]. There are linear models and nonlinear models used for stock prices prediction. Most of the nonlinear models come from deep learning. The machine learning (ML) techniques available are exploited by researchers for number of years. However, deep learning is relatively novel concept in forecasting stock prices.

The literature has revealed rich information about various methods used in predicting stock prices. Convolutional neural network (CNN) is one of the widely used models as explored in [1, 3–6]. CNN with Deep Q Network is employed in [4]. Long short-term memory (LSTM) is another technique used in deep learning for forecasting stock prices. It is used by various researchers as found in [2, 3, 5, 7–11]. LSTM with automatic stock trading is explored in [7] while LSTM with CNN combination in [3], LSTM with CNN along with feature fusion is employed in

✉ Subba Rao Polamuri
psr.subbu546@gmail.com

Kudipudi Srinivas
vrdrks@gmail.com

A. Krishna Mohan
krishna.ankala@gmail.com

1 Department of CSE, JNTUK, Kakinada, AP, India

2 Department of CSE, V. R Siddhartha Engineering College, Vijayawada, AP, India

3 Department of CSE, UCEK, JNTUK, Kakinada, AP, India

Published online: 28 July 2020

[8]. LSTM with adversarial training and attentive approach is investigated in [9] for enhancing prediction performance. LSTM, CNN and Conv1D-LSTM combination is used in [5] while innovation is achieved with LSTM and NN combination in [10]. Market state model is employed in [12] while information fusion approach is found in [13]. Deep learning with stock indicators along with 2D Principal Component Analysis (PCA) is used for making a closing price prediction system in [14]. The concept of trend lines as tool for stock prediction is employed in [15]. The combination of multilayer perceptron (MLP) and deep belief network (DBN) is used in [16] for enhancing stock prediction performance. From the literature, especially from [17–19], it is found that either traditional models or deep learning models will not be able to provide optimized performance in stock prices prediction. It is advocated that a hybrid model uses both traditional and deep learning models with employment of evolutionary approaches for optimal weights computation in order to reduce time taken by deep learning models.

The proposed hybrid model is divided into three phases. In first phase is creation and implementation of propose model called Multi-model-Based Hybrid Prediction Algorithm (MM-HPA), this is the combination of linear and nonlinear models including genetic algorithm (GA) for improving performance of stock prices prediction. In second phase is development of Stock Market Prices Prediction Framework (SMPPF). In third phase is performance evaluation of the proposed model.

1. Proposed a framework known as Stock Market Prices Prediction Framework (SMPPF) for improving performance of stock prices prediction.
2. Proposed a hybrid prediction model known as Multi-model-Based Hybrid Prediction Algorithm (MM-HPA) that is the combination of linear and nonlinear models including genetic algorithm (GA).
3. A prototype application is built to demonstrate proof of the concept that realizes the proposition found in the literature that reads “hybrid approaches that exploit both traditional and deep learning models along with evolutionary method for optimizing weights lead to superior performance in stock prices prediction”.

The remainder of the paper is structured as follows. Section 2 reviews literature on different prediction models such as linear, nonlinear and hybrid models for stock returns prediction. Section 3 presents preliminaries to ascertain know-how on the models used for the proposed framework. Section 4 presents the proposed Stock Market Prices Prediction Framework (SMPPF) with underlying hybrid algorithm. Section 5 presents experimental results and discussion. Section 6 concludes the paper and provides directions for future work.

2 Related Work

This section reviews literature on different stock returns prediction models that are related to our work. dos Santos Pinheiro and Dras [1] proposed a deep learning-based hypothesis known as Efficient Markets Hypothesis (EMH) that is associated with economic theory. They used RNN model with language and character level pre-training for forecasting inter-day and intra-day market dynamics. However, they envisaged risk of character embedding and intended to investigate further on their method. Yang et al. [2] explored high frequency price–volume data with deep learning model for efficient forecasting of stock returns. Unlike character and language embedding studied in [1], they focused on a methodology based on long short-term memory (LSTM). The main advantage of this method lies in its ability to deal with issues that are sensitive to time-series data. It also could deal with emotion analysis and language translation. It has intuition and imitation of memory possessed by humans. In future, they intended to investigate on avoiding more transaction fee in order to maximize profits with reinforcement learning.

Fister et al. [7] also employed deep learning-based LSTM model for superior trading strategy. Automatic stock trading is the main focus of their research. They used mechanical trading system to evaluate the model. They investigated on five different strategies related to automatic trading. The trading strategies they observed include surrogate trading with machine learning (ML), relative strength index (RSI) which is rule based and passive. They intended to improve their method in future with three approaches such as considering longer sequence length, hyper parameter tuning and an improved online surrogate model. Oncharoen and Vateekul [3] exploited deep learning models with technical indicators and event embedding. They found that traditional models could not incorporate event embedding but used numerical information and textual information for stock market predictions. The authors employed both LSTM and CNN models in order to increase accuracy in prediction. They normalized technical indicators besides using event embedding to arrive at a better prediction model. In future, they wanted to improve it further with RNN and attention mechanisms.

Lee et al. [4] investigated on CNN along with Deep Q-Network for improving stock prediction performance. Their method takes stock chart images as input and predicts global stock markets. Different countries stock market data are used for training. It could provide not only profits in global markets but also predicts profit probabilities of specific country with whose data the model is trained. They observed that ML techniques are employed for single market predictions and they intended to use them for predicting global markets in future. Kim and Kim [8] explored different representations of same data with deep learning models such as LSTM-CNN along with the concept of feature fusion. This model combines features of CNN and features of LSTM and outperforms single models in stock prediction. They observed that their method could be improved further in future by considering technical indicators and more representations of the data being used. Chong et al. [17] focused on different learning models and deep learning networks for finding differences and useful insights. They also covered autoencoders that are very useful in predictions. They observed that in future, there need to be more network functions and data representations with hybrid approaches for better predictions. Autoencoders are also used by Shengnan [20] for stock prices prediction.

Chen et al. [12] explored investment behaviours and stock intrinsic properties for improving stock trend prediction. They found that investment behaviour in the mutual fund portfolios have latent representations that exhibit intrinsic properties which are very useful for stock prediction. Dynamic market state and trend was proposed by them using such stock properties and discover correlations between given stock and the market dynamics. Afterwards, they used dynamic stock indicators to aggregate correlations in order to improve accuracy in prediction. In future, they intend to improve their market state model with more useful investment behaviours discovering intrinsic properties from diversified sources of data. Feng et al. [9, 21] investigated the problems such as generalization difficulty in stock prediction and stochastic challenge. Towards a solution they employed an adversarial model along with attention model combined with LSTM. Therefore, their proposed method is known as adversarial attentive LSTM. Their method could improve single models in performance. They intend to evaluate their method with commodities which explore different deep learning structures like CNN in future for reaping benefits in prediction. Chong et al. [17] on the other hand made an objective analysis of different stock market prediction methods based on deep learning. They opined that deep learning methods in combination of other methods may give rise to more effective solutions in future.

Jain et al. [5] employed combination of different deep learning methods like LSTM, CNN and Conv1D-LSTM for improving prediction accuracy of daily stocks. They found better performance when Conv1D-LSTM is used. In future, they intend to combine both regression and classification models for better prediction. Gudelek et al. [6] employed 2-D CNN for stock prices prediction. They used Exchange-Traded Funds (ETFs) for the empirical study. With respect to buy and hold decision making, their method was found superior. In future, they intend to improve buy-sell strategy algorithm, increase image size and incorporate feature selections for further enhancement. Lahmiri [13] employed information fusion model for predicting stock prices more effectively. Thus, they made a prediction system based on ensemble of neural networks (NN). They found that NN ensemble could provide better results than individual NN models. In future, they intended to improve their model with heterogeneous ensemble approach with deep learning. Pang et al. [10] proposed an innovation in deep learning models with LSTM. Instead of single stock model, they used multi-stock high dimensional data for automatic encoding along with embedder layer for better performance. In future, they intended to improve it by adding text information feature. Gao et al. [14] investigated on stock indicators and deep learning models along with 2D Principal Component Analysis (PCA) for improving closing price prediction. Thus, they arrived at a closing price prediction system that could perform better than the traditional models.

Jitpakdee and Pravithana [15] investigated on technical analysis tools known as trend lines to predict and analyse security and stock patterns. They incorporated different rules to arrive at trends lines. However, they found that it could lead to superior results if the trend lines are associated deep learning in future. Li and Liao [22] also focused on the trend analysis using different existing methods. Rechenthin [23] employed ML-based classification methods for stock prices prediction. However, deep learning which is promising for better prediction performance is not explored. Batres-Estrada [16] employed NN known as multilayer perceptron (MLP) along with deep learning model known as deep belief network (DBN) for better prediction of stock trends. Fisher and Krauss [11] discovered that LSTM could provide better results when compared with traditional methods known as Logistic Regression (LG) and Random Forest (RF) [24]. Maqsoon et al. [18] proposed a deep learning-based prediction model based on local and global event sentiments. They found that deep learning models take lot of time and that can be reduced with finding optimal weights using evolutionary models.

