FMDB *Transactions on Sustainable Computing Systems*

Chinese and Iran Stock Market Analysis Using DPO-Based Gated-Recurrent with Long-Short Term Memory

B. Gunapriya[1,*], M. Arunadevi Thirumalraj[2], B. Manjunatha[3], S. Gopikha[4], Robert Balku[5]

[1]Department of Electrical and Electronics Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[2]Department of Computer Science and Engineering, Karunya Institute of Technology and Science, Coimbatore, Tamil Nadu, India.
[2]Department of Computer Science and Business Management, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India.
[3]Department of Mechanical Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[4]Department of Information Technology, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India.
[5]Department of Business Administration, Budapest Metropolitan University, Budapest, Hungary.
gunapriya1978@gmail.com[1], aruna.devi96@gmail.com[2], manjunatha.princi@gmail.com[3], gopikha.in@gmail.com[4], robert.balku@gmail.com[5]

Abstract: The complex, dynamic, and non-linear nature of stock group value projection has historically made it both an enticing and demanding task for shareholders. An integral part of investing in stocks is accurately forecasting their prices. Stock price data is notoriously difficult to forecast with any degree of accuracy due to its high frequency, nonlinearity, and lengthy memory. This article focuses on the market predictions of GOF groups. Experimental evaluations were conducted using datasets such as the Tehran Stock Exchange and the Chinese Stock Index, specifically the CSI300 index. For each category, we combed through a decade-long database for relevant information. A1,2,5,10-,15-,20-, and 30-day time horizons are included in the value forecasts. A 1-D convolutional neural network is used to extract features, which are then passed to classifiers. A new predictive model is introduced here that effectively extracts nodes' trust levels from their spatial and temporal performance metrics by combining the power of VARMA, GRU, and LSTM neural networks. Doctor and patient optimisation (DPO) enhances classification accuracy by selecting the LSTM weight appropriately. When compared to LSTM, RNN, and CNN, the proposed model consistently outperforms them all in predicting stock prices.

Keywords: Vector Autoregressive; Moving Average; Long Memory; Gated Recurrent Unit; Doctor and Patient Optimisation; Stock Prediction; Non-Linear Nature; LSTM Neural Networks.

Received on: 04/12/2024, **Revised on:** 27/02/2025, **Accepted on:** 09/04/2025, **Published on:** 07/09/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSCS

DOI: https://doi.org/10.69888/FTSCS.2025.000478

Cite as: B. Gunapriya, M. A. Thirumalraj, B. Manjunatha, S. Gopikha, and R. Balku, "Chinese and Iran Stock Market Analysis Using DPO-Based Gated-Recurrent with Long-Short Term Memory," *FMDB Transactions on Sustainable Computing Systems,* vol. 3, no. 3, pp. 203-214, 2025.

1. Introduction

*Corresponding author.

The long-term unpredictability of stock prices makes forecasting inherently difficult. According to the dated market theory, stock prices can't be predicted because they behave randomly. However, new technical studies reveal that most stock prices are based on historical data; hence, trend analysis is crucial for accurate valuation [1]. Political events, overall economic circumstances, and other economic variables all affect the groupings and movements of the stock market [2]. A large market capitalisation is used to calculate the value of stock groupings. Statistical information can be derived from stock prices using various technical indicators. Stock indexes are a reliable indicator of a country's economic health, as they reflect the prices of highly invested stocks. Take stock market capitalisation as an example; research shows that it has a beneficial effect on national economic growth [3]. Investors face risk due to the uncertainty in stock price movements. Furthermore, governments often struggle to accurately assess market conditions. The inability of statistical models to accurately predict stock prices and movements is commonly attributed to the fact that stock prices are typically nonlinear, nonparametric, and dynamic [4]. According to popular belief, stock markets are essentially a collection of random numbers, and the Iranian stock market is no exception. It follows certain principles based on the preceding day's closing price.

Stock price forecasting is inherently challenging; as most traditional time-series prediction methods rely on stagnant trends [5]. Furthermore, given the numerous factors involved, forecasting stock prices is inherently challenging. The market is akin to a voting machine in the short term, but more like a weighing machine in the long term; therefore, it's possible to forecast its movements over a longer period [6]. One of the most powerful tools is machine learning (ML), which utilises various algorithms to enhance its performance in specific case studies. The widespread assumption is that ML is very good at finding meaningful data and patterns in datasets [7]. Ensemble models have proven more effective than individual methods for forecasting time series [8]. They are a machine learning-based alternative to classic ML approaches that combine multiple common algorithms to solve a specific issue. Boosting and bagging are among the most well-known and widely used ensemble methods for machine learning prediction problems. The best data scientists in the world have been leveraging new tree-based models, such as gradient boosting and XGBoost, to their advantage in recent contests. Indeed, deep learning (DL), a relatively new ML trend, has the potential to extract valuable information from financial time series with remarkable efficiency and what one would call a "deep nonlinear topology" in its underlying structure [9]. Due to their superior performance, recurrent neural networks (RNNs) have been a huge hit in the field, in contrast to simpler artificial neural networks [10].

Training will be inadequate if only the most recent data are utilised because the stock market prediction process is obviously dependent on both current and historical information. RNNs are well-suited for economic forecasting because they can remember previous events and establish connections between network nodes [11]. An enhanced subset of the RNN approach used in deep learning is the long short-term memory (LSTM) model. With its three gates, LSTM can analyse either individual data points or entire sequences, and it also eliminates the issues with RNN cells. For data that is not too unpredictable, traditional econometric approaches, such as ARMA and GARCH, achieve better forecast precision and accuracy [12]. To understand the nonlinear correlations between stocks and their influencing variables, traditional ML methods such as Support Vector Machines and Random Forests could be useful. However, these approaches place undue emphasis on sample selection during model building, making both model creation and updating rigid and hindering the ability to meet forecasting criteria [13]. Compared with more conventional machine learning models, deep learning models outperform them in analysing stock prices due to their superior learning capacity and self-adaptive capabilities. An example of a recurrent neural network is the LSTM network, which accounts for the time-series and non-linear nature of stock data and is thus better able to handle lengthy input sequences.

