Journal of King Saud University – Computer and Information Sciences 34 (2022) 7433–7444

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Multi-Model Generative Adversarial Network Hybrid Prediction Algorithm (MMGAN-HPA) for stock market prices prediction

Subba Rao Polamuri [a,*], Dr. Kudipudi Srinivas [b], Dr. A. Krishna Mohan [c]

[a] Research Scholar, Department of CSE, JNTUK, Kakinada, AP, India
[b] Professor, Dept of CSE, V. R Siddhartha Engineering College, Vijayawada, AP, India
[c] Professor, Dept. of CSE, UCEK, JNTUK, Kakinada, AP, India

ARTICLE INFO

Article history:
Received 5 April 2021
Revised 9 June 2021
Accepted 2 July 2021
Available online 7 July 2021

Keywords:
Deep learning
Generative Adversarial Network (GAN)
Recurrent Neural Network (RNN)
Convolutional Neural Network (CNN)
Stock market analysis

ABSTRACT

Deep learning has achieved greater success in optimizing solutions associated with Artificial Intelligence (AI). In the financial domain, it is widely used for stock market prediction, trade execution strategies and portfolio optimization. Stock market prediction is a very significant use case in this domain. Generative Adversarial Networks (GANs) with advanced AI models have gained significance of late. However, it is used in image-image-translation and other computer vision scenarios. GANs are not used much for stock market prediction due to its difficulty in setting the right set of hyperparameters. In this paper, overcome this problem with reinforcement learning and Bayesian optimization. A deep learning framework based on GAN, named Stock-GAN, is implemented with generator and discriminator. The former is realized with LSTM, a variant of Recurrent Neural Network (RNN), while the latter uses Convolutional Neural Network. An algorithm named Generative Adversarial Network based Hybrid Prediction Algorithm (GAN-HPA) is proposed. An empirical study revealed that Stock-GAN achieves promising performance in stock price prediction when compared with the state of the art model known as Multi-Model based Hybrid Prediction Algorithm (MM-HPA). Afterwards, MM-HPA and GAN-HPA combined to form yet another hybrid model known as MMGAN-HPA for improved performance over MM-HPA and GAN-HPA.

1. Introduction

Stock market behavior prediction is of interest to investors and all stock market stakeholders. Deep learning models are proved to be promising alternatives in stock price prediction research as they achieved great success (Zhang et al., 2019). Technical analysis of stock markets reveals trends in stock portfolios. Many classical algorithms such as Autoregressive Integrated Moving Average (ARIMA) (Nau, 2014) came into existence. However, they are linear models and suffer from performance issues as stock market data is of time-series in nature revealing temporal dimension. Linear models with traditional machine learning methods such as Support Vector Machine (SVM) are used for forecasting (Polamuri et al., 2019). Ensemble models as explored in (Asad, 2015) could provide better performance. However, deep learning models in Vargas et al. (2017), Akita et al. (2016), Mai et al. (2019), Liu et al. (2018), Eapen et al. (2019) and Katayama et al. (2019) outperformed traditional machine learning models. CNN is studied in Hoseinzade and Haratizadeh (2019), Zhang et al. (2019), Sezer and Ozbayoglu (2018), Tsantekidis et al. (2017) and CNN with LSTM combination is explored for stock price forecasting. LSTM is used in (Nelson et al., 2017) and (Bukhari et al., 2020) as a variant of RNN for overcoming problem of losing gradients.

In Chen et al. (2020) hybrid deep learning models are proposed and found to be more efficient. In Polamuri et al. (2020) both linear and non-linear models are combined to have better prediction performance. There are many insights from existing deep learning models that led to the research carried out in this paper. First, CNN is better for classification. Second, LSTM can better capture temporal data variations than RNN (Vargas et al., 2017). Third, deep learning models suffer from lack of pre-processing (or quality of data) unless there is an efficient NLP based approach towards feature selection (Bukhari et al., 2020). Fourth, GAN is found effective in learning internal representations of data using unsupervised

* Corresponding author.
E-mail addresses: psr.subbu546@gmail.com (S.R. Polamuri), vrdrks@gmail.com (D.K. Srinivas), krishna.ankala@gmail.com (D.A. Krishna Mohan).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.07.001
1319-1578/

approach besides generating new data that resembles real data (Zhang et al., 2019). Fifth, a hybrid of linear and non-linear models could provide better performance in stock prediction (Polamuri et al., 2020). Sixth, there is need for GAN based framework that exploits CNN, LSTM and novel pre-processing to improve the state of the art. Considering these observations, in this paper, we proposed a framework known as Stock-GAN based on deep learning models for better prediction of stock markets.

The framework uses LSTM as generator and CNN as discriminator. There is strong feature extraction procedure from stock data where linear models like ARIMA are applied. Novel pre-processing methodology is integrated as part of Stock-GAN. We investigated on right set of hyperparameters suitable for (G) and (D). In order to tune the hyperparameters dynamically, Stock-GAN incorporates reinforcement learning framework along with Bayesian optimization technique. The feature extraction process results in so many features and all of them may not contribute in stock prediction process. In order to filter them out (dimensionality reduction / feature selection), XGBoost is used to identify importance of features. Then PCA is applied and Eigen portfolios are created so as to reduce number of features generated by autoencoders. Section 4 delves more details on Stock-GAN. Our contributions in this paper are as follows.

1. A deep learning framework known as Stock-GAN is proposed for effectiveness in prediction of stock prices. Stock-GAN is realized with LSTM as generator and CNN as discriminator.
2. An algorithm named Generative Adversarial Network (GAN) based Hybrid Prediction Algorithm (GAN-HPA) is proposed with novel feature selection to realize and improve performance of Stock-GAN.
3. A prototype application is built using Python data science platform to evaluate the GAN-HPA and compared with same with another deep learning based hybrid algorithm named Multi-Model based Hybrid Prediction Algorithm (MM-HPA).
4. MM-HPA and GAN-HPA combined to form yet another hybrid model known as MMGAN-HPA for improved performance over MM-HPA and GAN-HPA.

The remainder of the paper is structured as follows. Section 2 reviews literature on advanced data science concepts for stock market prediction. Section 3 illustrates GAN architecture with its underlying mathematical model. Section 4 describes the proposed Stock-GAN architecture and its algorithm. Section 5 presents results of experiments with seven stock tickers collected from National Stock Exchange, India. Section 6 concludes the paper and provides directions for future scope of the research.

2. Related WORK

This section reviews literature on advanced data science concepts for stock market prediction. Vargas et al. (2017) studied both CNN and RNN architectures for prediction of stock markets based on data of financial news. They found that RNN is better for ascertaining temporal characteristics while CNN is better to obtain semantics from text. They intended to use reinforcement learning in future for better performance. Hiransha et al. (2018) investigated four deep learning models such as CNN, LSTM, RNN and MLP (Multilayer Perceptron) for prediction of stock prices. They found that deep learning models are better than non-linear models like ARIMA. Akita et al. (2016) used LSTM along with the concept of paragraph vector generated from news articles from stock price prediction. Chen et al. (2018) compared three traditional methods such as radial basis function neural network, extreme learning machine and back propagation neural network with deep learning and found that deep learning models exhibited better prediction capabilities. Pang et al. (2020) used LSTM with an embedded layer to predict stock markets.

Zhang et al. (2019) proposed a GAN architecture for stock market prediction with LSTM as discriminator and MLP as generator. They intend to optimize model in future by learning data distributions more accurately. Nelson et al. (2017) studied LSTM networks to predict stock markets. Akita et al. (2016) used deep learning models on financial data to predict bankruptcy. Hoseinzade and Haratizadeh (2019) proposed a CNN based framework for stock prediction. Zhang et al. (2019) proposed a deep learning model with LSTM and convolutional layers to process data of Limit Order Books (LOB). Their method showed better performance in extracting features and modelling. Polamuri et al. (2020) combined CNN model and representation learning model for stock prediction. Sezer and Ozbayoglu (2018) proposed a trading model known as CNN-TA based on 2-D CNN for determining holding and selling strategies. Zhang et al. (2018) proposed a novel methodology for stock market prediction with information fusion from different sources.