Zhong and Enke [19] focused on stock daily return prediction using hybrid ML models that involve DNNs. They also used PCA for data transformations. ANNs are used for pattern recognition while DNNs are used for classification. They found that DNNs take more time in prediction and that needs to be overcome in future. Tsantekidisa et al. [25] explored many deep learning models for stock price prediction. Fang et al. [26] proposed deep learning model for predicting cryptocurrency prices. From the literature, it is understood that either traditional models or deep learning models will not be able to provide optimized performance in stock prices prediction. It is advocated that a hybrid model that uses both traditional and deep learning models with employment of evolutionary approaches for optimal weights computation in order to reduce time taken by deep learning models. This is the main focus of this research paper that incorporates a hybrid prediction model with linear and nonlinear models along with evolutionary approach.

3 Preliminaries

Prior to realizing the proposed Multi-model-Based Hybrid Prediction Algorithm (MM-HPA), this section provides details of other models based on which the proposed model is built for averaging stock prediction performance. The MM-HPA is based on two linear and one nonlinear models. The linear models are known as autoregressive moving average model and exponential smoothing model.

3.1 Autoregressive Moving Average Prediction Model

Both autoregressive and moving average models are integrated to perform stock returns prediction. As the historical data points are used in the time-series data, it is essential to have better prediction that is achieved by combining the two models. When the given data is subjected to regression, the results are used further. The non-stationary data are converted into stationary data with differencing approach, and the result is indicated by d.

Algorithm 1: Autoregressive moving average model for stock prediction

1. Find p order of Autoregressive model
2. Find q order of moving average model
3. Estimation using p, d and q
4. Forecasting

Algorithm 1: Autoregressive moving average model for stock prediction

As presented in Algorithm 1, after regression for finding p value, the q value is known with moving average model indicating the moving average nature of the model. It also reflects the number of lagged values pertaining to error term. Afterwards, the forecasting of stock returns is completed.

Provided a time series stock market data, it can be represented as in Eq. 1.

$$R = \left(r_{t-(T-1)}, \ldots, r_{t-1,} r_t\right) \tag{1}$$

where T denotes the number of historical series of data, r_t denotes refers to actual value and R denoted pass returns. The autoregressive moving average model considers order of regression denoted as p and the delay with respect to a reference location is denoted as k. Its prediction is achieved as in Eq. 2

$$r_{t+1} - z = \Re\left(\widehat{r_{t-(p-1)}} - z, \ldots, r_{t-1} - z, r_t - z\right) \tag{2}$$

where autoregressive predictor is denoted as $\Re$ and z denotes moving reference expression. The prediction result is denoted as $\widehat{r_{t+1}} - z$, t denotes time, the moving reference expression is denoted as z and r_t refers to the actual value. The value of z is computed as in Eq. 3.

$$z = r_{t-(p-1)-k} \tag{3}$$

k denotes reference location. The final prediction for future value is denoted as $\hat{r}_{t+1}$ and computed as in Eq. 4

$$\hat{r}_{t+1} = \widehat{r_{t+1}} - z + z \tag{4}$$

A simple autoregressive model can be expressed as in Eq. 5 while its MA model is expressed as in Eq. 6.

$$r_t = \omega_1 r_{t-1} + \omega_2 r_{t-2} + \cdots + \omega_p r_{t-p} + \varepsilon_t \tag{5}$$

$$r_t = \varepsilon_t - \varphi_1 \varepsilon_{t-1} - \varphi_2 \varepsilon_{t-2} - \cdots \varphi_q \varepsilon_{t-q} \tag{6}$$

The combination of autoregressive model and MA model is expressed as in Eq. 7

$$\begin{aligned} r_t = {} & \omega_1 r_{t-1} + \omega_2 r_{t-2} + \cdots + \omega_p r_{t-p} + \varepsilon_t \\ & - \varphi_1 \varepsilon_{t-1} - \varphi_2 \varepsilon_{t-2} - \cdots \varphi_q \varepsilon_{t-q} \end{aligned} \tag{7}$$

where two coefficients are denoted as ω and φ while random errors is denoted as ε_t. When it comes to exponential smoothing, it is a linear model that considers historical time-series data and computes geometric sum as in Eq. 8.

$$\hat{r}_{t+1} = \hat{r}_t + \alpha\left(r_t - \hat{r}_t\right) \tag{8}$$

where the prediction for future value is denoted as $\hat{r}_{t+1}$ and the smoothing factor is denoted as α which is in the range between 0 and 1. The prediction error on the other hand is denoted as $r_t - \hat{r}_t$.

3.2 Recurrent Neural Network Model for Stock Prediction

Algorithm 2: RNN model for stock prediction

1. Load stock dataset
2. Data pre-processing
3. Feature extraction
4. Training RNN
5. Prediction of Stock Returns
6. Output generation
7. Visualization

Algorithm 2: RNN model for stock prediction

As presented in Algorithm 2, the training RNN results in making a knowledge model that is further used to perform prediction of stocks on the test data. Afterwards, the results are prepared for further visualization.

RNN model is deep learning-based model used for predicting stock returns. This model has many phases. In the first phase, raw data of stock data are loaded into a data structure in main memory. Afterwards, the data are subjected to pre-processing that includes filling in missing values and determining training and testing sets. Here is the RNN algorithm for realizing stock prediction model.

A deep learning model-based recurring neural network (RNN) is trained and used for prediction of stock returns. The network has a hidden layer and an output layer. It takes $(r_t - Z, r_{t-1} - Z, \ldots, r_{t-(p-1)} - Z)$ as input and returns $\widehat{r_{t+1}} - Z$ as output. Input layer exploits long-term memory to hold data and later on pass it to hidden layer. The neurons in the input layer exhibit order p and it has linear activation function. There are sixteen neurons used in the hidden layer. They are denoted as $(h_{1,1}, \ldots, h_{1,16})$. And there is a single neuron in the input layer. However, both layers have logistic activation function. Thus, a network is formed capable of learning and predicting very complex patterns that are nonlinear in nature. It also has algorithm for back propagation. RNN is a class of artificial neural network (ANN). In RNN, a directed graph is formed with temporal sequence to denote the connections between nodes. Moreover, RNNs are capable of using their internal state (memory) that helps in processing of inputs such as variable length sequences. This feature of RNN makes it suitable to solve wide range of complex real world problems.

To increase throughput and current data analysis of our downlink scheduling protocol in wireless communication, we need to add some upper and lower bound advancements for wireless communication in data transmission. The next section describes the computational evolution of downlink protocol hierarchy operations.

4 Proposed Prediction Framework

This section provides details of the proposed framework and the mechanism used in the underlying algorithm known as MM-HPA besides evaluation metrics.

4.1 The Framework

A framework is proposed to have an effective prediction for stock returns. It is named as Stock Market Prices Prediction Framework (SMPPF). The framework facilitates exploitation of linear and nonlinear prediction models besides making a hybrid of them for leveraging stock prediction performance reduce prediction error significantly. The nonlinear nature of time-series data is intelligently analysed by the framework with its underlying hybrid method named as Multi-model-Based Hybrid Prediction Algorithm (MM-HPA). This algorithm is based on the results of different prediction models. First, linear models such as autoregressive moving average and exponential smoothing are used in order to gain results. Afterwards, the results of autoregressive moving average model are given as training set to the deep learning model. The deep learning prediction model uses recurring neural network (RNN) that takes training set from a linear model. Then, the results of linear and nonlinear models (deep

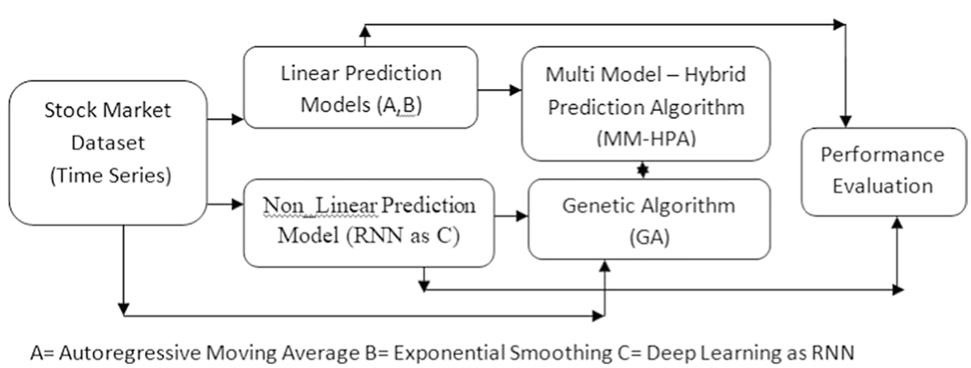


Fig. 1 Overview of the proposed prediction framework

learning as RNN) are given input to the MM-HPA as illustrated in Fig. 1.