Due to its superior memory capacity and gate structure, it is widely used for stock prediction, unlike other recurrent neural networks, which are limited to remembering short sequences [14]. In neural networks, the attention mechanism addresses the information overload problem when processing capacity is limited and assigns processing resources to more important tasks. The present research on the subject seldom addresses either the issue of local stock data features affecting the accuracy of model predictions or the fact that LSTM models for stock price forecasting do not permit a tighter relationship between historical and future information [15]. This research proposes a novel approach to addressing these challenges in high-performance blockchain networks. To establish the reliability of nodes based on their spatial and temporal performance metrics, our approach leverages the combined power of VARMA, GRU, and LSTM neural networks. Furthermore, the DPO method appropriately selects weights for the LSTM. Our validation analysis uses two publicly available datasets [27].

2. Related Works

Lu and Xu [16] employed support vector regression and logistic regression to forecast the next day's capital trajectory. Support vector regression and logistic regression both have mediocre accuracy in predicting future stock prices (89.97% and 89.90%, respectively). Due to the time, a new dataset is created by transforming the original dataset into a supervised learning format. This improves the forecast rule, which, in turn, yields decisions with a sequential support vector regression and logistic regression accuracy of 89.95%. One effective method for predicting stock prices is the Time-series Recurrent Neural Network (TRNN) proposed by Liu et al. [17]. The suggested methodology uses sliding windows to analyse time series data after establishing trade volume. Data compression is achieved by leveraging financial market characteristics to extract data patterns and pinpoint turning points. An RNN-based two-dimensional price-volume connection is employed to enhance the impact of volume on the current stock price. The TRNN model ensures accuracy and efficiency by processing and compressing trade volume data. We evaluate the TRNN model's performance and accuracy compared to the initial RNN and LSTM models. We then discuss how our TRNN model can be extended to different domains and whether comparable enlarged techniques are feasible.

A dual-channel attention model, LSTMA+TCNA, was constructed by Sivadasan et al. [18]. To enhance the weight of features at crucial moments, self-attention is applied to both the LSTMA and TCNA channels; the former is used to extract features, and the latter to capture dependencies. First, use the respective LSTMA and TCNA channels to forecast each frequency mode sub-window independently. Then combine the outcomes from the two channels to obtain the expected values. By combining the forecasts from each sub-window for each frequency mode, we can get the final forecasted stock series. The VMD-LSTMA+TCNA outperforms other state-of-the-art approaches across prediction accuracy, resilience, and generalizability, according to comprehensive trials conducted across stock markets. For more accurate predictions, try using GRU and LSTM networks, as suggested by Sivadasan et al. [18]. The results of comparing various datasets show that: (1) the MAPE of the current model is 1.748, while that of the proposed model is 0.6243; (2) the MAPE of the existing model is 1.92, while that of the recommended perfect is 0.7924; (3) the optional perfect has 1.191, while that of the current model is 2.99; and (4) the MAPE of the whole set of models is 0.7924, 0.8587, 1.191, and 0.8587, respectively. After that, we employ OHLC and a selection of technical indicators—including SMA, EMA, RSI, MACD, and ADX—to further refine the models.

We present an advanced method for predicting price trends and conducting trading simulations for the Shanghai Stock Exchange index that combines Boosting, Bagging, and NSGA-II. Deng et al. [19]. The SHAP technique is also used to interpret the model, enabling us to examine the sentiment variables and their importance in predicting outcomes on both global and local scales. Compared with the benchmark approaches, the suggested approach achieves higher hit ratios, cumulative returns, and maximum drawdowns, as shown in the trial data. This proves that the suggested approach can successfully simulate trading on Chinese stock indices and forecast price trends with a high degree of accuracy while minimising risk and maintaining a steady return. In addition, the suggested method's interpretability was enhanced by incorporating SHAP; as a result, market participants would have a useful reference to help them understand the relevant sentiment components and make relative judgments. To fully represent the complex relationships between stocks and their changes over time, Qian et al. [20] presented the MDGNN framework, which utilises a discrete dynamic graph. The graph provides a holistic view of the interconnections among stocks and related companies. Furthermore, a dynamic and effective method for forecasting stock investments is provided by encoding the temporal evolution of multiplex relations using the Transformer construction. Additionally, compared to state-of-the-art (SOTA) stock investment approaches, our proposed MDGNN system outperforms them on public datasets.

After outlining the reasoning for selecting the following economic and financial time series variables, Zhang [21] illustrates their link to market return: secondly, this paper uses Python to conduct descriptive statistical analysis of the following variables: CPI, momentum, dividend rate, market book ratio, P/E ratio, a turnover rate of trading volume calculation, market-capitalization-weighted idiosyncratic volatility, logarithm of stock market liquidity index, Amihud illiquidity index, skew. Thirdly, it allows you to compare trends by plotting variables and yields. After that, it uses multiple regression to build a neural network model using variables that are substantially associated. The last step is to forecast when the Chinese stock market will recover using the BP neural network. Two popular network models for predicting stock market values have been proposed by Mukherjee et al. [22]. The CNN and the Deep Feed-forward Neural Network (DFNN). By analysing data from the past few days, the models have been able to predict future data values. So long as the dataset remains valid, this method will continue to repeat itself recursively. Significant progress has been made in optimising this prediction using deep learning. On the one hand, the CNN model achieved 98.92% accuracy, while the ANN model achieved 97.66%. To make predictions, the CNN model relies on 2D histograms generated from the quantal dataset over a specific time period. The study of such datasets has not previously used this method. The current epidemic, which triggered a sharp weakening in stock prices, served as a case study for the representation's testing. With a 91% accuracy rate, the study's findings were respectable.