Chen et al. (2020) proposed hybrid deep learning model based on LSTM, MLP and attention mechanism. Their model is found to have better forecasting accuracy. However, they intend to improve it with sentiment models in future. Tsantekidis et al. (2017) proposed deep learning model based on CNN to ascertain movement of stocks using LOB data. Asad (2015) proposed an ensemble model for stock market prediction. A *meta*-learning algorithm is used to achieve ensemble of SVM, Random Forest and Relevance Vector Machine classifiers. They observed that ensemble method has least error rate. Eapen et al. (2019) proposed a deep learning model that combines bi-directional LSTM and CNN for stock index prediction. They could improve prediction performance with the hybrid approach. Bukhari et al. (2020) proposed an improved ARIMA and combined it with LSTM for stock market forecasting. Their model could improve accuracy in prediction.

Katayama et al. (2019) proposed a deep learning model based on sentiment polarity identification for financial market prediction. Their approach could improve polarity based market prediction. Lee et al. (2019) focused on a deep learning model that is based on Deep Q-Network with CNN as function apprximator and stock charts are taken as input for stock prediction. Polamuri et al. (2020) proposed MM-HPA, a hybrid algorithm that combines linear and non-linear prediction models for better performance. There are several insights from the literature. First, CNN is better for classification. Second, LSTM can better capture temporal data variations than RNN (Mai et al., 2019). Third, deep learning models suffer from lack of pre-processing (or quality of data) unless there is an efficient NLP based approach towards feature selection (Bukhari et al., 2020). Fourth, GAN is found effective in learning internal representations of data using unsupervised approach besides generating new data that resembles real data (Zhang et al., 2019). Fifth, a hybrid of linear and non-linear models could provide better performance in stock prediction (Rao et al., 2020). Sixth, there is need for GAN based framework that exploits CNN, LSTM and novel pre-processing to improve the state of the art. Considering these observations, in this paper, we proposed Stock-GAN based on deep learning models for better prediction of stock markets.

3. Preliminaries

Prediction of stock price movement using stock market data is non-trivial. By IanGoodfellow and his colleagues in 2014, a class of machine learning frameworks known as Generative Adversarial Networks (GAN) came into existence. It is in a game-theory setting where two neural networks contest with each other. GAN learns to generate new data from training set similar to that of training set. The architecture of GAN is shown in Fig. 1. It has two important components such as generator (G) and discriminator (D). "Generator is model used to generate new plausible examples from the problem domain while the discriminator is a model used to classify examples as real (*from the domain*) or fake (*generated*)".

Generator network directly generates a data sample that appears like real data. On the other hand, its adversary, known as discriminator network, distinguishes samples obtained from real data and samples generated by the generator framework. The discriminator is a classification model. As discussed in (Uddin, 2019), the objective function of the GAN architecture is in Eq. (1).

$$\arg \overset{\min}{G} \quad \overset{\max}{D} V(D,\ G) = E_{x\sim\ pdata\ (x)}[\log(D(x))] + E_{z\sim\ p_z\ (z)}[\log(1 - D(G(Z)))] \tag{1}$$

Where the discriminator function is denoted as D (x). This function results in probability that the input vector denoted as x is from training dataset. By taking x as input, the $D(x)$ produces a value between 0 and 1. Similarly, $G\ (z)$ is known as generator function which results in a matrix whose dimension is same as that of x depending on the z (noise vector). From the training dataset, probability distribution of samples is denoted as $P_{data}(x)$. The probability distribution of samples obtained from noise generator is denoted as $P_z(z)$. The expectation function which is resulted from the log-loss function as positive class is denoted as $E(.)$. The log-loss function is defined as in Eq. (2).

$$E(p|y) = \frac{-1}{N}\sum_{i=1}^{N}(y_i(logp_i) + (1 - y_i)(1 - p_i)) \tag{2}$$

Where the actual data is denoted as y_i, the estimation is denoted as p_i. When 0 or 1 is expected as response from the model, log function is used. When x is drawn from $p(x)$, with regard to probability distribution $p(x)$, $E(f(x)$ of given function $f(x)$ is expressed as in Eq. (3).

$$E_{x\sim\ p}(f(x)) \int p(x)f(x)dx \tag{3}$$

Two loops such as $\min_G V\ (D,G)$ and $\max_D V\ (D,G)$ are involved in Eq. (1). The aim of $\max_D V\ (D,G)$ is to maximize the right hand side by discriminator's parameter tuning. Equation (1) is the objective function which contains two loops denoting $\max_D V\ (D,G)$ and $\min_G V\ (D,G)$.Similarly, the aim of $\min_G V\ (D,G)$ is to minimize by generator's parameter tuning.

4. Proposed Stock-GAN Framework

GAN based approach with generator (G) and discriminator (D) is found to be suitable for effectively dealing with time-series stock market data. Without labelled data, GAN can quickly learn from internal representations of data, generate data, learn density distributions and use a trained discriminator as classifier. In this section, the proposed framework known as Stock-GAN, shown in Fig. 2, is described.

4.1. The Framework

Unlike the GAN IN where MLP plays the role of generator, LSTM is preferred to realize generator in Stock-GAN while CNN is used as discriminator. The rationale behind this is that LSTM is that it can handle time-series data well and is capable of learning long-term dependencies when compared with RNN while CNN is good for classification. Hybrid models with linear and non-linear prediction approaches are found effective in stock prediction. Therefore, apart from using LSTM and CNN as generator and discriminator respectively in Stock-GAN, there is strong feature extraction procedure from stock data where linear models like ARIMA are applied. Novel pre-processing methodology (Polamuri et al., 2020) is integrated as part of Stock-GAN. Otherwise deep learning models suffer from performance problems. We investigated on right set of hyperparameters suitable for (G) and (D). In order to tune the hyperparameters dynamically, Stock-GAN incorporates reinforcement learning framework along with Bayesian optimization technique.

A novel pre-processing methodology is used for improving quality of data prior to giving it to (G). Newest innovations in NLP are part of the methodology. For instance, BERT is used for observing sentiments. Trend directions are ascertained using Fourier transforms. High-level features are extracted using stacked autoencoders. Stock function approximation is made with a linear learning model known as ARIMA. Self-Organized Maps (SOM) is used to analyse anomalies in stock movements. This information is later useful to LSTM (G). Statistical checks are made to see that data is of high quality. The feature extraction process results in so many features and all of them may not contribute in stock prediction process. In order to filter them out (dimensionality reduction / feature selection), XGBoost is used to identify importance of features. Then PCA is applied and Eigen portfolios are created so as to reduce number of features generated by autoencoders. Fourier transforms used in the methodology create series of sine waves to analyse trend directions. The transforms are as in Eq. (4).

$$\mathrm{G(f)} = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}dt \tag{4}$$

ARIMA (Nau, 2014) is used to know stock trends and extract new features or patterns as in Eq. (5).

$$(1 - \lambda_1\mathrm{B})(1 - \lambda_2\mathrm{B})y_t = (1 - \theta_t\mathrm{B})e_t \tag{5}$$

ARIMA is an improved form of ARMA and provide better approximation of real stock price. After extracting high-level features using autoencoders, XGBoost (Lee et al., 2019) model used for feature importance prediction is as in Eq. (6).

$$\mathscr{L}^{(t)} = \sum_{t=1}^{n} l(yi, yt^{t-1} + f_t(X_i)) + \Omega(f_t) \tag{6}$$

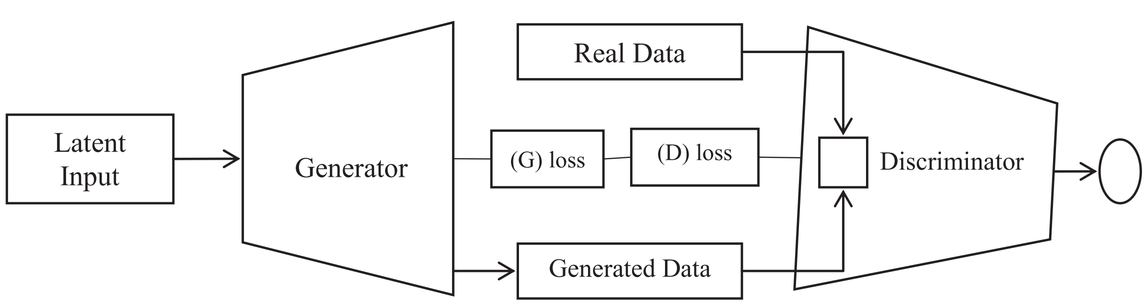

Fig. 1. Architecture of typical GAN model.