The deep learning model utilizes supervised learning method to have a better prediction performance over linear prediction models. It in fact reduces prediction error. However, the proposed framework is intended to have improvement in the accuracy of prediction further. Therefore, the MM-HPA algorithm is defined to achieve even better results and reduce prediction error significantly. In fact, the MM-HPA is the combination of three prediction models such as deep learning model as RNN model, exponential smoothing and autoregressive moving average. The weights needed by the MM-HPA are to be optimized. Towards this end, a genetic algorithm (GA) is used. The MM-HPA is realized as follows. It considers number of prediction models denoted as s. In this case, s value is 3 as there are three prediction models used for MM-HPA. The actual stock value is denoted as r_t $(t = 1, \ldots, T)$ at given time t and the prediction value is denoted as $\hat{r}_{lt}$ $(l = 1, \ldots, s)$ by using given model l. Thus, the prediction error can be computed as $\varepsilon_{lt} = r_t - \hat{r}_{lt}$. The final prediction of the MM-HPA is obtained as in Eq. 9.

$$\hat{r}_t = \sum_{i=1}^{s} w_i \hat{r}_{it} \quad (t = 1, \ldots T) \tag{9}$$

A condition is to be considered which ensures sum of weights to be 1. It is as in Eq. 10 and the prediction error of MM-HPA is computed as in Eq. 11

$$\sum_{i=1}^{s} \omega_i = 1 \tag{10}$$

$$\varepsilon_t = r_t - \hat{r}_t \tag{11}$$

The final prediction of MM-HPA is based on the combination of the three prediction models aforementioned. At a given time t, the prediction of MM-HPA is denoted as $\hat{r}_{\text{hybrid}(t)}$. The time t is expressed as $(t = 1, \ldots, T)$. The predictions obtained from the exponential smoothing are expressed as $\omega_3 \hat{r}_{\text{ES}(t)}$. Similarly, the results of the autoregressive MA are denoted as $\omega_2 \hat{r}_{\text{arMA}(t)}$ and the prediction results of the deep learning model are denoted as $\omega_1 \hat{r}_{\text{RNN}(t)}$. The resultant prediction is expressed as in Eq. 12.

$$\hat{r}_{\text{hybrid}(t)} = \omega_1 \hat{r}_{\text{RNN}(t)} + \omega_2 \hat{r}_{\text{arMA}(t)} + \omega_3 \hat{r}_{\text{ES}(t)} \quad (t = 1, \ldots T) \tag{12}$$

For better prediction performance, MM-HPA needed optimized weights computed with the help of GA. This optimization is made with Eqs. 13–15

$$\text{Min} \frac{\sum_{t=1}^{T} \left(r_t - \hat{r}_{\text{hybrid}(t)} \right)^2}{T} \tag{13}$$

$$\omega_1 + \omega_2 + \omega_3 = 1 \tag{14}$$

$$0 \le \omega_i \le 1 \quad (i = 1, \ldots 3) \tag{15}$$

As shown in Eq. 15, an objective function is employed to minimize MSE between actual and predicted stock returns. The weights range is between 0 and 1 as in Eq. 15 while Eq. 14 ensures the combined weights should be equal to 1. The GA uses different parameters to achieve this. They include number of generations for each stock as 100, mutation probability is 0.010, crossover probability is 0.80 and population size considered for every stock is 20.

4.2 Multi-model-Based Hybrid Prediction Algorithm

The proposed algorithm known as MM-HPA takes time series stock market data as input and produces effective predictions of stock returns.

Algorithm 3: Multi-Model based Hybrid Prediction Algorithm

Input: Stock market dataset denoted as r_t(t= 1, ……., T)

Output: Predicted stock returns denoted as $\widehat{r_t}$(t= 1, ……., T)

Step 1: Initialize E to 2 (number of linear prediction models)
Step 2: Initialize F to 1 (number of non-liner prediction models)
Step 3: For each prediction mode e in E
Step 4: Find $\hat{r}_{t,L_i}$ denoted as linear predictions
Step 5: For each prediction model f in F
Step 6: Find $\hat{r}_{t,\,NL_j}$ denoted as non-linear predictions
Step 7: End For
Step 8: End For
Step 9: Compute weights for linear models
Step 10: Compute weights for non-linear models

$$\text{Min } \frac{\sum_{t=1}^{T}(r_t-\widehat{r_t})^2}{T}$$

Such that total weights lead to 1 and

$0 \le W_{Lit} \le 1, 1 \le i \le E$

$0 \le W_{NLjt} \le 1, 1 \le j \le F$

Step 11: Finally compute final predictions as

$$\widehat{r_t} = \sum_{i=1}^{E} W_{Lit}\,\hat{r}_{t,L_i} + \sum_{j=1}^{F} W_{NLjt}\,\hat{r}_{t,\,NL_j}$$

Step 12: Return $\widehat{r_t}$

As presented in Algorithm 1, the models described in Sect. 3 are used as part of MM-HPA. Step 1 and Step 2 of the algorithm initialises required number of linear and nonlinear prediction models, respectively. Though the algorithm can support multiple models, in this paper two linear and one nonlinear prediction models are used. Accordingly, the E and F are initialized. Step 3 starts an iterative process to find linear predictions with the two models while Step 5 stars an iterative process to find nonlinear predictions for the RNN model that is deep learning related. Step 4 and Step 6 find linear and nonlinear predictions, respectively. Step 7 and Step 8 end the iterative process for E and F prediction models. Step 9 computes weights for linear models while Step 10 computes weights for nonlinear models. Here, optimal weights are obtained with GA-based solution. In other words, the MM-HPA takes help of GA for optimal weights. The two optimal weight computations are based on the given rules provided in Step 10. Step 11 makes the final predictions that are much better than individual models.

4.3 Evaluation Metrics

In order to evaluate the performance of the proposed MM-HPA model and the linear and nonlinear models that are used in the process, two important metrics are employed. They are known as mean absolute error (MAE) and mean squared error (MSE).

$$\text{MAE} = \frac{1}{n}\sum |y - \hat{y}| \tag{16}$$

$$\text{MSE} = \frac{1}{n}\sum (y - \hat{y})^2 \tag{17}$$

As in Eqs. 16 and 17, the standard metrics are used in this paper to find the prediction error exhibited by individual models and the proposed hybrid algorithm.

Table 1 Prediction error of RNN model (daily predictions)

	Train (daily)		Test (daily)		Correlation
	MAE	MSE	MAE	MSE	
TCS	0.006132852360938809	0.00019333867640809678	0.005567248560374257	0.00013314948556937954	0.9913504028147111
Wipro	0.005854871453872572	0.00019737677588896725	0.004876163187366192	0.00011351931569646815	0.9924095599431554
Maruti	0.006076769393292211	0.00019610843784383697	0.005037874364154098	7.655193281512817e−05	0.9947702466947094
Tata Steel	0.007980804868219863	0.00021820722750340966	0.007152796277472974	0.00014373997313038683	0.9972783332994127
BHEL	0.005263989458658865	0.00019688465947962998	0.004028101231428839	5.3720707855391226e−05	0.9961921938856078
Axis Bank	0.006196936553654571	0.0001981685359385715	0.00495936956133689	6.351643894239576e−05	0.9980960864326753

Table 2 Prediction error of RNN model (weekly predictions)

	Train (daily)		Test (weekly)		Correlation
	MAE	MSE	MAE	MSE	
TCS	0.006559065520940792	0.0001970794945962206	0.005818003207275415	0.00014352765409500674	0.9903505710097423
Wipro	0.006556894680085991	0.0002014478407848704	0.004464857808782151	4.4301068848308485e−05	0.9970332750456801
Maruti	0.005828547923426543	0.00019653785481447508	0.006305769147860213	0.00019519374788183418	0.9868232909702479
Tata Steel	0.005434738807491071	0.00019577300907344796	0.006635449188002854	0.000591733862989719	0.987296248896416
BHEL	0.005237665800710342	0.00019653823385721938	0.004382733503163973	8.245120420237617e−05	0.9941643640406931
Axis Bank	0.006809387062232764	0.00020776487694785741	0.005790383994412815	0.00013549266741772538	0.9956350839286132

Table 3 Prediction error comparison between Linear and MM-HP Model (daily returns)

	Linear (daily)			MM-HP model (daily)		
	MAE	MSE	Correlation	MAE	MSE	Correlation
TCS	0.040545	0.002796	0.122036	0.003031	0.0000297	0.996368
BHEL	0.049675	0.004239	0.18333	0.002433	0.0000266	0.996113
Wipro	0.041344	0.00318	0.159626	0.001904	0.0000104	0.998837
Axis Bank	0.060081	0.005741	0.222467	0.00293	0.000042	0.997305
Maruti	0.045817	0.003567	0.174413	0.003118	0.0000563	0.991858
Tata Steel	0.064424	0.007261	0.270297	0.002758	0.0000498	0.997769

4.4 Experiment Setup

Experiments on the proposed MM-HPA are made with a prototype application. For empirical study, six stock datasets are collected from National Stock Exchange (NSE) of India website http://www.nseindia.com. The datasets are related to different tickers such as Tatal Steel, Maruti, Wipro, Axis Bank, TCS and BHEL. Python data science platform (Anaconda with Spyder IDE) is used for application development. The application is executed in a Laptop with Intel Core i5-4210U CPU with 4 GB RAM and 1.70 GHz processor running Windows 10 64-bit operating system. Experiments are made with stock prediction for both daily and weekly returns.