A stock market prediction model is proposed by Verma et al. [23] that incorporates TI computing, feature engineering, and stock market prediction as its modules. The Discrete Wavelet Transform (DWT) is proposed as a data decomposition tool in the feature engineering component, with Chicken Swarm Optimisation (CSO) to manage the large number of features produced by DWT. To select the optimal subset of features, CSO is utilised. A feature engineering component, DWT-CSO, has been proposed. Deep Learning (DL) and Machine Learning (ML) algorithms are used to forecast stock market trends. For this experiment, we utilise datasets containing stock indexes from both the US (S&P 500 and DJI) and India (NIFTY 50 and BSE). The suggested DWT-CSO enhanced performance. An improvement of 19.43% for NIFTY50, 15.89% for BSE, and 19.59% for the S&P 500 is reflected in the enhanced accuracy of the prediction models.

The Wilcoxon test is used to statistically investigate CSOs. To precisely estimate stock market prices, Patil et al. [24] propose a Deep Recurrent Rider LSTM approach. The two classifiers used for accurate stock market forecasting are Deep Recurrent

Neural Network (Deep RNN) and Rider Deep Long Short-Term Memory (Rider Deep LSTM). Combining the Rider idea with Deep LSTM yields the Rider Deep LSTM, while the suggested Shuffled Crow Search Optimisation (SCSO) trains the Deep RNN. In addition, the SCSO is created by combining the SSO and CSA algorithms, which stand for Shuffled Shepherd Optimisation and Crowd Search, respectively. The last step is to use the error condition to determine the expected output. With an MSE of 0.018 and an RMSE of 0.132, respectively, the suggested Deep Recurrent Rider LSTM demonstrated superior performance and accuracy. The proposed classification methodology enhances classification accuracy and yields reliable predictions for the stock market. Novel frameworks for RNN, LSTM, GRU, and Bi-LSTM models have been proposed by Bhambu [25]. We conducted a study comparing the suggested time series forecasting models across short- and long-horizon settings. When properly tuned, the Bi-LSTM technique outperforms competing deep neural networks, according to computational findings computed across five time-series datasets.

3. Proposed System

3.1. Research Data Description

3.1.1. First Dataset

Using data generously provided by Metals, this study intends to produce a short-run projection for the developing Iranian stock market. The data is available from November 2009 to November 2019. Ten technical indicators are computed using the group prices, lows, highs, and opening and closing times. This study relies on data collected from the Exchange Technology Management Co (TSETMC) online repository [26]. A preprocessing step must be performed before the data can be used for training. Data cleaning is a technique used to identify and correct incorrect entries in a dataset. It involves identifying unnecessary or incorrect elements in the data and then correcting, replacing, or removing them. The interquartile range (IQR) is a measure of statistical dispersion used to detect and adjust datasets that contain outliers. It is resistant to these outliers. The values of ten practical pointers for each group are individually normalised to avoid the impact of larger indicators on smaller ones. This is an essential consideration.

3.1.2. Second Dataset

This study uses the CSI 300 Index data set, which spans 2675 trading days in 2021, to forecast stock prices and closing prices. Among the most-watched stock market indexes in China, the Shanghai Shenzhen CSI 300 Index ranks among the highest [28]. It is considered a barometer of market movements on both the Shanghai and Shenzhen stock exchanges, as it comprises 300 A-share firms listed on those exchanges. The chosen features include: volume, turnover, return, opening price, lowest price, maximum price, and closing price. A straightforward, one-step prediction method was employed. Training, testing, and validation sets all utilise the same data in a 6:2:2 ratio.

3.2. Data Pre-Processing

There are missing values in the data from several stations due to sensor errors. We employed a linear interpolation method to address this issue. By averaging the last two observations in the dataset, linear interpolation filled in the gaps where values were missing. It all starts with this formula.:

$$f(x) = f(x_0) + \frac{f(x_1)-f(x_0)}{x_1-x_0} \times (x - x_0) \quad (1)$$

The above equations describe a relationship between an independent variable (x) and two known values (x_0 and x_1) for that variable. Weather and traffic data, each with its own unique physical characteristics and dimensions, are among the many variables that affect air quality. To determine air quality, these parameters may be useful for precise analysis and smooth training. In the data pre-processing stage of our methodology, we have integrated all data sources into a single dataset. The following equation is used to normalise each dataset using linear scaling: where x^i represent the normalised data.

$$x^i = \frac{x-x_{min}}{x_{max}-x_{min}} \quad (2)$$

3.3. Feature Extraction using 1D-CNN

There are typically four layers in a CNN model: input, convolutional, pooling, and output. The following equation describes a standard convolution procedure:

$$X^{(l)} = \phi\left(X^{(l-1)} * W^{(l)} + b^{(l)}\right) \tag{3}$$

$X^{(l)}$ and $X^{(l-1)}$ symbolise the lth and $(l-1)$th layers. $W^{(l)}$ and $b^{(l)}$ are, respectively, the activation function, meaning χ, and the lth layer's weight and bias. To optimise the model setup and reduce data dimensions, the CNN network includes a pooling layer after the convolutional layer. The pooling layer can select useful data from the input layer. The following equation depicts the combination of convolutional and pooling layers in a CNN.

$$X^{(l-1)} = \phi\left(\text{pool}\max X^{(l+1)} * W^{(l+1)} + b^{(l+1)}\right) \tag{4}$$

To capture the spatial-temporal characteristics across all the data, one-dimensional convolutional layers are employed to extract recurrent local features and deep spatial similarity features from various datasets.

