Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 12, No. 3, 2024, 393-408

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Advanced Stock Price Forecasting using a 1D-CNN-GRU-LSTM Model

Fatemeh Moodi, Amir Jahangard Rafsanjani*, Sajjad Zarifzadeh and Mohammad Ali Zare Chahooki

Department of Computer Engineering, Yazd University, Yazd, Iran.

Article Info

Article History:
Received 05 August 2024
Revised 27 August 2024
Accepted 09 October 2024

DOI:10.22044/jadm.2024.14831.2581

Keywords:
Hybrid Deep Neural Network, 1D-CNN-GRU-LSTM Network, Stock Price Movement Forecasting, Tweet Sentiment Analysis, Technical Indicators

**Corresponding author: jahangard@yazd.ac.ir (A. Jahangard Rafsanjani).*

Abstract

This article proposes a novel hybrid network integrating three distinct architectures – CNN, GRU, and LSTM – to predict stock price movements. Here Combining Feature Extraction and Sequence Learning along with Complementary Strengths can improve Predictive Performance. CNNs can effectively identify short-term dependencies and relevant features in time series, such as trends or spikes in stock prices. GRUs are designed to handle sequential data. They are particularly useful for capturing dependencies over time while being computationally less expensive than LSTMs. In the hybrid model, GRUs help maintain relevant historical information in the sequence without suffering from vanishing gradient problems, making them more efficient for long sequences. LSTMs excel at learning long-term dependencies in sequential data, thanks to their memory cell structure. By retaining information over longer periods, LSTMs in the hybrid model ensure that important trends over time are not lost, providing a deeper understanding of the time series data. The novelty of the 1D-CNN-GRU-LSTM hybrid model lies in its ability to simultaneously capture short-term patterns and long-term dependencies in time series data, offering a more nuanced and accurate prediction of stock prices. The data set comprises technical indicators, sentiment analysis, and various aspects derived from pertinent tweets. Stock price movement is categorized into three categories: Rise, Fall, and Stable. Evaluation of this model on five years of transaction data demonstrates its capability to forecast stock price movements with an accuracy of 0.93717. The improvement of proposed hybrid model for stock movement prediction over existing models is 12% for accuracy and F1-score metrics.

1. Introduction

Predicting stock prices has piqued the interest of scientists across diverse fields, including financial engineering, economics, operations research, statistics, and machine learning (ML). Despite significant efforts over the past few decades, precise prediction of stock prices and their movements remains challenging, even with the utilization of advanced ML methods [1]. The analysis of financial time series data presents several challenges, including non-stationarity, nonlinearity, and high volatility [2], prompting researchers to continuously seek more accurate and refined models for analysis. Additionally, it's important to note that price fluctuations in stock markets are influenced also by nonlinear factors such as unexpected events, investor behavior, and political considerations [3-6]. This has led researchers to continuously explore more advanced models for effective time series analysis and prediction [7].

Neural networks have been applied to these challenges, showing promise in stock market trend prediction. However, despite their benefits, traditional neural networks may still not accurately predict financial markets [8]. Specifically, shallow neural networks cannot effectively represent data features, resulting in reduced analysis accuracy. To overcome this issue, deep networks are utilized, which can analyze data features and dependencies and capture long-term trends and fluctuations [9]. Due to DL's high analytical power, its application in the capital market has experienced significant growth. Among the DL models utilized in financial research, Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM) have demonstrated promising outcomes [10-12]. However, forecasting volatile and highly stochastic time series data remains difficult, even with these models. This calls for more refined methods that can better handle financial data's intricate nature.

Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRU) [44], and LSTM models have emerged as specialized tools for time series data. While RNNs excel in handling sequential information, they struggle with long-term dependencies, leading to issues like vanishing gradients [13]. GRU and LSTM models address these issues by incorporating mechanisms to retain long-term memory, though each has distinct strengths —LSTM is more powerful in managing long-term dependencies, while GRU is more computationally efficient.

Each of these models —RNN, CNN, LSTM, and GRU— offers unique advantages. RNNs are effective for sequential tasks like language translation, CNNs excel in feature extraction from images, and LSTM and GRU are designed to overcome limitations in learning long-term dependencies in time series data. This has motivated the development of hybrid neural networks that combine these strengths to enhance prediction accuracy. Consequently, these hybrid models can leverage the advantages of individual models to achieve better forecasting results.

The present study endeavors to introduce a novel hybrid model for predicting stock price movements that can simultaneously capture short-term patterns and long-term dependencies in time series data, offering a more nuanced and accurate prediction of stock prices. To achieve this goal, we utilize four stock price indices for practical evaluation and compare the performance of our proposed model with non-hybrid alternatives. Our novel hybrid model offers several key features:

1. Introduction of an efficient deep hybrid system designed to predict stock price movements with high accuracy.
2. Utilization of sentiment analysis on tweets related to various company stocks, focusing on aspects such as retweet and like counts, to enhance prediction capabilities.
3. Calculation of the percentage increase or decrease in stock prices within specified rolling windows.
4. Computation of return rates for the four stock price indices over specific periods, followed by a comparison of actual returns with predicted returns.
5. Comparison of the performance of our proposed hybrid model with previous models.

Experimental results indicate that our proposed hybrid model outperforms individual models in forecasting stock movement directions.