5 Experimental Results

Both linear and nonlinear models are used for empirical study. Then, the hybrid approach is employed for improving prediction performance. The nonlinear model (deep learning with RNN) is used with 50% data for training and 50% for testing. The learning rate of RNN is 0.5 and the 0.3 is set as momentum while error rate threshold is used as 0.0002. For each ticker, RNN is trained and used different number of epochs in order to reach the error threshold set. The results are presented in the Results outcomes sections.

6 Conclusion and Future Work

In this paper, a framework known as SMPPF is proposed for effective stock returns prediction. The SMPPF is realized with a hybrid prediction algorithm based on linear and nonlinear prediction models. The algorithm is known as MM-HPA. Linear prediction models such as autoregressive MA and exponential smoothing and a deep learning model are used for realizing the hybrid algorithm. Six companies' stock datasets are collected from NSE and used for empirical study. The linear models showed higher prediction error while the deep learning model showed less prediction error. In other words, the nonlinear model showed better performance over the linear counterparts. Even though deep learning model performed better than other models, a hybrid prediction algorithm is proposed to combine the features of them along with GA for weights optimization. The rationale for GA as an evolutionary algorithm is that there are multiple prediction models involved in the prediction framework and there is need for optimizing their weights. The empirical study revealed that the MM-HPA showed better prediction performance over linear and nonlinear models. Moreover, its prediction error is significantly reduced. Therefore, the MM-HPA is found to be the highly accurate prediction model as it can ascertain nonlinear trends in data for effective prediction of stock returns. In future, we intent to improve the prediction performance of the proposed MM-HPA with Pareto Optimization and Greedy Heuristic Methods and to minimize its computation time.

Table 4 Prediction error comparison between Linear and MM-HP model (weekly returns)

	Linear (weekly)			MM-HP model (weekly)		
	MAE	MSE	Correlation	MAE	MSE	Correlation
TCS	0.040545	0.002796	0.122036	0.002634	0.0000349	0.995742
BHEL	0.049675	0.004239	0.18333	0.002671	0.0000229	0.996741
Wipro	0.041344	0.00318	0.159626	0.002397	0.000024	0.997037
Axis Bank	0.060081	0.005741	0.222467	0.002804	0.0000302	0.998247
Maruti	0.045817	0.003567	0.174413	0.002218	0.0000216	0.997091
Tata Steel	0.064424	0.007261	0.270297	0.004635	0.0000677	0.997238

Table 5 Prediction error comparison between RNN and MM-HP model (daily returns)

	RNN (daily)			MM-HP model (daily)		
	MAE	MSE	Correlation	MAE	MSE	Correlation
TCS	0.006132852360938809	0.00019333867640809678	0.9913504028147111	0.003031	0.0000297	0.996368
BHEL	0.005854871453872572	0.00019737677588896725	0.9924095599431554	0.002433	0.0000266	0.996113
Wipro	0.006076769393292211	0.00019610843784383697	0.9947702466947094	0.001904	0.0000104	0.998837
Axis Bank	0.007980804868219863	0.00021820722750340966	0.9972783332994127	0.00293	0.000042	0.997305
Maruti	0.005263989458658865	0.00019688465947962998	0.9961921938856078	0.003118	0.0000563	0.991858
Tata Steel	0.006196936553654571	0.00019816853593857515	0.9980960864326753	0.002758	0.0000498	0.997769

Table 6 Prediction error comparison between RNN and MM-HP Model (weekly returns)

	RNN (weekly)			MM-HP model (weekly)		
	MAE	MSE	Correlation	MAE	MSE	Correlation
TCS	0.006559065520940792	0.0001970794945962206	0.9903505710097423	0.002634	0.0000349	0.995742
BHEL	0.006556894680085991	0.0002014478407848704	0.9970332750456801	0.002671	0.0000229	0.996741
WIPRO	0.005828547923426543	0.00019653785481447508	0.9868232909702479	0.002397	0.000024	0.997037
Axis Bank	0.005434738807491071	0.00019577300907344796	0.987296248896416	0.002804	0.0000302	0.998247
Maruti	0.005237665800710342	0.00019653823385721938	0.9941643640406931	0.002218	0.0000216	0.997091
Tata Steel	0.006809387062232764	0.00020776487694785741	0.9956350839286132	0.004635	0.0000677	0.997238

7 Result Outcomes

7.1 Prediction Error Comparison Between Various Models

Prediction error is computed in terms of MAE and MSE metrics. Linear, nonlinear and **MM_HP Model** models are observed with these metrics. The results are provided in this sub section.

As presented in Table 1, the MAE and MSE values are presented for nonlinear or deep learning model which is based on RNN. The error rate is very low indicating high performance of the mode in predicting stock returns. The daily returns are predicted with least error rate on both train and test data. The correlation between predicted and target returns is very high. High correlation reflects better performance of the model. It does mean that RNN-based prediction model provides satisfactory performance in stock returns prediction (daily predictions).

As presented in Table 2, the MAE and MSE values are presented for nonlinear or deep learning model which is based on RNN. The error rate is very low indicating high performance of the mode in predicting stock returns. The weekly returns are predicted with least error rate on both train and test data. The correlation between predicted and target returns is very high. High correlation reflects better performance of the model. It does mean that RNN-based prediction model provides satisfactory performance in stock returns prediction (weekly predictions).

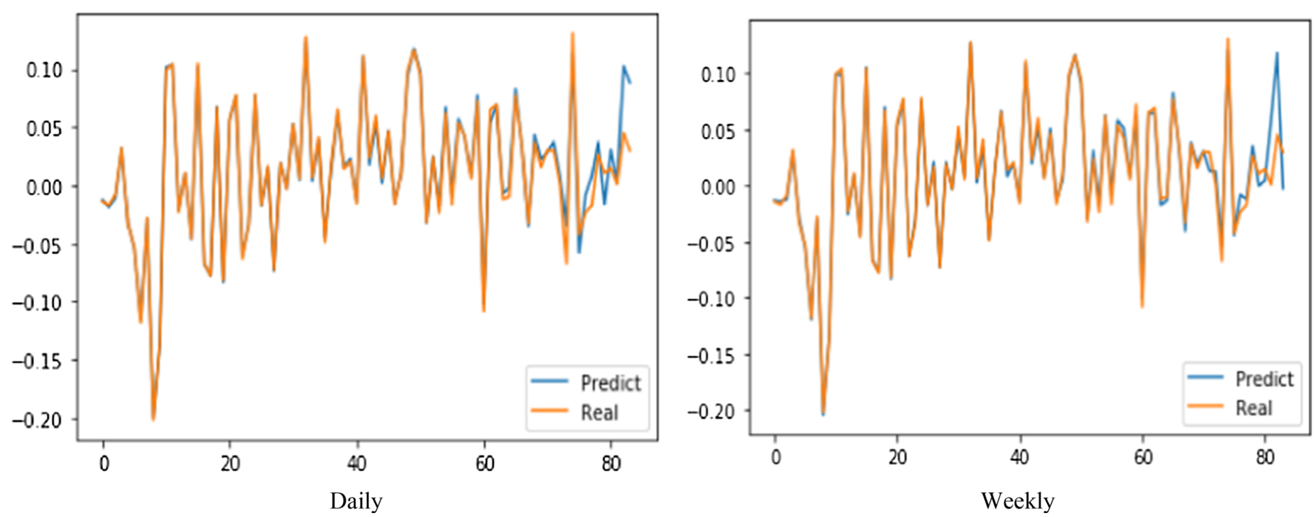


Fig. 2 RNN model prediction performance for TCS

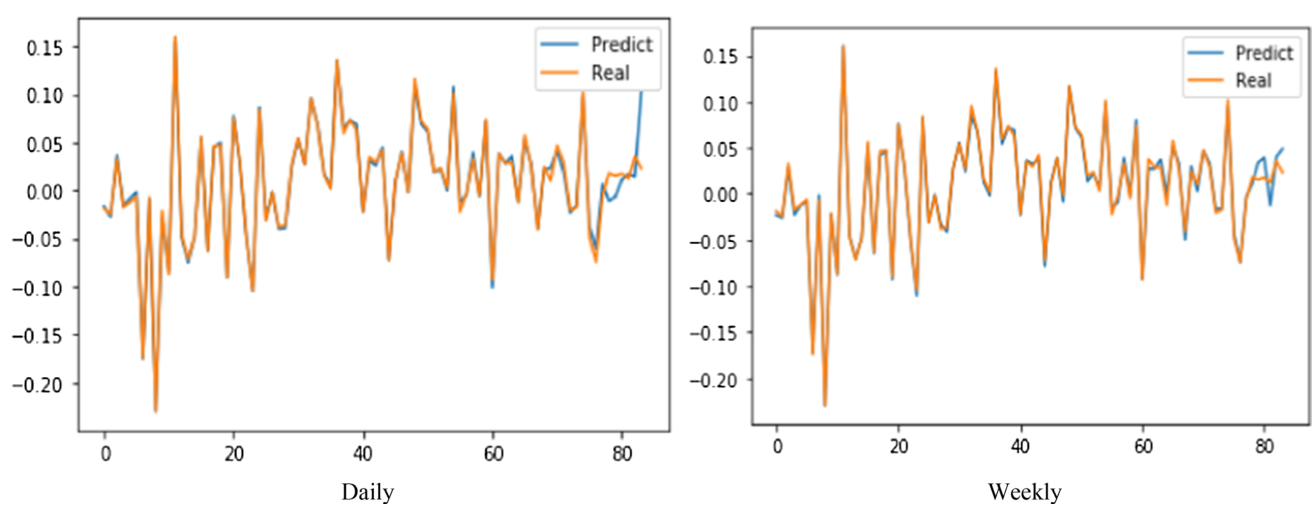


Fig. 3 RNN model prediction performance for Wipro

As presented in Table 3, there is performance comparison between linear and hybrid models as MM-HPA in terms of error rate in case of daily returns. It is observed from the results that the MM-HPA model outperforms both linear and RNN model (when compared with Table 2 also). The error rate is very low in terms of MSE and MAE besides achieving very high correlation values. For instance, with linear model the MAE value for ticker TCS is 0.040545 which is very higher than the MM_HP model for same ticker. For TCS, the MM-HP model shows 0.003031 which is the least error rate found. In the same fashion, for TCS, the correlation in case of linear model is very low that is 0.122036 indicating least performance. In case of MM-HP model for TCS, the correlation value is very high that is 0.996368 indicating highest performance. It is better than the correlation observed for RNN model in Table 2 as well. Therefore, it is concluded that MM-HP model showed better performance than linear and nonlinear models in case of daily returns.