3.4. Varma GRU LSTM-Based Predictive Model

To address these issues, we developed a predictive model utilising VARMA and a combination of GRU and LSTM approaches. A combination of LSTM and GRU operations is employed to analyse these relative trust levels, facilitating the discovery of hidden patterns in the data. A collection of input (i), functional (f), output (c), and convolutional (c) characteristics is first estimated by the model using equations 8, 10, 11, and 12, in the following way:

$$i = \mathrm{var}(RTI * U^{i} + h(t-1) * W^{i}) \tag{5}$$

Where U and W represent constants of the LSTM procedure, while h characterises a kernel metric, which is incrementally updated by the GRU procedure. The var function is evaluated via equation (6) as follows,

$$var(x) = \frac{\left(\sum_{i=1}^{N}\left(x(i) - \sum_{j=1}^{N}\frac{x(j)}{N}\right)^{2}\right)}{N+1} \tag{6}$$

Where NN is the total count of input samples.

$$f = \mathrm{var}\left(RTI * U^{f} + h(t-1) * W^{f}\right) \tag{7}$$

$$o = \mathrm{var}(RTI * U^{o} + h(t-1) * W^{o}) \tag{8}$$

$$C = \tanh(RTI * U^{g} + h(t-1) * W^{g}) \tag{9}$$

A fusion of these features is used to estimate a cascaded temporal output (T) via equation (10),

$$T = \mathrm{var}(f * RTI(t-1) + i * C) \tag{10}$$

Based on these metrics, the output kernel is updated via equation (11),

$$h(out) = \tanh(T) * o \tag{11}$$

The output kernel and cascaded temporal output forgetting and retaining (z and r) factors via equations (12-13) as follows,

$$z = \mathrm{var}(Wz * [h(out) * T]) \tag{12}$$

$$r = \mathrm{var}(Wr * [h(out) * T]) \tag{13}$$

A fusion of these metrics is achieved through equation (14), which enhances the feature vector, and equation (15), which estimates the efficient kernel metric sets.

$$xout = (1-z) * h(t) + z * h(out) \tag{14}$$
$$h(t) = \tanh(W * [r * h(out) * T]) \tag{15}$$

The value of h is used to update the LSTM, and this is continued across multiple Iteration sets. These iterations are finished once equation (16) is satisfied,

$$\frac{var(xout(new))}{var(xout(old))} \approx 1 \quad (16)$$

Following convergence, the VARMA (Vector Autoregressive Moving Average) model leverages these LSTM and GRU properties to forecast node trust levels. Parts of the VARMA model are autoregressive (AR), and others are moving-average (MA). Here we see the representation of the Autoregressive (AR) model in equation (17),

$$\mathrm{T(t) = c + \phi T(t-1) + \theta e(t-1) + e(t)} \quad (17)$$

The variables T(t), c, ϕ, and T(t-1) represent the current and previous trust levels, respectively, in the LSTM and GRU, with ϕ representing the coefficient that indicates the influence of the prior trust level on the current one. Ǫ represents the influence of the past error term on the current trust levels through the moving average coefficient, while e(t-1) shows the gap between the projected and actual trust steps through the lagged error term. At each time step, the difference between the expected and observed levels of trust is captured by the current error term e(t). The DPO model, which will be covered in the section that follows, is responsible for selecting the LSTM weight optimally.

3.4.1. Doctor and Patient Optimization (DPO)

This section presents the Doctor besides Patient Optimisation (DPO) method, which is designed to address optimisation issues related to LSTM weight selection. Designing DPO involves simulating the processes of a patient's therapy. Vaccination, medication administration, and surgery comprise the three stages of the optional practice. The populace is vaccinated initially in this approach to prevent infection. Medication prescriptions are filled in the second stage. Patients with life-threatening conditions get surgery in the third and final stage. People requiring medical attention make up the DPO population. Equation (18) specifies this group of patients.

$$\mathrm{P} = \begin{bmatrix} \mathrm{P_1} \\ \vdots \\ \mathrm{P_i} \\ \vdots \\ \mathrm{P_N} \end{bmatrix} = \begin{bmatrix} \mathrm{p_1^1} & \cdots & \cdots & \cdots & \mathrm{p_1^m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \mathrm{p_i^1} & \cdots & \mathrm{p_i^d} & \cdots & \mathrm{p_i^m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \mathrm{p_N^1} & \cdots & \cdots & \cdots & \mathrm{p_N^m} \end{bmatrix} \quad (18)$$

Where P is the patient's populace, Pi is the ith patient, p_i^d the patient is denoted as N, which represents the total number of patients (population), and m is the total quantity of the variable. This populace is treated and updated in three stages. The required data in this procedure are calculated using Equations (19) and (22).

$$\mathrm{dosage_i} = 2 - \frac{\mathrm{F_i^n}}{\mathrm{F_{best}^n}} \quad (19)$$

$$\mathrm{F_i^n} = \frac{\mathrm{fit_i - f_{worst}}}{\sum_{j=1}^{N}(\mathrm{fit_i - f_{worst}})} \quad (20)$$

$$\mathrm{f_{worst}} = \max(\mathrm{fit})\ \&\mathrm{P_{worst}} = \mathrm{P(location(f_{worst}))} \quad (21)$$

$$\mathrm{f_{best}} = \min(\mathrm{fit})\&\mathrm{P_{best}} = \mathrm{P(location(f_{best}))} \quad (22)$$

Here, $dosage_i$ is the dosage of inoculation or the patient F_i^n is the regularised fitness of the ith enduring, F_{best}^n is the standardised finest patient, f_{worst} is the patient, f_{best} is the function patient, P_{worst} is the position of the nastiest patient, besides P_{best} It is the site of the finest patient.

3.4.1.1. Phase A: Vaccination

A significant step in the civic health procedure is vaccination. This equation is shown by Equations (23) and (24).

$$\mathrm{V_i^d} = \mathrm{rand} \times \left(\mathrm{dosage_i} \times \mathrm{p_i^d} - \mathrm{p_{worst}^d}\right) \quad (23)$$
$$\mathrm{V_i^d} = \mathrm{rand} \times \left(\mathrm{dosage_i} \times \mathrm{p_i^d} - \mathrm{p_{worst}^d}\right) \quad (24)$$

Here, V_i^d is the ith patient, rand is an independent random variable uniformly distributed on [0, 1], and pd worst is the worst patient.