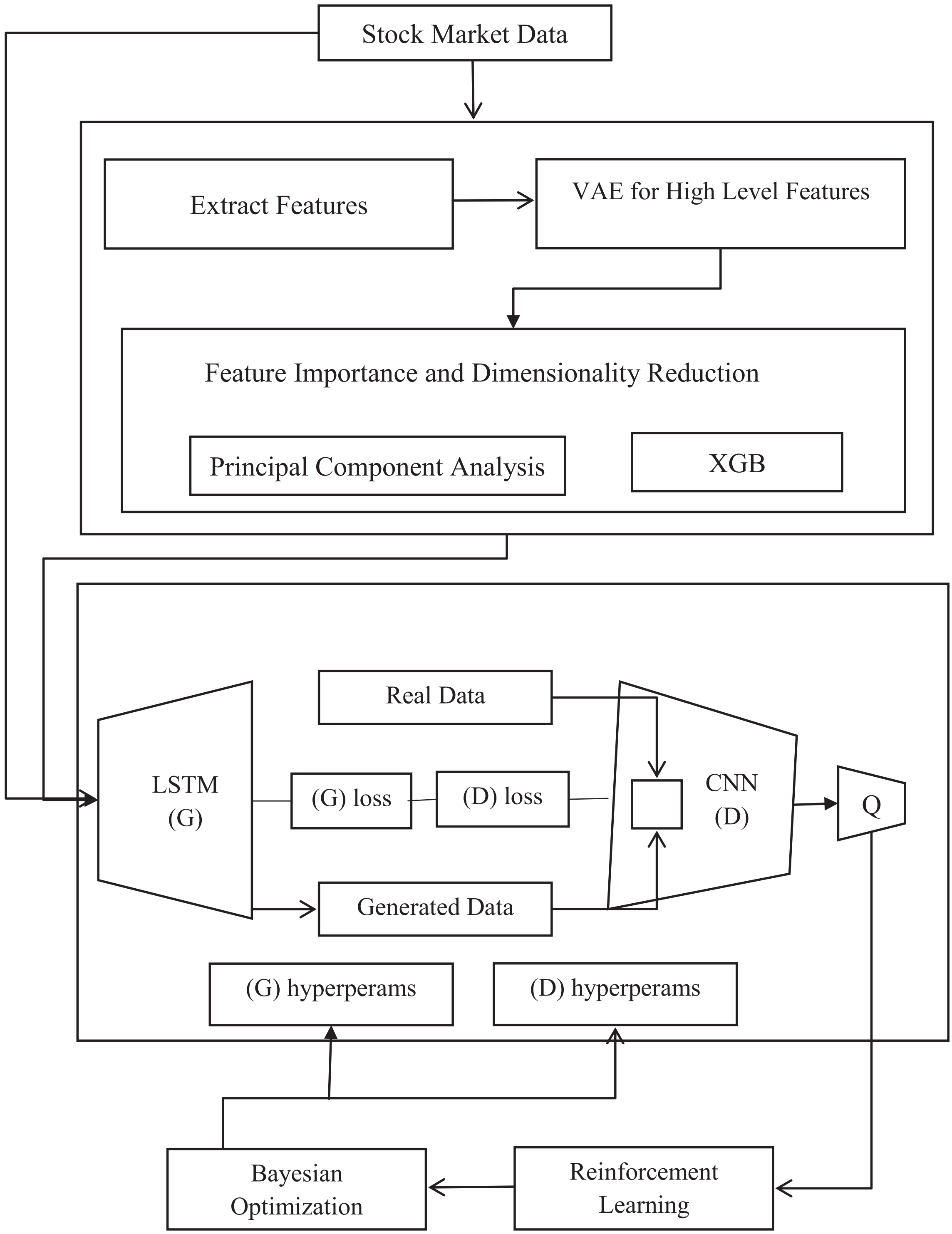

Fig. 2. Architecture of Stock-GAN.

A combined loss function for (G) and (D) is as in Eq. (7).

$$\mathrm{L}(\mathrm{D},\mathrm{G}) = \mathbb{E}_{x\sim\, p_r(x)}[\log\mathrm{D}(\mathrm{x})] + \mathbb{E}_{z\sim\, p_z(z)}[\log(1-\mathrm{D}(\mathrm{G}(\mathrm{z})))] \tag{7}$$

As RNN vanishes gradients many times, LSTM is used as (G) where four gates are used namely update gate, input gate, forget gate and output gate. LSTM is preferred over Gated Recurring Unit (GRU) as LSTM has 4 gates, has internal memory state and supports non-linearity. Equations from 8 to 13 show the mathematical model involved with LSTM.

$$g_t = \tanh\left(X_t W_{xg} + h_{t-1} W_{hg} + b_g\right) \tag{8}$$

$$i_t = \sigma(X_t W_{xi} + h_{t-1} W_{hi} + b_i) \tag{9}$$

$$f_t = \sigma\left(X_t W_{xf} + h_{t-1} W_{hf} + b_f\right) \tag{10}$$

$$o_t = \sigma(X_t W_{xo} + h_{t-1} W_{ho} + b_o) \tag{11}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{12}$$

$$h_t = o_t \odot \tanh(c_t) \quad (13)$$

The two activation functions used in the LSTM model are as in Eqs. (14) and (15).

$$\sigma(x) = \left[\frac{1}{1+\exp(-x_1)}, \cdots\cdots \frac{1}{1+\exp(-x_k)}\right]^T \quad (14)$$

$$\mathrm{Tanh(x)} = \left[\frac{1-\exp(-2x_1)}{1+\exp(-2x_1)}, \cdots\cdots, \frac{1-\exp(-2x_k)}{1+\exp(-2x_k)}\right]^T \quad (15)$$

CNN architecture shown in Fig. 3 is used as (D). It is preferred for reasons such as its ability to deal with spatial data and ability to detect features.

As presented in Fig. 3, the CNN has mechanisms to discriminate and classify data. In other words, it used to predict stock prices. Both LSTM (G) and CNN (D) need hyperparameters known as number of filters (filters), dropout used in LSTM (dropout), size of kernel in CNN (kernel_size), padding in CNN (padding), batch normalization momentum used in CNN (batchnorm_momentum), the alpha parameter required by LeakyReLU in GNN (lrelu_alpha), number of strides required by CNN (strides), learning rate required by CNN (cnn_lr), batch size required by CNN and LSTM (batch_size). These are tracked and optimized by using deep reinforcement learning and Bayesian optimization techniques Fig. 4.

4.2. Generative Adversarial Network based hybrid prediction Algorithm

The Multi-Model based Hybrid prediction Algorithm (MM-HPA), MAE and MSE values are presented for non-linear 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. All stock tickers in terms of MAE, MSE and correlation. The high correlation and low error rate reflected in the results indicate the performance of the proposed algorithm.