As presented in Table 4, there is performance comparison between linear and hybrid models as MM-HP model in terms of error rate in case of weekly returns. It is observed from the results that the hybrid model outperforms both linear and RNN models (when compared with Table 3 also). The error rate is very low in terms of MSE and MAE besides achieving very high correlation values. For instance, with linear model the MAE value for ticker TCS is 0.040545 which is very higher than MM-HP model for same ticker. For TCS, the MM-HP model shows 0.002634 which is the least error rate found. In the same fashion, for TCS, the correlation in case of linear model is very low that is 0.122036 indicating least performance. In case of MM-HP model for TCS, the correlation value is very high that is 0.995742 indicating highest performance. It is better than the correlation observed for RNN model in Table 2 as well. Therefore, it is concluded that the MM-HP model showed better

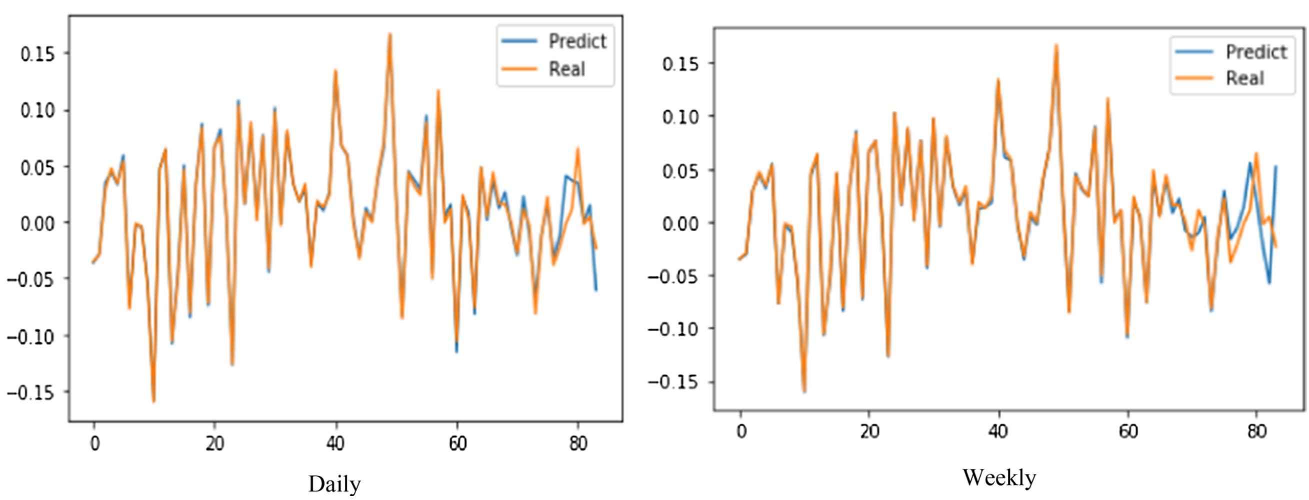


Fig. 4 RNN model prediction performance for Maruti

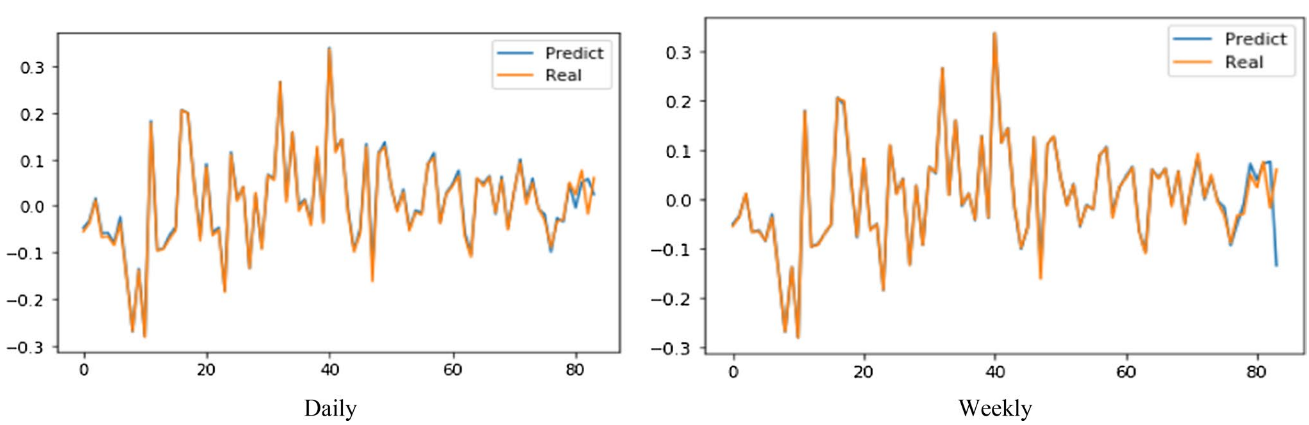


Fig. 5 RNN model prediction performance for Tata Steel

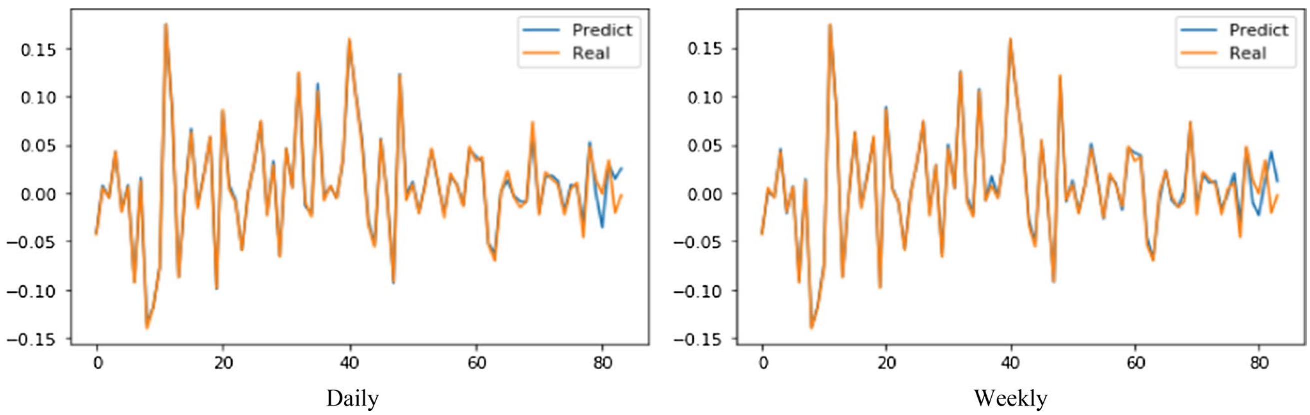


Fig. 6 RNN model prediction performance for BHEL

performance than linear and nonlinear models in case of daily returns.

As presented in Table 5, there is performance comparison between RNN model and hybrid models as MM-HPA model in terms of error rate in case of daily returns. It is observed from the results that the MM-HPA model outperforms RNN model (when compared with Table 1 also). The error rate is very low in terms of MSE and MAE besides achieving very high correlation values. For instance, with RNN model the MAE value for ticker TCS is 0.006132852360938809 which is very higher than the MM-HP model for same ticker. For TCS, the MM-HP model shows 0.003031 which is the least error rate found. In the same fashion, for TCS, the correlation in case of RNN model is very low that is 0.9913504028147111 indicating least performance. In case of MM-HP model for TCS, the correlation value is very high that is 0.996368 indicating highest performance. It is better than the correlation observed for RNN model in Table 1 as well. Therefore, it is concluded that MM-HP model showed better performance than linear and nonlinear models (RNN) in case of daily returns.