3.4.1.2. Phase B: Drug Administration

At this stage of the treatment procedure, the doctor determines which medications are most suitable for each patient based on their individual health needs and medical history. Eqs. (25) and (26) model the process of drug delivery.

$$\mathrm{d}_\mathrm{i}^\mathrm{d} = \mathrm{rand} \times \left(\mathrm{p}_\mathrm{best}^\mathrm{d} - \mathrm{dosage}_\mathrm{i} \times \mathrm{p}_\mathrm{i}^\mathrm{d}\right) \tag{25}$$

$$\mathrm{P}_\mathrm{i} = \begin{cases} \mathrm{P}_\mathrm{i} + \mathrm{d}_\mathrm{i}, & \mathrm{fit}(\mathrm{P}_\mathrm{i} + \mathrm{d}_\mathrm{i}) \leq \mathrm{fit}_\mathrm{i} \\ \mathrm{P}_\mathrm{i}, & \mathrm{else} \end{cases} \tag{26}$$

Here, d_i^d is the DTH drug the best for the patient.

3.4.1.3. Phase C: Surgery

Patients with severe diseases do not get enough relief from medicine and vaccines. The patient's condition will improve after surgery in these instances. Equation (27) models this stage of therapy.

$$\mathrm{P}_\mathrm{i} = \begin{cases} 0.6 \times \mathrm{P}_\mathrm{i} + 0.4 \times \mathrm{P}_\mathrm{best}, & \mathrm{F}_\mathrm{best}^\mathrm{n} - \mathrm{F}_\mathrm{i}^\mathrm{n} \geq 0.9\ \mathrm{F}_\mathrm{best}^\mathrm{n} \\ \mathrm{P}_\mathrm{i}, & \mathrm{else} \end{cases} \tag{27}$$

3.4.1.4. Implementation of DPO

Once the suggested DPO method has been designed, optimisation issues may be solved using it. In Algorithm 1, we see the DPO implementation in action.

Procedure 1. The pseudo code of DPO

Start DPO
$System\ tuning\ and\ parameters\ determination.$
$Formation\ of\ the\ initial\ population\ of\ patients: P.$
$For\ iteration = 1: iteration\ max$
$Fitness\ function\ evaluation.$
$Updating\ f_{worst}\ and\ P_{worst}\ based\ (21).$
$Updating\ f_{best}\ and\ P_{best}\ based\ (22).$
$Updating\ F_i^n\ based\ (20).$
$For\ i = 1: N$
$Updating\ dosage_i\ based\ (19).$
$Updating\ P_i\ based\ phase\ a.$
$Updating\ P_i\ based\ phase\ b.$
$Updating\ P_i\ based\ phase\ c.$
$End\ for\ i$
$Saving\ f_{best}\ and\ P_{best}.$
$End\ for\ iteration$
$Return\ best\ solution.$
End DPO

4. Result and Discussion

4.1. Evaluation Metrics

In this unit, the four measurement systems used in the study are presented.

4.1.1. Mean Absolute Percentage Error

The performance of prediction algorithms is often measured by the Mean Absolute Percentage Error (MAPE). MAPE, which often displays accuracy as a percentage, is another method for evaluating the efficacy of machine learning forecasting systems. You may see its formula in equation (28).

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \times 100 \quad (28)$$

where A_t is the actual value and F_t is the estimated value of the alteration among those that are divided by A_t. The total value of each predicted value is summed and then divided by the total dataset size. The last step is to multiply by 100 to get the percentage mistake.

4.1.2. Mean Absolute Error

The MAE is a metric for comparing two numbers that represent the discrepancy between them. When calculating MAE, the disparity between predicted and actual values is averaged. Regression analysis in learning often uses MAE as a measure of prediction error. The equation that shows the formula is given in equation (29).

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|A_t - F_t| \quad (29)$$

where A_t is the true value besides F_t is the forecast value. In the formulation, the total charge of the difference among those is allocated by n (the sum of examples), in addition to being totalled for every forecasted value.

4.1.3. Relative Root Mean Square Error

The standard metric for regression error is RMSE. A prediction model's residuals, or prediction errors, reveal the dispersion of real-world values around the model and the distance between the model's predictions and the real values. The statistic shows the concentration of data around the model that best fits the data. Measured as the root-mean-squared error (RMSE), it is the sum of the squares of all discrepancies between forecasts and data. Like RMSE, RRMSE normalises the forecaster model's total squared error. Equation (30) displays the formula.

$$\text{RRMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\frac{A_t - F_t}{A_t}\right)^2} \quad (30)$$

where A_t is the empirical worth, F_t is the estimated value, and n is the sum of trials.

4.2. Validation Investigation of the Projected Model on the First Dataset

Table 1 presents the financial data 10 days in advance. In the analysis of the CNN model with the parameter set to 1, the model achieved an MAPE of 1.29, an MAE of 23.05, an rRMSE of 0.0235, and an MSE of 4948.07. Then, the RNN model, with a parameter of 150, attains a MAP of 0.92, MA of 15.80, and RMS of 0.0142, corresponding to 1403.

Table 1: Diversified financials 10 days ahead

Prediction Representations	Error Measures				
	Parameters	**MAPE**	**MAE**	**rRMSE**	**MSE**
	ntrees				
CNN	1	1.29	15.09	0.0235	4959.07
RNN	150	0.92	19.19	0.0142	1403.24
GRU	200	0.92	15.51	0.0143	1280.91
LSTM	250	0.91	15.09	0.0152	912.53
GRU-LSTM	300	1.02	19.19	0.0203	4312.09
GRU-LSTM-DPO	350	0.88	14.86	0.0120	804.97

Then, the GRU model with a parameter of 200 attained an MAPE of 0.92, an MAE of 15.51, an rRMSE of 0.0141, and an MSE of 1290.91. Then, the LSTM model with a parameter of 250 achieved an MAPE of 0.91, an MAE of 15.09, an rRMSE of 0.0132, and an MSE of 912.51. Then, the GRU-LSTM model, with a parameter of 300, attained an MAPE of 1.02, an MAE of 19.19, and an rRMSE of 0.0203, corresponding to an MSE of 4312.09. Then, the GRU-LSTM-DPO model, with a parameter of 350, attained an MAPE of 0.88, an E of 0.86, and an RRMan SE of 0120, along with an E of 804.97 co.