The reasoning behind this is that, as opposed to RNN, LSTM is better at handling time-series data and learning long-term dependencies, while CNN is better at classification. In stock prediction, hybrid models with linear and non-linear prediction approaches have been found to be accurate. As a result, in addition to using LSTM and CNN as generator and discriminator in Stock-GAN, there is a robust feature extraction procedure from stock data that employs linear models such as ARIMA. Stock-GAN incorporates a novel pre-processing technique. Deep learning models, on the other hand, have performance issues. We looked into the best set of hyperparameters for (G) and (H) (D). Stock-GAN includes a reinforcement learning system to dynamically tune the hyperparameters.

An algorithm named Generative Adversarial Network (GAN) based Hybrid Prediction Algorithm (GAN-HPA) is proposed to realize the Stock GAN framework.

Algorithm: Generative Adversarial Network based Hybrid Prediction Algorithm

---

Inputs
Stock dataset with actual stock prices *D*
Hyperparameters *H*
Output
Predicted stock prices *D'*
Step 1: Create stock price trend approximations using Fourier transforms G(f) = $\int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}\mathrm{d}t$
Step 2: Find predictions with ARIMA (linear model) $(1-\lambda_1\mathrm{B})(1-\lambda_2\mathrm{B})y_t = (1-\theta_t\mathrm{B})e_t$
Step 3: Apply stacked autoencoders on results of Step 1 and Step 2 to extract new features (for further denoising of data)
Step 5: Find feature importance using XGBoost (output is given to generator)
$\mathscr{L}^{(t)} = \sum_{t=1}^{n} l(yi, yt^{t-1} + f_t(X_i)) + \Omega(f_t)$
Step 6: For each iteration until convergence
Step 7: Deep learning with LSTM as generator (takes Step 5 output and hyperparameters)
Step 8: CNN as discriminator (takes real stock prices and LSTM predictions)
Step 9: Compute the combined loss function of generator and discriminator
L(D,G) = $\mathbb{E}_{x\sim p_r(x)}$[logD(x)] + $\mathbb{E}_{z\sim p_z(z)}$[log(1-D(G(z)))]
Step 10: Tune hyperparameters *H*

(*continued on next page*)

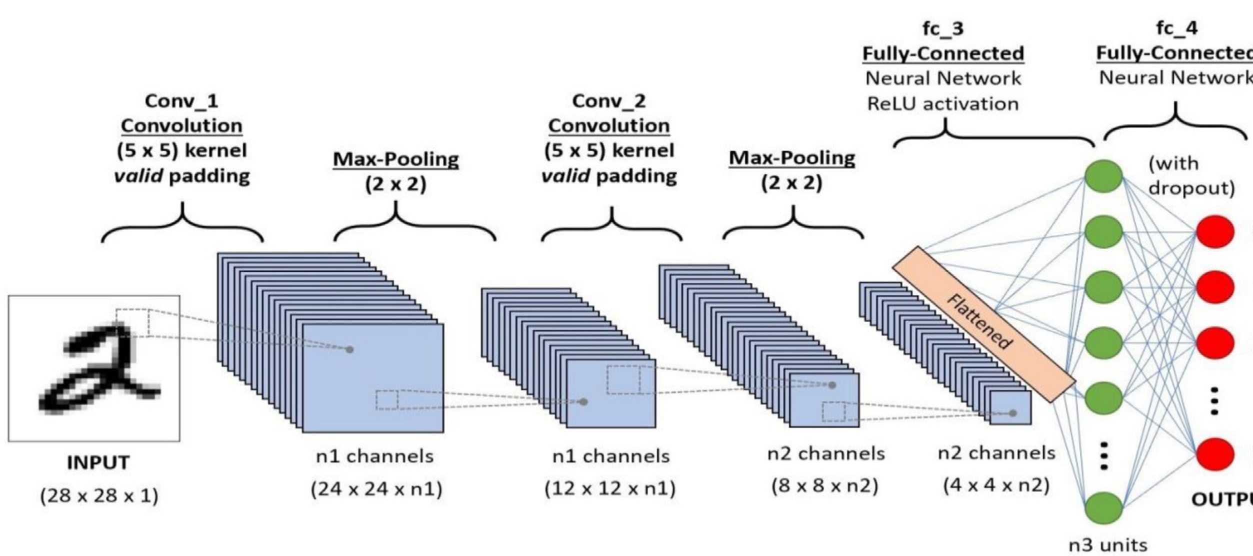

Fig. 3. Architecture of CNN.

Algorithm (*continued*)

Algorithm: Generative Adversarial Network based Hybrid Prediction Algorithm

Step 11: Update *D'*
Step 12: End For
Step 13: Compute error rate
Step 13: Return *D'*

Algorithm 1:. *Generative adversarial network based hybrid prediction algorithm*

As presented in Algorithm 1, it takes different inputs such as stock dataset and hyperparameters and produces predictions for given stock ticker dataset. In Step 1, stock prices trend approximations are made using Fourier transforms. As said before, Fourier transforms is better technique for dealing with time-series data. In Step 2, ARIMA is used to find stock predictions that are reused further. With these two steps most of the features are extracted from stock dataset. However, the noise in the features is to be removed. Stacked autoencoders are applied in Step 3 in order to achieve it and to ensure final set of features. At this stage, it is important to reduce number of dimensions. XGBoost is used in Step 5 to find importance of features that have discriminative power. The results of this and hyperparameters are given as input to generator (LSTM). The outcome of the generator and original stock data is given to discriminator. The discriminator classifies data and the iterative process from Step 6 through Step 12 continues until convergence. In the process, parameters are tuned in Step 10 using Bayesian approximation and predictions are updated in Step 11. Finally, the algorithm returns final predictions besides evaluating the error rate.

4.3. Hybrid of GAN-HPA and MM-HPA

Investigation is made further to exploit good features of MM-HPA defined in our previous research work and integrates with GAN-HPA. The hybrid algorithm that combines GAN-HPA and MM-HPA appropriately is named as Multi-Model Generative Adversarial Network Hybrid Prediction Algorithm (MMGAN-HPA) in Fig. 4.

Algorithm: Multi-Model Generative Adversarial Network Hybrid Prediction Algorithm

Inputs
Stock market dataset denoted as r_t(t = 1, ……., T)
Hyperparameters *H*
Output
Predicted stock prices *D'*

1. Initialize E to 2 (number of linear prediction models)

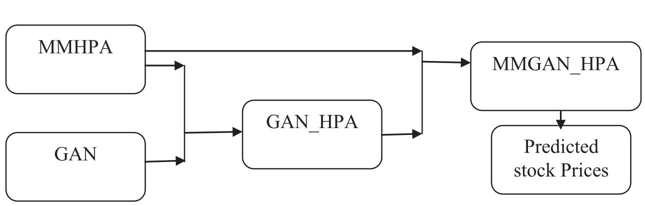

Fig. 4. Architecture of MMGAN-HPA.

Algorithm (*continued*)

Algorithm: Multi-Model Generative Adversarial Network Hybrid Prediction Algorithm

2. Initialize F to 1 (number of non-liner prediction models)
3. For each prediction mode e in E
4. Find $\hat{r}_{t,L_i}$ denoted as linear predictions
5. For each prediction model f in F
6. Find $\hat{r}_{t,NL_j}$ denoted as non-linear predictions
7. End For
8. End For
9. For each iteration until convergence
10. Deep learning with LSTM as generator (takes outcomes of Step 4 and Step 5 through Step 7 and hyperparameters)
11. CNN as discriminator (takes real stock prices and LSTM predictions)
12. Compute the combined loss function of generator and discriminatorL(D,G) = $\mathbb{E}_{x\sim\ p_r(x)}$[logD(x)] + $\mathbb{E}_{z\sim\ p_z(z)}$[log(1-D(G(z)))]
13. Tune hyperparameters *H*
14. Update *D'*
15. End For
16. Compute error rate
17. Return *D'*

Algorithm 2:. *Multi-Model Generative Adversarial Network Hybrid Prediction Algorithm*

Algorithm 2 takes stock dataset as input and it has pre-processing prior to GAN approach. In the pre-processing steps (Step 1 through Step 8), the algorithm employs linear and non-linear models in order to extract features from dataset. More on the linear and non-linear models used to arrive at features can be found in our prior work. The results of pre-processing and hyperparameters are given as input to generator (LSTM). The outcome of the generator and original stock data is given to discriminator. The discriminator classifies data and the iterative process from Step 9 through Step 15 continues until convergence. In the process, parameters are tuned in Step 16 using Bayesian approximation and predictions are updated in Step 17. Finally, the algorithm returns final predictions besides evaluating the error rate.