As presented in Table 6, there is performance comparison between RNN model and hybrid models as MM-HPA model in terms of error rate in case of daily returns. It is observed from the results that the MM-HPA model outperforms RNN model (when compared with Table 2 also). The error rate is very low in terms of MSE and MAE besides achieving very high correlation values. For instance, with RNN model the MAE value for ticker TCS is 0.006559065520940792 which is very higher than the MM-HP model for same ticker. For TCS, the MM-HP model shows 0.002634 which is the least error rate found. In the same fashion, for TCS, the correlation in case of RNN model is very low that is 0.9903505710097423 indicating least performance. In case of MM-HP model for TCS, the correlation value is very high that is 0.995742 indicating highest performance. It is better than the correlation observed for RNN model in Table 2 as well. Therefore, it is concluded that MM-HP model showed better performance than linear and nonlinear models (RNN) in case of Weekly returns.

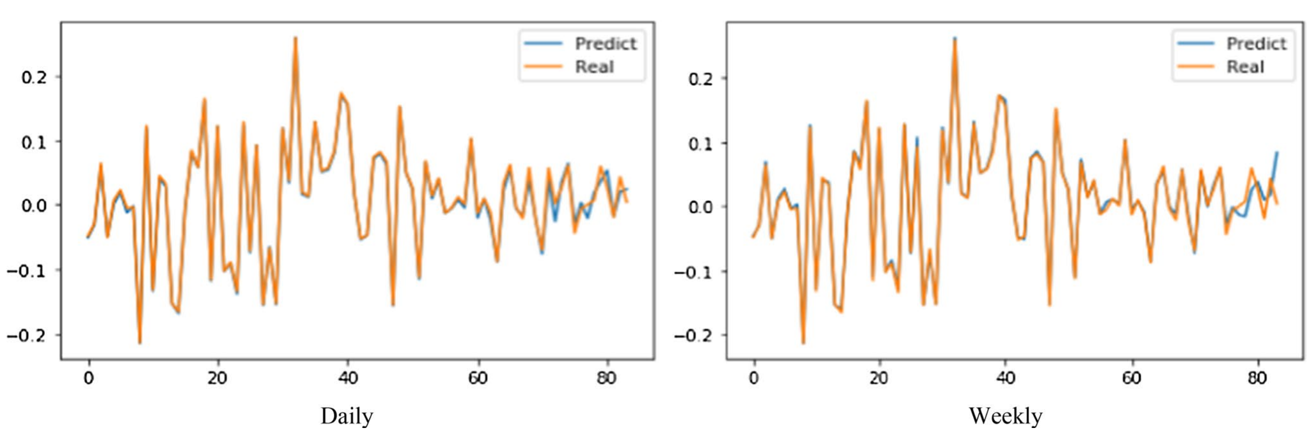


Fig. 7 RNN model prediction performance for Axis Bank

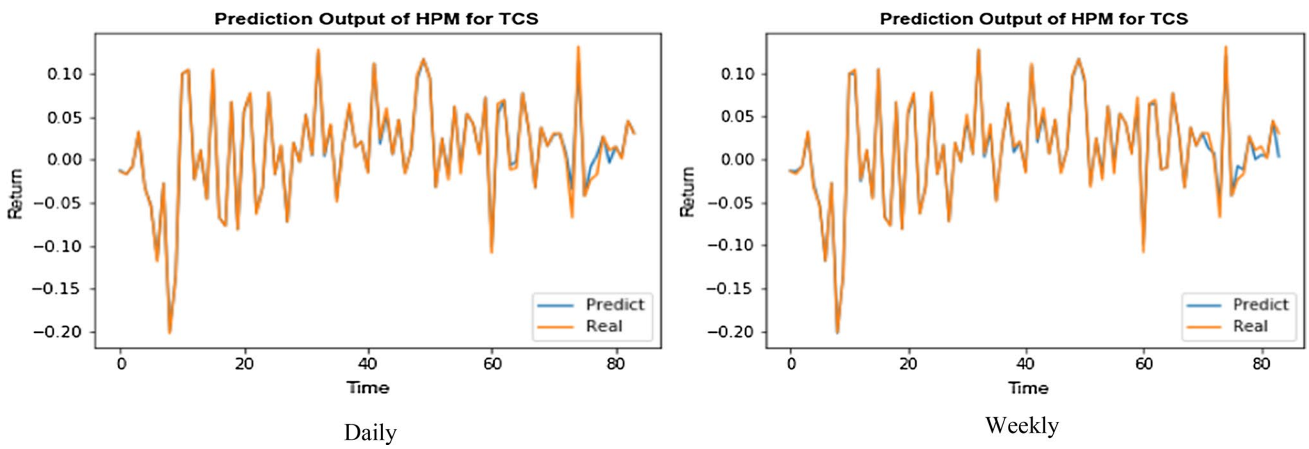


Fig. 8 Comparison daily and weekly prediction performance of MM-HPA for TCS

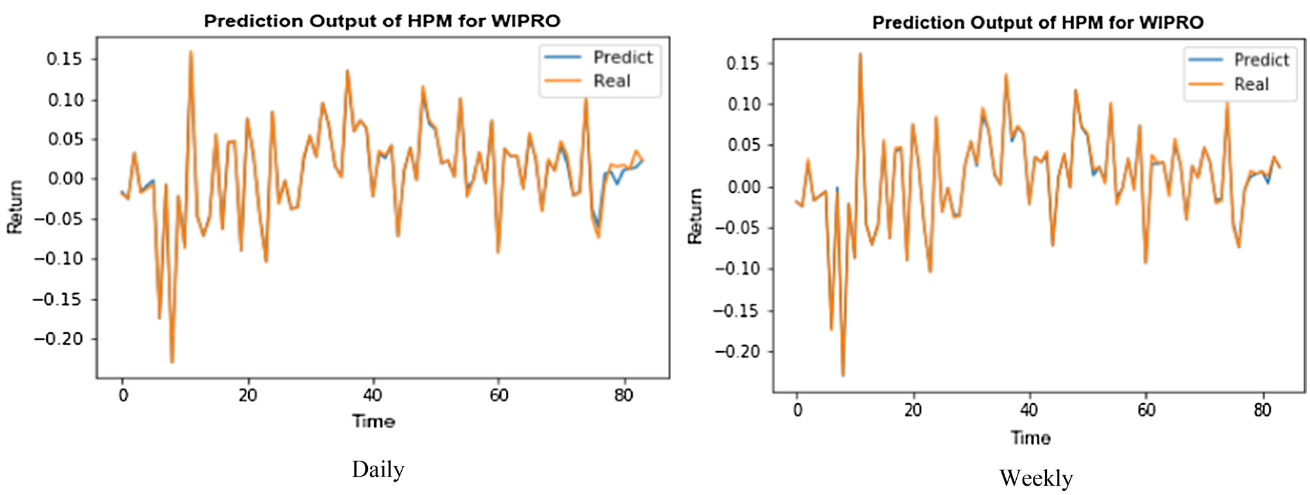


Fig. 9 Comparison daily and weekly prediction performance of MM-HPA for Wipro

7.2 Results of Deep Learning as Recurrent Neural Networks for Stock Prediction

Stock prediction performance is presented in this section for deep learning model made up of RNN. It is observed for weekly and daily returns prediction. The observations are made for six stock tickers.

As presented in Fig. 2, the daily stock returns prediction results (left side) and weekly stock returns prediction results (right side) are provided. The horizontal axis shows time, and vertical axis shows return value. Two time-series data series are provided, namely real data series and predicted data series. There is visible difference between real and predicted stock returns on daily and weekly basis as both data series are shown in different colour. From the results it is understood that the prediction performance, when RNN model is used, is high as the correlation is high and error rate is low as discussed earlier. These observations are related to the ticker TCS.

As presented in Fig. 3, the daily stock returns prediction results (left side) and weekly stock returns prediction results (right side) are provided. The horizontal axis shows time, and vertical axis shows return value. Two time-series data series are provided, namely real data series and predicted data series. There is visible difference between real and predicted stock returns on daily and weekly basis as both data series are shown in different colour. From the results it is understood that the prediction performance, when RNN model is used, is high as the correlation is high and error rate is low as discussed earlier. These observations are related to the ticker Wipro.