In Table 2 above, it is described that the financials are 15 days. In the analysis of the CNN model, the parameter was set to 1, and the model attained an MAPE of 1.52, an MAE of 25.93, and an RMSE of 0.0250, corresponding to an MSE of 2893.8. Then, the RNN model with a parameter of 150 attained an MAPE of 1.10, an MAE of 18.31, an rRMSE of 0.0160, and an MSE of 1320.55. Then, the GRU model, with a parameter of 200, attained an MAPE of 1.12, an MAE of 18.39, and an rRMSE of 0.0171, corresponding to an MSE of 1322.25.

Table 2: Diversified financials 15 days ahead

Prediction Models	Error Measures				
	Parameters	MAPE	MAE	rRMSE	MSE
CNN	1	1.52	25.93	0.0250	2893.88
RNN	150	1.10	18.31	0.0161	1320.55
GRU	200	1.12	18.39	0.0173	1322.25
LSTM	250	1.11	19.56	0.0164	1687.85
GRU-LSTM	300	1.14	19.51	0.0199	1781.15
GRU-LSTM-DPO	350	1.14	19.81	0.0162	1724.65

Then, the LSTM model with a parameter of 250 achieved an MAPE of 1.11, an E of 0.56, an RMSE of 0.164, and an MSE of 1687.85. Then, the GRU-LSTM model, with a parameter of 300, achieved an MAPE of 1.14, an MAE of 19.51, an rRMSE of 0.0199, and an MSE of 1781.31, respectively. Then, the GRU-LSTM-DPO model, with a parameter of 350, achieved an MAPE of 1.14, an MAE of 19.81, an rRMSE of 0.0162, and an MSE of 1724.65, respectively.

Table 3: Diversified financials 20 days ahead

Prediction Models	Error Measures				
	Parameters	MAPE	MAE	rRMSE	MSE
	ntrees				
CNN	1	1.66	28.96	0.0298	4715.43
RNN	150	1.45	24.00	0.0215	2146.32
LSTM	250	1.39	23.91	0.0198	2494.78
GRU	200	1.47	24.46	0.0216	2317.71
GRU-LSTM	300	1.35	23.05	0.0142	2002.51
GRU-LSTM-DPO	350	1.45	24.12	0.0202	2056.23

Table 3 above presents the financial data 20 days in advance. In the analysis of the CNN model, with the parameter set to 1, the model attained the following results: MAPE of 1.66, MAE of 28.94, rRMSE of 0.0298, and MSE of 4715.42. Then, the RNN model with a parameter of 150 attained an MAPE of 1.45, an MAE of 24.00, an rRMSE of 0.0215, and an MSE of 2146.32. Then, the LSTM model with a parameter of 250 attained an MAPE of 1.39, an MAE of 23.91, an rRMSE of 0.0198, and an MSE of 2494.78. Then, the GRU model with a parameter of 200 attained an MAPE of 1.47, an MAE of 24.46, an rRMSE of 0.0216, and an MSE of 2317.71. Then, the GRU-LSTM model with a parameter of 300 attained an MAPE of 1.35, an MAE of 23.05, an rRMSE of 0.0142, and an MSE of 2002.51. Then, the GRU-LSTM-DPO model, with a parameter of 350, attained an MAPE of 1.45, an MAE of 24.12, and an rRMSE of 0.0202, corresponding to an MSE of 2056.23.

4.3. Validation Analysis of the Proposed Model on the Second Dataset

Table 4 above describes the financial data 10 days ahead. In the analysis of the CNN model with the parameter set to 1, the model achieved an MAPE of 2.09, an MAE of 34.00, an rRMSE of 0.0382, and an MSE of 5,129.32. Then, the RNN model with a parameter of 150 attained an MAPE of 1.88, an MAE of 31.47, an rRMSE of 0.0283, and a corresponding MSE of 3219.3.

Table 4: Diversified financials 10 days ahead in the analysis

Prediction Models	Error Measures				
	Parameters	MSE	MAE	MAPE	rRMSE
CNN	1	5129.32	34.00	2.09	0.0382
RNN	150	3219.30	31.47	1.88	0.0283
GRU	200	3246.80	31.37	1.86	0.0281
LSTM	250	2122.81	25.63	1.58	0.0251
GRU-LSTM	300	3356.01	28.01	1.74	0.0322
GRU-LSTM-DPO	350	3600.53	31.07	1.77	0.0257

Then, the GRU model, with a parameter of 200, attained an MAPE of 1.86, an MAE of 31.36, and an rRMSE of 0.0279, corresponding to an MSE of 3246.80. Then, the LSTM model with a parameter of 250 attained an MAPE of 1.58, an MAE of 25.63, an rRMSE of 0.0251, and an MSE of 2122.81. Then, the GRU-LSTM model, with a parameter of 300, and the PE as 1.74, AE as 28.00, and SE as 0.00, as well as an MSE of 33.00. Then, the GRU-LSTM-DPO 350 model achieved an MAPE of 1.77, an MAE of 31.07, an RMSE of 0.0257, and an MSE of 3600.53 (Table 5).