4.4. Evaluation metrics

Performance of the GAN-HPA model is evaluated using metrics such as Mean Absolute Error (MSA) and Mean Squared Error (MSE) as in Eqs. (16) and (17).

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

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

With these metrics, the prediction error is computed to ascertain performance of the hybrid prediction model.

5. Experimental Results

The Stock-GAN framework is implemented using Python data science platform. Stock datasets for various stock tickers such as TCS, Tata Steel, Axis Bank, BHEL, Wipro and Maruti are collected from NSE (National Stock Exchange).

Table 1
Performance of GAN-HPA.

Stock ticker	Performance of GAN-HPA		
	MAE	MSE	CORRELATION
TCS	0.00263397	0.00003490	0.99586647
BHEL	0.00267097	0.00002290	0.99686559
WIPRO	0.00239697	0.00002400	0.99716163
AXISBANK	0.00280396	0.00003020	0.99837178
MARUTI	0.00221797	0.00002160	0.99721564
TATASTEEL	0.00463494	0.00006770	0.99736265

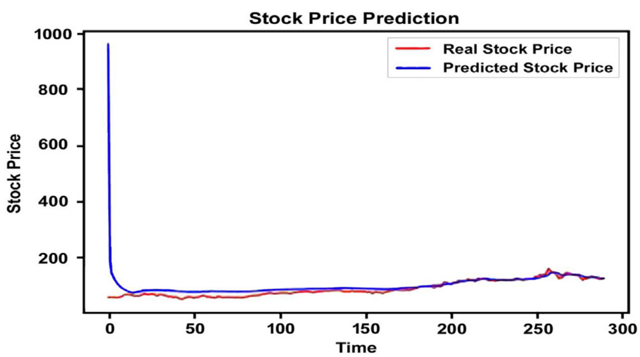

Fig. 5. Prediction performance with Axis Bank dataset.

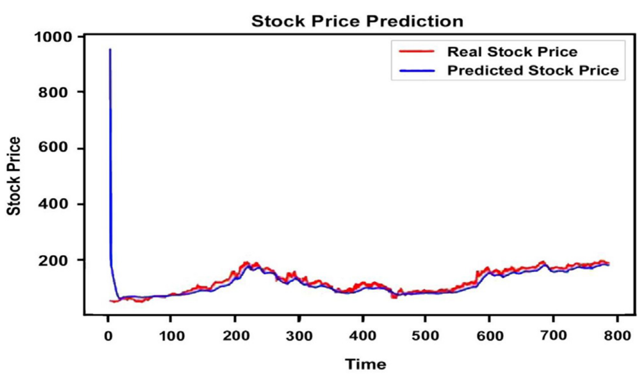

Fig. 6. Prediction performance with BHEL dataset.

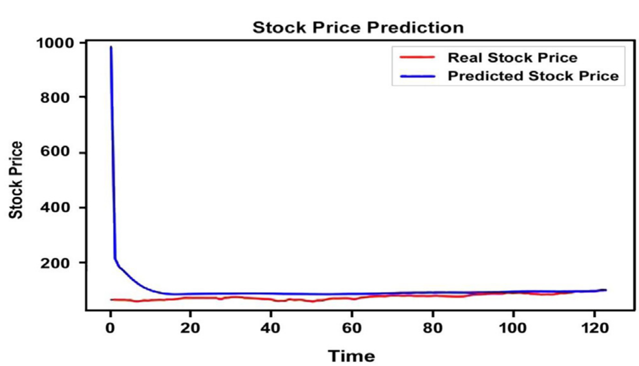

Fig. 7. Prediction performance with Maruti dataset.

5.1. Results of GAN-HPA

The stock prediction performance of GAN-HPA is evaluated using the metrics provided in Section 4.3. It is compared against another deep learning based hybrid prediction model known as MM-HPA.

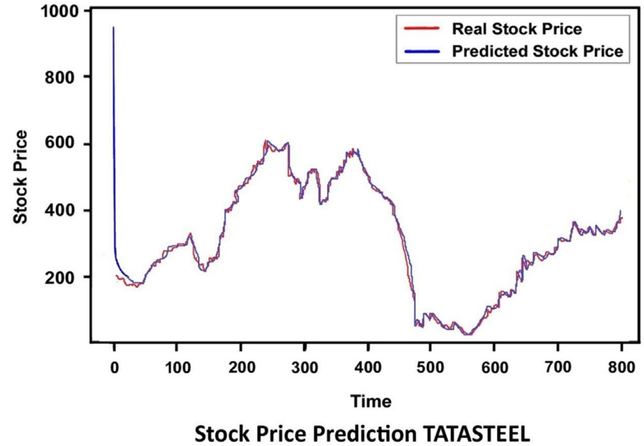

Fig. 8. Prediction performance with Tata Steel dataset.

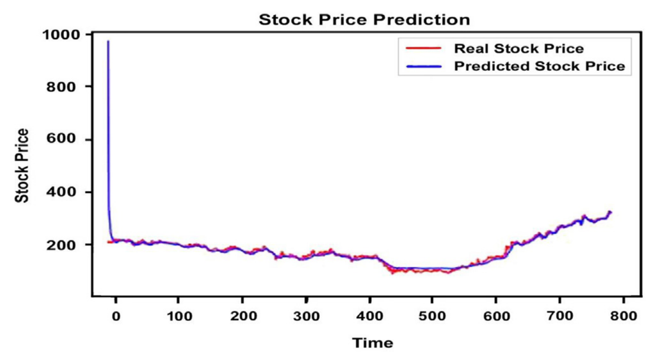

Fig. 9. Prediction performance with TCS dataset.

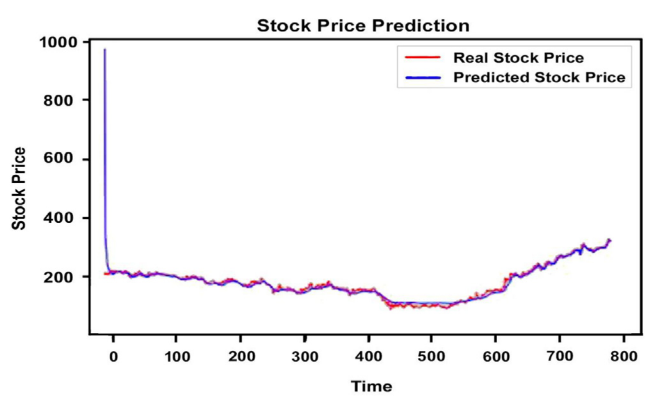

Fig. 10. Prediction performance with Wipro dataset.

As presented in Table 1, the prediction performance of GAN-HPA is provided for all stock tickers in terms of MAE, MSE and correlation. High correlation and low error rate reflected in the results indicate the performance of the proposed algorithm.

As presented in Fig. 5, the timeline is provided in horizontal axis while the vertical axis shows the stock price. There is close match between the real stock price and predicted stock price thus reflecting minimal error rate in prediction of stock prices.