As presented in Fig. 4, the daily stock returns prediction results (left side) and weekly stock returns prediction results (right side) are provided. The horizontal axis shows time, and vertical axis shows return value. Two time-series

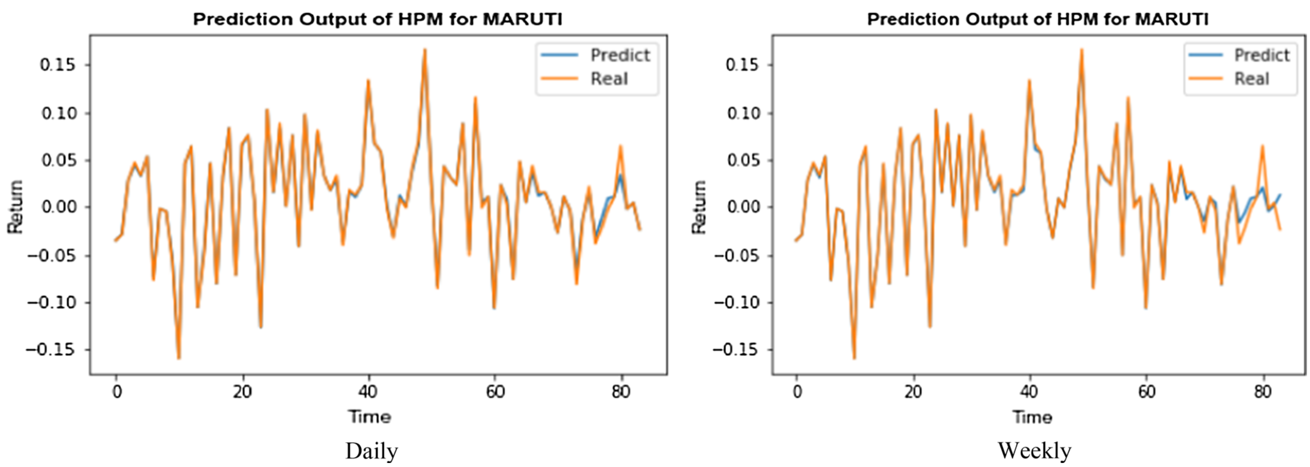


Fig. 10 Comparison daily and weekly prediction performance of MM-HPA for Maruti

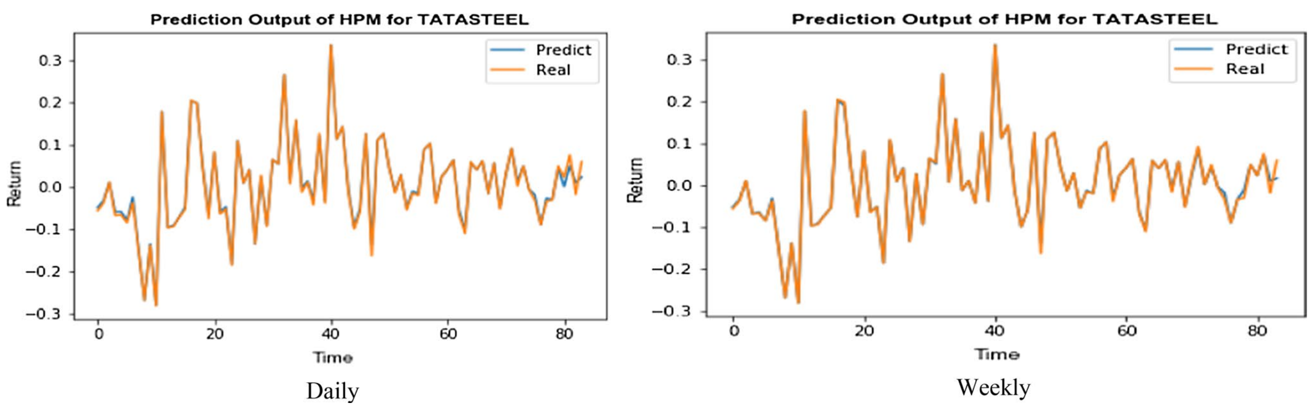


Fig. 11 Comparison daily and weekly prediction performance of MM-HPA for Tata Steel

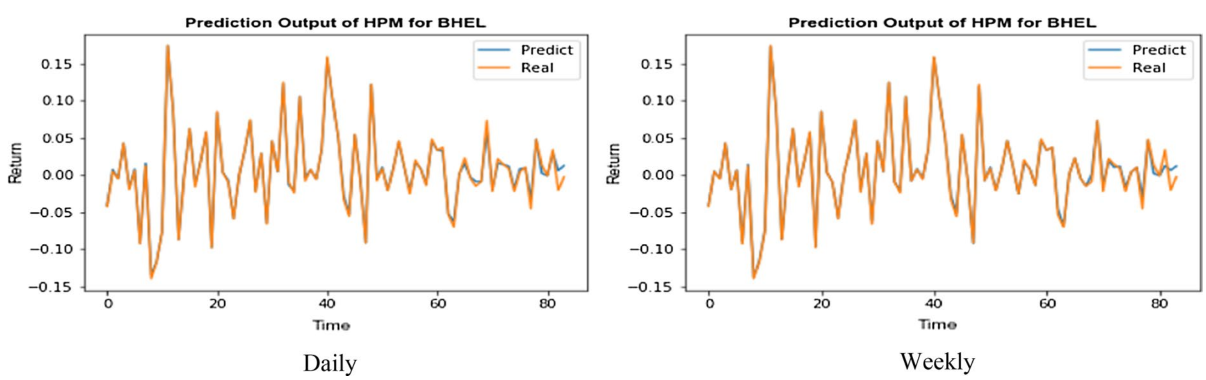
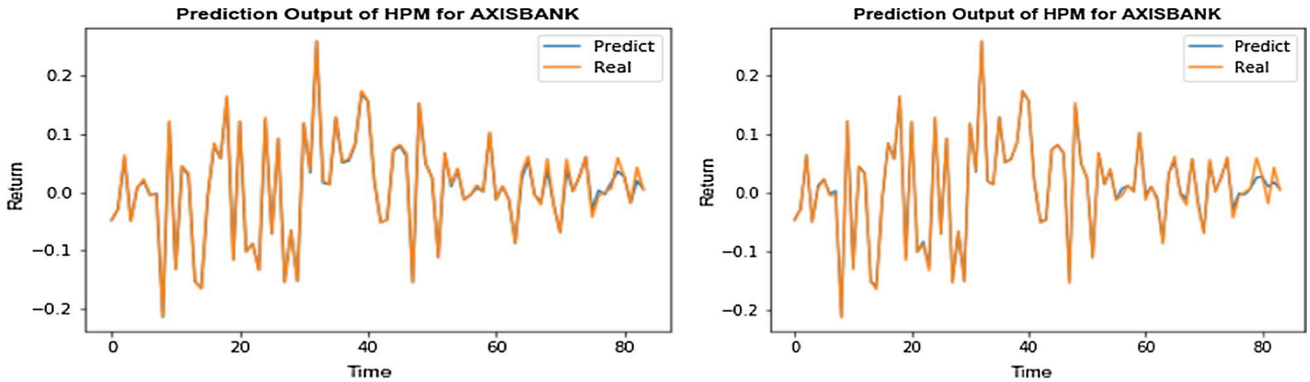


Fig. 12 Comparison daily and weekly prediction performance of MM-HPA for BHEL

data series are provided, namely real data series and predicted data series. There is visible difference between real and predicted stock returns on daily and weekly basis as both data series are shown in different colour. From the results it is understood that the prediction performance, when RNN model is used, is high as the correlation is high and error rate is low as discussed earlier. These observations are related to the ticker Maruti.



Daily | Weekly

Fig. 13 Comparison daily and weekly prediction performance of MM-HPA for Axis Bank

As presented in Fig. 5, the daily stock returns prediction results (left side) and weekly stock returns prediction results (right side) are provided. The horizontal axis shows time, and vertical axis shows return value. Two time-series data series are provided, namely real data series and predicted data series. There is visible difference between real and predicted stock returns on daily and weekly basis as both data series are shown in different colour. From the results it is understood that the prediction performance, when RNN model is used, is high as the correlation is high and error rate is low as discussed earlier. These observations are related to the ticker Tata Steel.

As presented in Fig. 6, the daily stock returns prediction results (left side) and weekly stock returns prediction results (right side) are provided. The horizontal axis shows time, and vertical axis shows return value. Two time-series data series are provided, namely real data series and predicted data series. There is visible difference between real and predicted stock returns on daily and weekly basis as both data series are shown in different colour. From the results it is understood that the prediction performance, when RNN model is used, is high as the correlation is high and error rate is low as discussed earlier. These observations are related to the ticker BHEL.

As presented in Fig. 7, the daily stock returns prediction results (left side) and weekly stock returns prediction results (right side) are provided. The horizontal axis shows time, and vertical axis shows return value. Two time-series data series are provided, namely real data series and predicted data series. There is visible difference between real and predicted stock returns on daily and weekly basis as both data series are shown in different colour. From the results it is understood that the prediction performance, when RNN model is used, is high as the correlation is high and error rate is low as discussed earlier. These observations are related to the ticker Axis Bank.

Table 7 Acronyms

Abbreviation	Description
ANNs	Artificial Neural Networks
CNN	Convolutional Neural Network
DBN	Deep Belief Network
EMH	Efficient Markets Hypothesis
ETFs	Exchange-Traded Funds
GA	Generic Algorithm
LG	Logistic Regression
LSTM	Long Short-Term Memory
MA	Moving Average
ML	Machine Learning
MLP	Multilayer Perceptron
MM-HPA	Multi-model-Based Hybrid Prediction Algorithm
MSA	Mean Absolute Error
MSE	Mean Squared Error
NN	Neural Networks
NSE	National Stock Exchange
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
RF	Random Forest
RNN	Recurring Neural Network
RSI	Relative Strength Index
SMPPF	Stock Market Prices Prediction Framework

7.3 Results for MM-HPA for Stock Prediction

This section presents results of daily and weekly stock predictions using the proposed hybrid prediction model that exploits both linear and nonlinear models. The model is known as MM-HPA.