Table 5: Expanded financial data 15 days ahead

Prediction Models	Error Measures				
	Parameters	MAPE	MAE	rRMSE	MSE
LSTM	250	1.83	28.83	0.0301	3146.31
GRU-LSTM	300	1.97	35.95	0.0390	8759.44
CNN	1	2.28	41.29	0.0451	11,051.93
RNN	150	2.24	37.61	0.0349	4997.20
GRU	200	2.24	37.28	0.0349	4755.32
GRU-LSTM-DPO	350	2.03	35.37	0.0305	5534.65

In Table 6 above, the financial data is 20 days ahead. In the analysis of the LSTM model, the parameter was set to 250, and the model attained an MAPE of 1.83, an MAE of 28.83, a 0.0301, and an rRMSE of 3146.31, respectively. Then, the GRU-LSTM model with a parameter of 300 attained an MAPE of 1.97, an MAE of 35.95, an rRMSE of 0.0390, and an MSE of 8759.44. Then, the CNN model, with the parameter set to 1, attained an MAPE of 2.28, an MAE of 41.29, and an rRMSE of 0.0451, corresponding to an MSE of 11,051.9. Then, the RNN model, with a parameter of 150, attained an MAPE of 2.24, an MAE of 37.61, an rRMSE of 0.0349, and an MSE of 4997.20, respectively. Then, the GRU model, with a parameter of 200, attained an MAPE of 2.24, anE as of.28, and rRMan SE as of0349 and rRM5.32 co, Then, the GRU-LSTM-DPO model, with a parameter of 350, attains the following results: the PE as 2, the AE as 35, the SE as 0.0, and the MSE as 55.34.65, correspondingly.

Table 6: Expanded financial data 20 days ahead

Prediction Representations	Error Measures				
	Parameters	MAPE	MAE	rRMSE	MSE
	ntrees				
CNN	1	2.80	49.12	0.0571	14,227.36
RNN	150	2.56	42.43	0.0388	5916.29
GRU	200	2.57	42.68	0.0393	6008.33
LSTM	250	2.01	33.25	0.0309	4340.29
GRU-LSTM	300	2.17	39.10	0.0385	8573.37
GRU-LSTM-DPO	350	2.30	39.30	0.0358	6406.16

Table 6 describes financial data from 20 days ago. In CNN perfect, the parameter is 1, and the model attained an MAPE of 2.80, an E of 0.12, and an rRMan SE of 0.571, corresponding to an E of 14,227.06. Then, the RNN model with a parameter of 150 achieved an MAPE of 2.56, an MAE of 42.43, an RMSE of 0.0388, and an MSE of 5916.19. Then, the GRU model, with a parameter of 200, achieved the highest APE.57, MAan E asof2.66, and rRan MSE asof.0393, ancorresponding to an SE of 6008.33 c Then, the LSTM model, with a parameter of 250, and the OFE as one and OFAE as 33.25, and an rRMSE of 0.0, as well as an MSE of 43.40.09, correspondingly. Then, the GRU-LSTM model, with a parameter of 300, and the PE as 0.17, AE as 39.10, and SE as 0.0, along with an MSE of 85. Then, the GRU-LSTM-DPO model, with a parameter of 350, attained an MAPE of 2.30, an MAE of 39.30, an rRMSE of 0.0358, and an MSE of 6406.16, respectively.

5. Conclusion

Predicting the future value of an asset (such as a stock) is crucial for anyone involved in financial investment. The issue's intricacy makes accurate prediction hard. This study proposes VARMA, a hybrid of GRU besides LSTM, to forecast stock prices using the CSI 300 index as a starting point. The goal is to improve prediction accuracy. A one-dimensional CNN is used for feature extraction, with DPO selecting the LSTM weight appropriately. A comparison of the suggested model with LSTM, GRU, RNN, and CNN reveals that it outperforms all four in predicting stock price indices. Using 12 stocks selected from China and abroad, the four representations mentioned above were employed to estimate stock prices or indices, thereby conducting a model stability study of the proposed model. Once complete, the test results demonstrate that the suggested model can accurately forecast stock market indices worldwide, not just in China. This suggests that the model may be used in a wide range of situations. Nevertheless, this paper's assessment index is based on stock trading data, and other market variables impact stock prices and indices. Therefore, the suggested model also has limitations, primarily because of its structural similarity to previous models. Additional effort, such as incorporating stock data into the index, merging the most recent models, or creating new and challenging models, may be undertaken to further enhance forecast accuracy.

Acknowledgment: N/A

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request. All authors confirm that data can be shared in accordance with ethical and privacy requirements.

Funding Statement: This manuscript and the associated research were carried out without any external financial support or funding. All authors contributed voluntarily to the development of this work.

Conflicts of Interest Statement: The authors declare that they have no conflicts of interest related to this work. This study represents an original collaborative contribution, and all sources, citations, and references have been properly acknowledged.

Ethics and Consent Statement: This study was conducted in accordance with established ethical guidelines. All authors confirm that informed consent was obtained from all participants involved in the research.