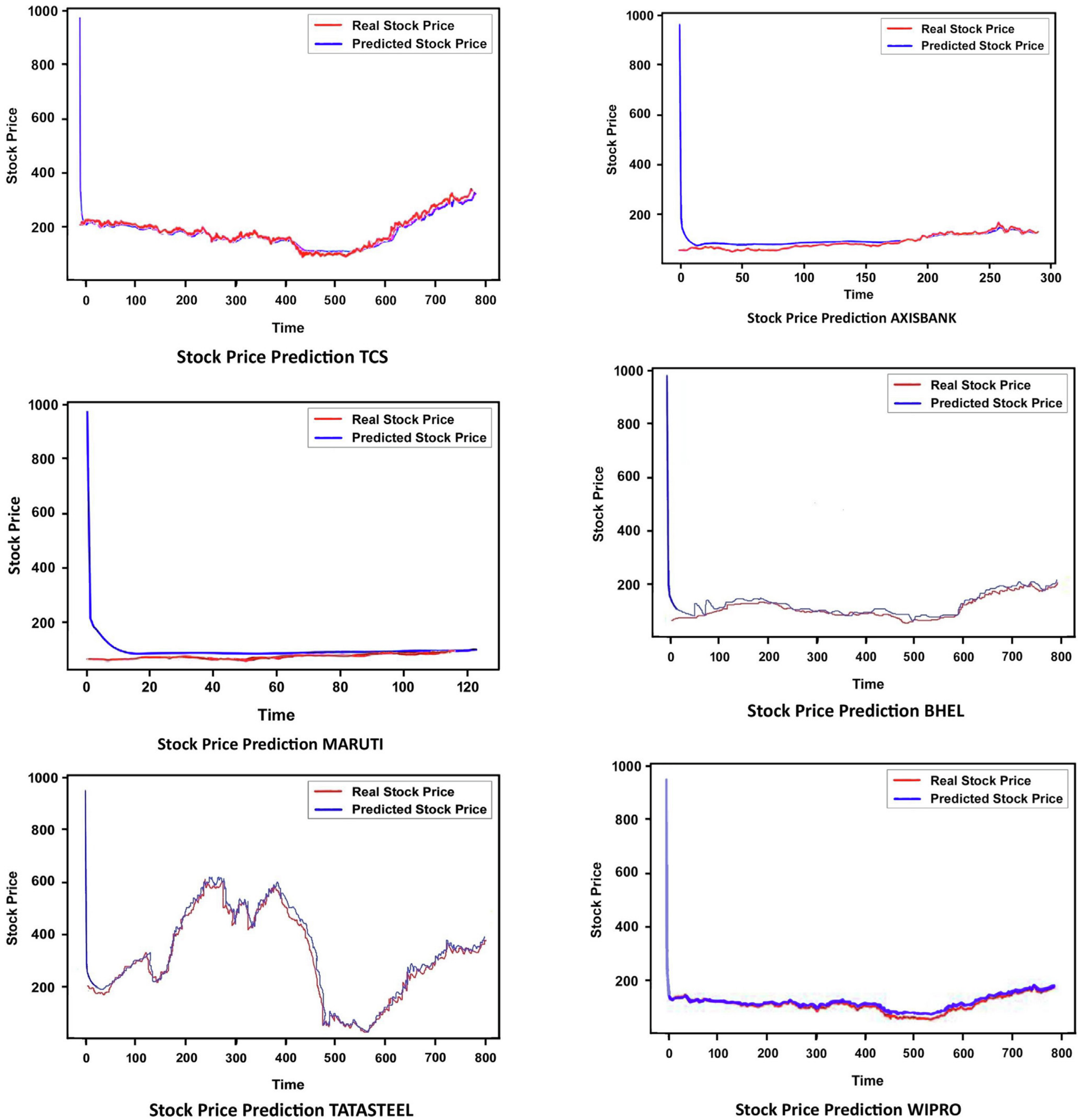

Fig. 11. Prediction Results of MMGAN-HPA Algorithm.

As presented in Fig. 6, the timeline is provided in horizontal axis while the vertical axis shows the stock price. There is close match between the real stock price and predicted stock price thus reflecting minimal error rate in prediction of stock prices. The prediction is based on the BHEL dataset.

As presented in Fig. 7, the timeline is provided in horizontal axis while the vertical axis shows the stock price. There is close match between the real stock price and predicted stock price thus reflecting minimal error rate in prediction of stock prices. The prediction is based on the Maruti dataset.

As presented in Fig. 8, the timeline is provided in horizontal axis while the vertical axis shows the stock price. There is close match between the real stock price and predicted stock price thus reflecting minimal error rate in prediction of stock prices. The prediction is based on the Tata Steel dataset.

As presented in Fig. 9, the timeline is provided in horizontal axis while the vertical axis shows the stock price. There is close match between the real stock price and predicted stock price thus reflecting minimal error rate in prediction of stock prices. The prediction is based on the TCS dataset.

As presented in Fig. 10, the timeline is provided in horizontal axis while the vertical axis shows the stock price. There is close match between the real stock price and predicted stock price thus reflecting minimal error rate in prediction of stock prices. The prediction is based on the Wipro dataset.

Table 2
Performance comparison among the three hybrid prediction models like MMHPA, GANHPA and MMGANHPA.

Stock ticker	MM-HPA			GAN-HPA			MMGAN-HPA		
	MAE	MSE	CORRELATION	MAE	MSE	CORRELATION	MAE	MSE	CORRELATION
TCS	0.00263400	0.00003490	0.99574200	0.00263397	0.00003490	0.99586647	0.00263344	0.00003490	0.995985974
BHEL	0.00267100	0.00002290	0.99674100	0.00267097	0.00002290	0.99686559	0.00267044	0.00002290	0.996985214
WIPRO	0.00239700	0.00002400	0.99703700	0.00239697	0.00002400	0.99716163	0.00239649	0.00002400	0.997281289
AXISBANK	0.00280400	0.00003020	0.99824700	0.00280396	0.00003020	0.99837178	0.00280340	0.00003020	0.998491585
MARUTI	0.00221800	0.00002160	0.99709100	0.00221797	0.00002160	0.99721564	0.00221753	0.00002160	0.997335306
TATASTEEL	0.00463500	0.00006770	0.99723800	0.00463494	0.00006770	0.99736265	0.00463401	0.00006770	0.997482334

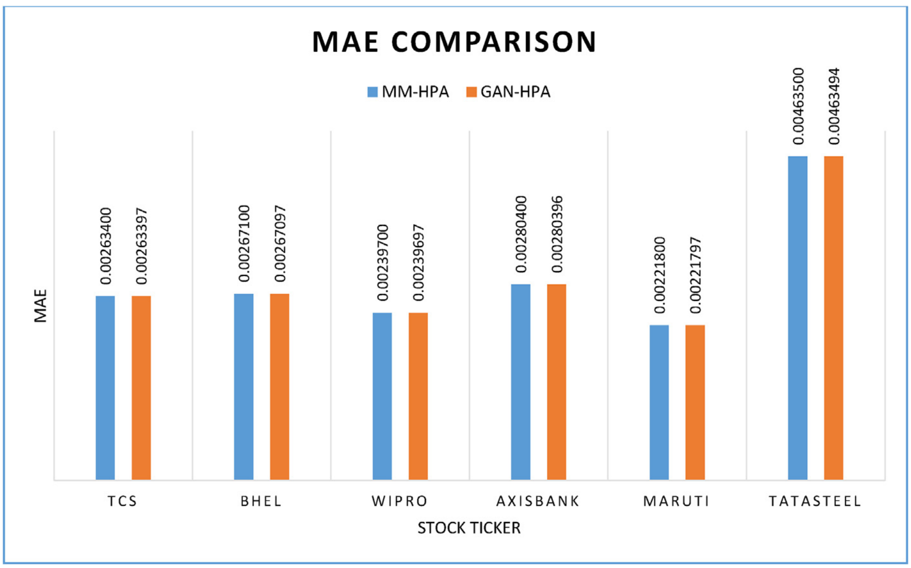

Fig. 12. Performance comparison in terms of MAE.