As presented in Fig. 8, the stock returns prediction results on daily (left) and weekly (right) basis are visualized. Both real stock returns and predicted returns are plotted in the

Table 8 Notations

Notation	Description
$\hat{r}_t$	Predicted value
ε_t	Random errors
K	Reference location
P	Order of regression
Q	Order of moving average model
R	Pass returns
$\mathfrak{R}$	Autoregressive predictor
r_t	Refers to actual value
T	Number of historical time series in the dataset
t	Denotes time
Z	Moving reference expression
φ	Coefficient in moving average model
ω	Coefficient in regressive model

graph. The horizontal axis shows time, and vertical axis shows return value. The results are obtained for the ticker TCS. The results revealed that the proposed hybrid model named MM-HPA outperforms the deep learning model made up of RNN. It is evident in the stock returns as well as the error rate details presented earlier. The rationale behind this is that the hybrid model gets the benefits of linear and nonlinear models. From the results it is observed that MM-HPA outperforms the other models.

As presented in Fig. 9, the stock returns prediction results on daily (left) and weekly (right) basis are visualized. Both real stock returns and predicted returns are plotted in the graph. The horizontal axis shows time, and vertical axis shows return value. The results are obtained for the ticker Wipro. The results revealed that the proposed hybrid model named MM-HPA outperforms the deep learning model made up of RNN. It is evident in the stock returns as well as the error rate details presented earlier. The rationale behind this is that the hybrid model gets the benefits of linear and nonlinear models. From the results it is observed that MM-HPA outperforms the other models.

As presented in Fig. 10, the stock returns prediction results on daily (left) and weekly (right) basis are visualized. Both real stock returns and predicted returns are plotted in the graph. The horizontal axis shows time, and vertical axis shows return value. The results are obtained for the ticker Maruti. The results revealed that the proposed hybrid model named MM-HPA outperforms the deep learning model made up of RNN. It is evident in the stock returns as well as the error rate details presented earlier. The rationale behind this is that the hybrid model gets the benefits of linear and nonlinear models. From the results it is observed that MM-HPA outperforms the other models.

As presented in Fig. 11, the stock returns prediction results on daily (left) and weekly (right) basis are visualized. Both real stock returns and predicted returns are plotted in the graph. The horizontal axis shows time, and vertical axis shows return value. The results are obtained for the ticker Tata Steel. The results revealed that the proposed hybrid model named MM-HPA outperforms the deep learning model made up of RNN. It is evident in the stock returns as well as the error rate details presented earlier. The rationale behind this is that the hybrid model gets the benefits of linear and nonlinear models. From the results it is observed that MM-HPA outperforms the other models.

As presented in Fig. 12, the stock returns prediction results on daily (left) and weekly (right) basis are visualized. Both real stock returns and predicted returns are plotted in the graph. The horizontal axis shows time, and vertical axis shows return value. The results are obtained for the ticker BHEL. The results revealed that the proposed hybrid model named MM-HPA outperforms the deep learning model made up of RNN. It is evident in the stock returns as well as the error rate details presented earlier. The rationale behind this is that the hybrid model gets the benefits of linear and nonlinear models. From the results it is observed that MM-HPA outperforms the other models.

As presented in Fig. 13, the stock returns prediction results on daily (left) and weekly (right) basis are visualized. Both real stock returns and predicted returns are plotted in the graph. The horizontal axis shows time, and vertical axis shows return value. The results are obtained for the ticker Axis Bank. The results revealed that the proposed hybrid model named MM-HPA outperforms the deep learning model made up of RNN. It is evident in the stock returns as well as the error rate details presented earlier. The rationale behind this is that the hybrid model gets the benefits of linear and nonlinear models. From the results it is observed that MM-HPA outperforms the other models. From the empirical study it is evident that the MM-HPA which is the proposed hybrid model that outperforms linear and nonlinear (deep learning) models in terms of stock returns (weekly and daily), error rate (MAE and MSE) and correlation analysis (Tables 7, 8).

References

1. dos Santos Pinheiro, L.; Dras, M.: Stock market prediction with deep learning a character-based neural language model for event-based trading, no. 5, pp. 1–10 (2017)
2. Yang, J.; Li, Y.; Chen, X.; Cao, J.; Jiang, K.: Deep learning for stock selection based on high frequency price volume data. Computational Finance (q-fin.CP); Machine Learning (cs.LG), vol. 6(1), pp. 1–25. arXiv:1911.02502 [q-fin.CP] (2019)

3. Oncharoen, P.; Vateekul, P.: Deep learning for stock market prediction using event embedding and technical indicators. In: 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), pp. 1–6 (2018)
4. Lee, J., Kim, R., Koh, Y., Kang, J.: Global stock market prediction based on stock chart images using deep Q-network. IEEE Access **7**, 167260–167277 (2019)
5. Jain, S.; Gupta, R.; Moghe, A.A.: Stock price prediction on daily stock data using deep neural networks, vol. 5(7), pp. 46–54 (2018)
6. Gudelek, M.U., Boluk, S.A., Ozbayoglu, A.M.: A deep learning based stock trading model with 2-D CNN trend detection. In: 2017 IEEE Symposium Series on Computational Intelligence, pp. 1–8 (2017)
7. Fister, D.; Mun, J.C.; Jagri, V.; Jagri, T.: Deep learning for stock market trading a superior trading strategy LSTM networks, machine learning, automated stock trading. Neural. Netw. World. (2019). https://doi.org/10.14311/NNW.2019.29.011
8. Kim, T., Kim, H.Y.: Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. PLoS One. **14**(2), e0212320 (2019). https://doi.org/10.1371/journal.pone.0212320
9. Feng, F.; Chen, H.; He, X.; Ding, J.; Sun, M.; Chua, T.S. (2019). Enhancing stock movement prediction with adversarial training. In: International Joint Conferences on Artificial Intelligence Organization, vol. 1(2), pp. 1–7
10. Pang, X., Zhou, Y., Wang, P., Lin, W., Chang, V.: An innovative neural network approach for stock market prediction. J. Supercomput. **76**(3), 2098–2118 (2018)
11. Fischer, T.; Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. Eur. J. Oper. Res. **12**(1), 1–16 (2018)
12. Chen, C.; Zhao, L.; Bian, J.; Xing, C.; Liu, T.Y.: Investment behaviours can tell what inside: exploring stock intrinsic properties for stock trend prediction. In: Applied Data Science Track Paper, KDD'19, Anchorage, AK, vol. 4(8), pp. 1–9, 4–8 August (2019).
13. Lahmiri, S.: A technical analysis information fusion approach for stock price analysis and modeling. Fluct. Noise Lett. **17**(01), 1–14 (2018)
14. Xiuli, T.; Chai, Y.; Tang, Y.: Deep learning with stock indicators and two dimensional principal component analysis for closing price prediction system, vol. 1(1), pp. 1–4. IEEE (2016)
15. Jitpakdee, P.; Pravithana, N.: A novel method of trend lines generation using hough transform method. Int. J. Comput. Acad. Res. (IJCAR) **6**(4), 125–135 (2017)
16. Batres-Estrada, G.: Deep learning for multivariate financial time series. Math. Stat. Probab. Theory Stat. **4**(1), 1–94 (2015)
17. Chong, E.; Han, C.; Park, F.: Deep learning networks for stock market analysis and prediction methodology, data representations and case studies. Multilayer Neural Netw. **15**, 187–205 (2017)
18. Maqsood, H., Mehmood, I., Maqsood, M., Yasir, M., Afzal, S., Aadil, F., Selim, M.M., Muhammad, K.: A local and global event sentiment based efficient stock exchange forecasting using deep learning. Int. J. Inf. Manag. **50**, 432–451 (2019)
19. Zhong, X.; Enke, D.: Predicting the daily return direction of the stock market using hybrid machine learning algorithms. Financ. Innov. **5**(1), 1–20 (2019)
20. Shengnan, Y.U.: Financial time series analysis of stock data, vol. 1(16), pp. 1–20 (2018). https://www.cse.ust.hk/~rossiter/independent_studies_projects/lstm_stock_prediction/lstm_stock_prediction.pdf
21. Rao, P.S.; Srinivas, K.; Mohan, A.K.: A survey on stock market prediction using machine learning techniques. In: ICDSMLA (2019)
22. Li, W.; Liao, J.: A comparative study on trend forecasting approach for stock price time series. In: 2017 11th IEEE International Conference on Anti-Counterfeiting, Security, and Identification, pp. 1–5 (2017)
23. Rechenthin, M.: Machine-learning classification techniques for the analysis and prediction of high-frequency stock direction. Lowa Research Online, pp. 1–292 (2014)
24. Rao, P.S., Srinivas, K., Mohan, A.K.: Stock market prices prediction using random forest and extra tree regression. Int. J. Recent Technol. Eng. (IJRTE) **8**(3), 1224–1228 (2019)
25. Tsantekidis, A.; Passalis, N.; Tefas, A.; Kanniainen, J.; Gabbouj, M.; Iosifidis, A.: Using deep learning for price prediction by exploiting stationary limit order book features, pp. 1–19 (2020)
26. Fang, F.; Chung, W.; Ventre, C.; Basios, M.; Kanthan, L.; Li, L.; Wu, F.: Ascertaining price formation in cryptocurrency markets with deep learning. Comput. Sci. Econ. ArXiv arXiv:2003.00803 (2020)