Reference

1. D. Kumar, P. K. Sarangi, and R. Verma, "A systematic review of stock market prediction using machine learning and statistical techniques," *Materials Today: Proceedings*, vol. 49, no. 8, pp. 3187–3191, 2022.
2. M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, "Machine learning techniques and data for stock market forecasting: A literature review," *Expert Systems with Applications*, vol. 197, no. 7, p. 116659, 2022.
3. F. J. G. Peñalvo, T. Maan, S. K. Singh, S. Kumar, V. Arya, K. T. Chui, and G. P. Singh, "Sustainable Stock Market Prediction Framework Using Machine Learning Models," *International Journal of Software Science and Computational Intelligence*, vol. 14, no. 1, pp. 1–15, 2022.
4. M. M. Akhtar, A. S. Zamani, S. Khan, A. S. A. Shatat, S. Dilshad, and F. Samdani, "Stock market prediction based on statistical data using machine learning algorithms," *Journal of King Saud University - Science*, vol. 34, no. 4, p. 101940, 2022.
5. M. Bansal, A. Goyal, and A. Choudhary, "Stock market prediction with high accuracy using machine learning techniques," *Procedia Computer Science*, vol. 215, no. 12, pp. 247–265, 2022.
6. P. Soni, Y. Tewari, and D. Krishnan, "Machine Learning approaches in stock price prediction: A systematic review," *Journal of Physics: Conference Series*, vol. 2161, no. 1, p. 012065, 2022.
7. P. Chhajer, M. Shah, and A. Kshirsagar, "The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction," *Decision Analytics Journal*, vol. 2, no. 3, p. 100015, 2022.
8. U. R. Saxena, P. Sharma, and G. Gupta, "Comprehensive study of machine learning algorithms for stock market prediction during COVID-19," *Journal of Computers, Mechanical and Management*, vol. 1, no. 2, pp. 14–21, 2022.
9. M. Leippold, Q. Wang, and W. Zhou, "Machine learning in the Chinese stock market," *Journal of Financial Economics*, vol. 145, no. 2, pp. 64–82, 2022.
10. Q. Liu, Z. Tao, Y. Tse, and C. Wang, "Stock market prediction with deep learning: The case of China," *Finance Research Letters*, vol. 46, no. 5, p. 102209, 2022.
11. S. Albahli, A. Irtaza, T. Nazir, A. Mehmood, A. Alkhalifah, and W. Albattah, "A Machine Learning Method for Prediction of Stock Market Using Real-Time Twitter Data," *Electronics*, vol. 11, no. 20, p. 3414, 2022.
12. A. Khattak, A. Khan, H. Ullah, M. U. Asghar, A. Arif, F. M. Kundi, and M. Z. Asghar, "An efficient supervised machine learning technique for forecasting stock market trends," Information and Knowledge in Internet of Things, *Springer,* Cham, Switzerland, 2022.
13. V. V. Prasad, S. Gumparthi, L. Y. Venkataramana, S. Srinethe, R. M. Sruthi Sree, and K. Nishanthi, "Prediction of stock prices using statistical and machine learning models: a comparative analysis," *The Computer Journal*, vol. 65, no. 5, pp. 1338–1351, 2022.
14. H. Napitupulu, A. Sambas, A. Murniati, and V. A. Kusumaningtyas, "Artificial Neural Network-Based Machine Learning Approach to Stock Market Prediction Model on the Indonesia Stock Exchange During the COVID-19," *Engineering Letters*, vol. 30, no. 3, p. 9, 2022.
15. S. Saranya, M. Mahendhiran, and S. Sathya, "Stock market prediction analysis using machine learning techniques," *AIP Conference Proceedings*, vol. 2742, no. 1, p. 020025, 2024.

16. M. Lu and X. Xu, “TRNN: An efficient time-series recurrent neural network for stock price prediction,” *Information Sciences*, vol. 657, no. 2, p. 119951, 2024.
17. Y. Liu, S. Huang, X. Tian, F. Zhang, F. Zhao, and C. Zhang, “A stock series prediction model based on variational mode decomposition and dual-channel attention network,” *Expert Systems with Applications*, vol. 238, no. 3, p. 121708, 2024.
18. E. T. Sivadasan, N. Mohana Sundaram, and R. Santhosh, “Stock market forecasting using deep learning with long short-term memory and gated recurrent unit,” *Soft Computing*, vol. 8, no. 1, pp. 3267–3282, 2024.
19. S. Deng, Y. Zhu, Y. Yu, and X. Huang, “An integrated approach of ensemble learning methods for stock index prediction using investor sentiments,” *Expert Systems with Applications*, vol. 238, no. 3, p. 121710, 2024.
20. H. Qian, H. Zhou, Q. Zhao, H. Chen, H. Yao, J. Wang, and J. Zhou, “MDGNN: Multi-Relational Dynamic Graph Neural Network for Comprehensive and Dynamic Stock Investment Prediction,” *arXiv preprint* arXiv:2402.06633, 2024. Available: https://arxiv.org/abs/2402.06633 [Accessed by 24/10/2024].
21. Y. Zhang, “Analysis and Prediction of China Stock Market Return Based on BP Neural Network,” in Internet Finance and Digital Economy, *World Scientific Publishing*, Singapore, 2024.
22. S. Mukherjee, B. Sadhukhan, N. Sarkar, D. Roy, and S. De, “Stock market prediction using deep learning algorithms,” *CAAI Transactions on Intelligence Technology*, vol. 8, no. 1, pp. 82–94, 2023.
23. S. Verma, S. P. Sahu, and T. P. Sahu, “Discrete wavelet transform-based feature engineering for stock market prediction,” *International Journal of Information Technology*, vol. 15, no. 2, pp. 1179–1188, 2023.
24. P. R. Patil, D. Parasar, and S. Charhate, “Wrapper-Based Feature Selection and Optimization-Enabled Hybrid Deep Learning Framework for Stock Market Prediction,” *International Journal of Information Technology & Decision Making*, vol. 23, no. 1, pp. 1–26, 2023.
25. A. Bhambu, “Stock Market Prediction Using Deep Learning Techniques for Short and Long Horizon,” *in Soft Computing for Problem Solving: SocProS 2022*, Mandi, India, 2023.
26. TSETMC, “Tehran Stock Exchange Market Website.” 2025, Available: www.tsetmc.com, [Accessed by 29/04/2025].
27. J. Zhang, L. Ye, and Y. Lai, “Stock Price Prediction Using CN-BiLSTM-Attention Model,” *Mathematics*, vol. 11, no. 9, p. 1985, 2023.
28. M. Dehghani, M. Mardaneh, J. M. Guerrero, O. P. Malik, R. A. Ramirez-Mendoza, J. Matas, and L. Parra-Arroyo, “A new ‘Doctor and Patient’ optimization algorithm: An application to energy commitment problem,” *Applied Sciences*, vol. 10, no. 17, p. 5791, 2020.