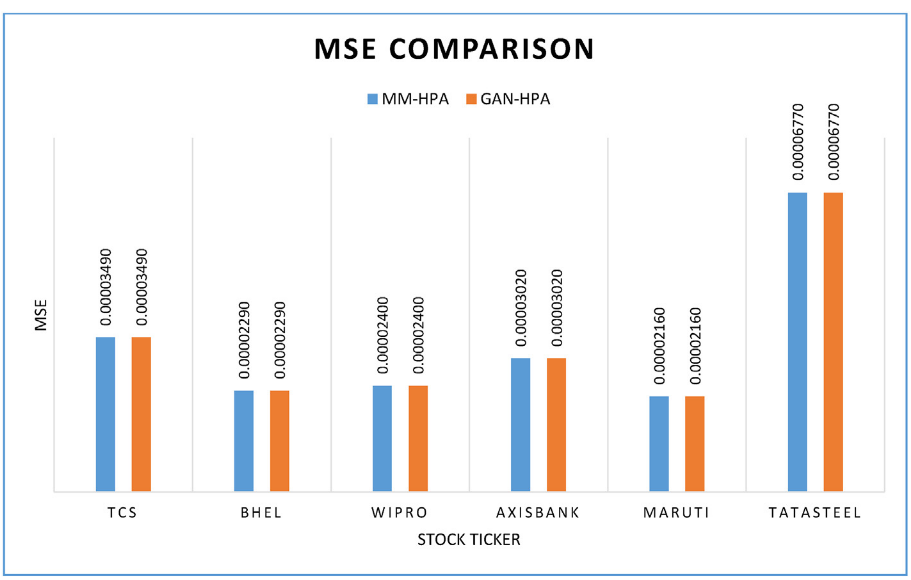

Fig. 13. Performance comparison in terms of MSE.

5.2. Results of MMGAN-HPA

Experimental results in terms of stock price predictions and performance measures like MAE, MSE and correlation are provided in this section.

As presented in Fig. 11, the prediction results for all the stock tickers are provided. The results revealed that there is further improvement in prediction accuracy of MMGAN-HPA when compared with that of GAN-HPA. It is evident with the results comparison presented in Section 5.3.

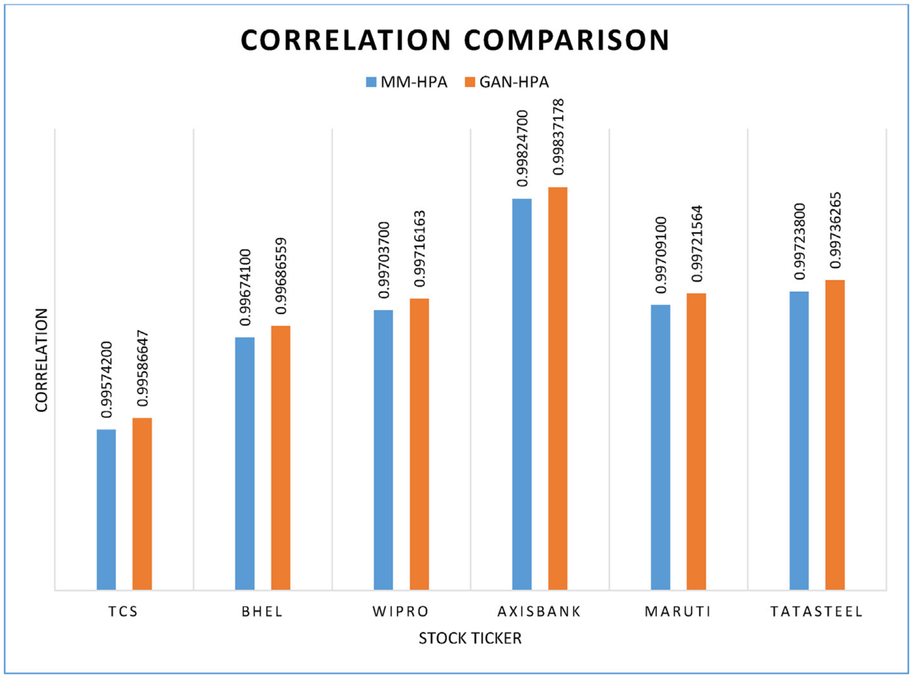

Fig. 14. Performance comparison in terms of correlation.

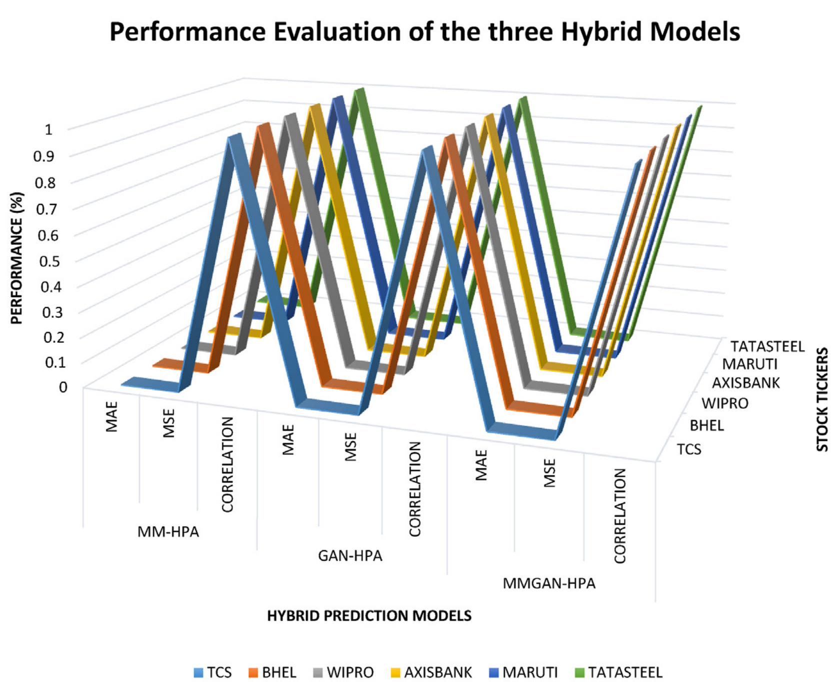

Fig. 15. Performance comparison among hybrid prediction models.

5.3. Performance Comparison

This section evaluates the performance of the two prediction algorithms namely GAN-HPA and MMGAN-HPA defined in this paper. Besides the performance of the two algorithms is compared with the MM-PHA, another hybrid algorithm for stock prices prediction, defined in (Pang et al., 2020).

The prediction performance of the proposed hybrid prediction model named GAN-HPA is compared against state of the art hybrid prediction model named MM-HPA. Both models are based on deep learning. The main difference is that the former is based on GAN architecture while the latter is based on the combination of linear and non-linear prediction models. Table 2 shows the performance difference in terms of MAE, MSE and correlation and compares all the three hybrid prediction models.

As presented in Figs. 12–14, the stock tickers are provided in horizontal axis and the error rate in terms of MAE, MSE and correlation are provided in vertical axis respectively. Different MAE and MSE are exhibited by stock tickers. In the same fashion different correlation values are observed for different stock tickers. However, the GAN-HPA showed relatively better performance over its predecessor MM-HPA. Though the both are hybrid prediction models, the Stock-GAN framework based hybrid prediction model GAN-HPA has advantages of generator and discriminator (LSTM-CNN) combination with hyperparamters tuning. Thus, its prediction performance is better than that of MM-HPA. The data presented in Table 2 and further visualized in Fig. 15 reveal that there is performance improvement when the MM-HPA and GAN-HPA combined to form yet another hybrid model known as MMGAN-HPA.

Hybrid deep learning models are proposed and found to be more efficient. Both linear and non-linear models are combined to have better prediction performance. There are many insights from existing deep learning models that led to the research carried out in this paper. First, CNN is better for classification. Second, LSTM can better capture temporal data variations than RNN. Third, deep learning models suffer from lack of pre-processing (or quality of data) unless there is an efficient NLP based approach towards feature selection. Fourth, GAN is found effective in learning internal representations of data using an unsupervised approach besides generating new data that resembles real data. Fifth, a hybrid of linear and non-linear models could provide better performance in stock prediction. Sixth, there is a need for a GAN based framework that exploits CNN, LSTM and novel pre-processing to improve the state of art. Considering these observations MMGAN-HPA is better performance.

6. Conclusion and Future Work

In this paper, Stock-GAN, a GAN based framework is proposed for stock price prediction. LSTM and CNN are used as generator and discriminator models. The GAN gets selected features from novel pre-processing methodology. The outcomes of the GAN are subjected to reinforcement learning and Bayesian optimization in order to adjust the hyperparameters of both models in Stock-GAN. Linear model such as ARIMA is used for stock approximation. Autoencoders are used to extract features from data. XGBoost is used to know the importance of features in order to filter out the features that do not contribute to the prediction objective function. An algorithm named Generative Adversarial Network (GAN) based Hybrid Prediction Algorithm (GAN-HPA) is proposed. Feature extraction is made from dataset using Natural Language Processing (NLP), Self-Organizing Maps (SOM), ARIMA, Eigen portfolios and Fourier transform. Further denoising of features is made with the help of stacked autoencoders. Feature importance prediction and dimensionality reduction is made using Principal Component Analysis (PCA) and XGB. Apache MXNet API and Python data science platform are used for implementation of Stock-GAN. Afterwards, MM-HPA and GAN-HPA combined to form yet another hybrid model known as MMGAN-HPA. Empirical study revealed that Stock-GAN achieves promising performance in stock prices prediction when compared with the state of the art in terms of stock price prediction, MAE, MSE and correlation coefficient. The proposed GAN-HPA algorithm outperforms the existing MM-HPA model in stock price forecasting. But MMGAN-HPA showed improvement of the GAN-HPA. In future, we intend to enhance Stock-GAN further to deal with diversified forms of stock data. Another direction for future work is to explore latent causal relationships in the stock market data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Akita, R., Yoshihara, A., Matsubara, T., Uehara, K., 2016. 1XPHULFDO DQG 7H [WXDO, QIRUPDWLRQ: Deep Learning for stock prediction using numerical and textual infromation. IEEE, 8–13.

Asad, M.: Optimized Stock market prediction using ensemble learning. 9th Int. Conf. Appl. Inf. Commun. Technol. AICT 2015 - Proc. 263–268 (2015). 10.1109/ICAICT.2015.7338559.

Bukhari, A.H., Raja, M.A.Z., Sulaiman, M., Islam, S., Shoaib, M., Kumam, P., 2020. Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting. IEEE Access. 8, 71326–71338. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2020.2985763.

Chen, L., Qiao, Z., Wang, M., Wang, C., Du, R., Stanley, H.E., 2018. Which artificial intelligence algorithm better predicts the chinese stock market? IEEE Access. 6, 48625–48633. https://doi.org/10.1109/ACCESS.2018.2859809.

Chen, Q., Zhang, W., Lou, Y.u., 2020. Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multi-layer perceptron, and bidirectional long-short term memory neural network. IEEE Access. 8, 117365–117376. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2020.3004284.

Eapen, J., Bein, D., Verma, A.: Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction. 2019 IEEE 9th Annu. Comput. Commun. Work. Conf. CCWC 2019. 264–270 (2019). 10.1109/CCWC.2019.8666592.

Hiransha, M., Gopalakrishnan, E.A., Menon, V.K., Soman, K.P., 2018. NSE stock market prediction using deep-learning models. Procedia Comput. Sci. 132, 1351–1362. https://doi.org/10.1016/j.procs.2018.05.050.

Hoseinzade, E., Haratizadeh, S., 2019. CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Syst. Appl. 129, 273–285. https://doi.org/10.1016/j.eswa.2019.03.029.

Katayama, D., Kino, Y., Tsuda, K., 2019. A method of sentiment polarity identification in financial news using deep learning. Procedia Comput. Sci. 159, 1287–1294. https://doi.org/10.1016/j.procs.2019.09.298.

Lee, J., Kim, R., Koh, Y., Kang, J., 2019. Global stock market prediction based on stock chart images using deep q-network. IEEE Access. 7, 167260–167277. https://doi.org/10.1109/Access.628763910.1109/ACCESS.2019.2953542.

Liu, Y., Zeng, Q., Yang, H., Carrio, A., 2018. Stock price movement prediction from financial news with deep learning and knowledge graph embedding. Springer International Publishing.

Mai, F., Tian, S., Lee, C., Ma, L., 2019. Deep learning models for bankruptcy prediction using textual disclosures. Eur. J. Oper. Res. 274, 743–758. https://doi.org/10.1016/j.ejor.2018.10.024.

National Stock Exchange, India. Available at: http://www.nseindia.com.

Nau, R., 2014. Mathematical structure of ARIMA models. 1, 1–8.

Nelson, D.M.Q., Pereira, A.C.M., De Oliveira, R.A.: Stock market's price movement prediction with LSTM neural networks. Proc. Int. Jt. Conf. Neural Networks. 2017-May, 1419–1426 (2017). 10.1109/IJCNN.2017.7966019.

Pang, X., Zhou, Y., Wang, P., Lin, W., Chang, V., 2020. An innovative neural network approach for stock market prediction. J. Supercomput. 76 (3), 2098–2118. https://doi.org/10.1007/s11227-017-2228-y.

Polamuri, S.R., Srinivas, K., Krishna Mohan, A.: Stock market prices prediction using random forest and extra tree regression. Int. J. Recent Technol. Eng. 8, 1224–1228 (2019). 10.35940/ijrte.C4314.098319.

Polamuri, S.R., Srinivas, K., Mohan, A.K., 2020. Multi model-based hybrid prediction algorithm (MM-HPA) for Stock Market Prices Prediction Framework (SMPPF). Arab. J. Sci. Eng. https://doi.org/10.1007/s13369-020-04782-2.

Polamuri, S.R., Srinivas, K., Mohan, A.K., 2020. Novel greedy heuristic optimized multi-instance quantitative for the prediction of stock price. Solid State Technol. 63 (5), 4654–4672.

Rao, P.S., Srinivas, K., Mohan, A.K.: A Survey on Stock Market Prediction Using Machine Learning Techniques. In: Machine Learning and Applications. pp. 923–931 (2020).

Sezer, O.B., Ozbayoglu, A.M., 2018. Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. Appl. Soft Comput. J. 70, 525–538. https://doi.org/10.1016/j.asoc.2018.04.024.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Forecasting stock prices from the limit order book using convolutional neural networks. Proc. - 2017 IEEE 19th Conf. Bus. Informatics, CBI 2017. 1, 7–12 (2017). 10.1109/CBI.2017.23.

Uddin, S.M.N., 2019. Intuitive approach to understand the mathematics behind GAN GAN. Intuitive Approach Math. 10.13140/RG.2.2.12650.36805.

Vargas, M.R., De Lima, B.S.L.P., Evsukoff, A.G.: Deep learning for stock market prediction from financial news articles. 2017 IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. CIVEMSA 2017 - Proc. 60–65 (2017). 10.1109/CIVEMSA.2017.7995302.

Zhang, X., Zhang, Y., Wang, S., Yao, Y., Fang, B., Yu, P.S., 2018. Improving stock market prediction via heterogeneous information fusion. Knowledge-Based Syst. 143, 236–247. https://doi.org/10.1016/j.knosys.2017.12.025.

Zhang, K., Zhong, G., Dong, J., Wang, S., Wang, Y., 2019. Stock market prediction based on generative adversarial network. Procedia Comput. Sci. 147, 400–406. https://doi.org/10.1016/j.procs.2019.01.256.

Zhang, Z., Zohren, S., Roberts, S., 2019. DeepLOB: Deep convolutional neural networks for limit order books. IEEE Trans. Signal Process. 67 (11), 3001–3012. https://doi.org/10.1109/TSP.7810.1109/TSP.2019.2907260.