The practical implications of the proposed model are significant in areas such as risk management, decision-making, strategy optimization, portfolio management, and algorithmic trading, particularly when incorporating sentiment analysis to capture market trends influenced by investor sentiment.

In conclusion, the hybrid 1D-CNN-GRU-LSTM model effectively combines the strengths of individual models, yielding low MSE values and high accuracy rates, making it a valuable tool for real-world financial forecasting and decision-making processes.

2. Related works

Faghihi Nezhad and Minaei Bidgoli propose an ensemble learning (EL) model that combines learners based on intelligence and metaheuristic optimization methods [14]. Their two-stage structure accounts for the direction of stock prices, utilizing genetic algorithm (GA) and particle swarm optimization (PSO) techniques to optimize the aggregation results of base learners. Assessment findings from stock market data demonstrate that their approach achieves higher accuracy compared to other existing models.

Zhang et al. introduce an innovative system for predicting stock price trends, capable of predicting not only trends but also the rate of rise or fall within specific durations [15]. The system operates entirely through machine learning processes, generating training samples from original transaction data and constructing prediction models without human intervention. Using a sliding window, stock data is divided into clips, classified into primary categories based on close price shapes, and further categorized for different levels of rise and fall rates. Training sets are derived from these clips, addressing issues of imbalanced category distribution. The system is evaluated using stock data from the Shenzhen Growth Enterprise Market over seven years.

Chen proposes a CNN-GRU-attention model to predict long-term stock price trends, employing three data decomposition techniques for preprocessing and selecting the most optimal approach [16]. Comparative analysis identifies CEEMDAN as the optimal preprocessing technique. Six models are tested on the validation set, with the CNN-GRU-Attention model exhibiting superior alignment with actual values and significantly enhancing prediction quality. Evaluation based on performance indicators such as MAE, RMSE, and R reveals the CNN-GRU-attention model's highest prediction accuracy.

Song and Choi propose three hybrid models based on Recurrent Neural Networks (RNN) – CNN-LSTM, GRU-CNN, and ensemble models – to forecast one-time-step and multi-time-step closing prices of companies [17]. Their aim is to forecast the closing prices of three stock markets. Results reveal that the proposed models significantly outperform benchmark models, evaluated using MSE and MAE metrics. They introduce a new feature called "medium" to represent the average of high and low prices, which enhances model performance. Empirical findings show that the proposed models surpass benchmark models in 48.1% and 40.7% of instances for single-time-step forecasting, and in 81.5% of instances for multi-time-step forecasting. Additionally, they introduce a new feature to mitigate the impact of the highest and lowest prices, further improving model performance.

Karim et al. introduce an innovative hybrid deep learning architecture combining bidirectional long short-term memory (BiLSTM) and GRU networks [18]. They compare the performance of individual modules such as LSTM, Bi-LSTM, GRU, and traditional NN for stock price forecasting, using NIFTY-50 stock market price data. Results demonstrate the superior performance of the proposed hybrid model compared to individual models.

The study conducted by Cao et al. utilized a combination model consisting of a hybrid neural network (CNN) and SVM Machine (SVM) for forecasting stock indices [9]. Results showed that the hybrid neural network effectively handled both continuous and classified variables. Initially, a CNN model was constructed for stock index forecasting, and the impact of model parameters on prediction results was examined. This led to the development of a CNN-SVM model for stock index prediction, which was empirically analyzed and confirmed to be feasible and effective.

In another study, a hybrid model named FS-CNN-BGRU was proposed for stock prediction [19]. This model integrates feature selection (FS), CNN for feature extraction, and bidirectional gated recurrent unit (BGRU) for processing temporal data sequences. Data from four key indices of China spanning from January 1991 to December 2020 were used, with an 80:20 ratio for dividing the data into training and testing sets. Assessment metrics such as MAPE (Mean Absolute Percentage Error) and R2 indicated that the proposed FS-CNN-BGRU model outperformed other models.

Predicting stock prices has long been challenging due to nonlinearity and significant volatility within financial time series. Recent advancements in DL techniques, including LSTM and CNN models, have improved the analysis of such data. Rezaei et al. proposed hybrid algorithms, CEEMD-CNN-LSTM and EMD-CNN-LSTM, which leverage empirical mode decomposition algorithms (EMD and CEEMD) for decomposing time series into different frequency spectra [20]. These algorithms were applied to one-step-ahead prediction and demonstrated improved prediction accuracy compared to alternative approaches. Empirical findings indicated that the CEEMD-CNN-LSTM algorithm yielded more precise outcomes than CEEMD-LSTM and LSTM algorithms, and similarly, EMD-CNN-LSTM outperformed EMD-LSTM and LSTM algorithms. Incorporating CNN in the proposed models effectively enhanced model performance during data training in conjunction with LSTM. Additionally, the decomposed algorithms showed superior performance compared to undecomposed models, highlighting the benefits of CEEMD and EMD models.

Gite et al. employed LSTM and eXplainable Artificial Intelligence (XAI) in their research on stock price prediction [21]. They utilized LSTM-CNN for predicting opening prices and XAI for interpreting the model. Data extracted from the National Stock Exchange (NSE) and news headlines from Pulse [22] were utilized. The data was split into training and testing sets using an 80:20 ratio. The model was trained over one hundred epochs, achieving an overall accuracy of 74.76% and a loss of 0.1693 over the fifteen epochs. The absolute deviation between the observed and projected values, multiplied by 100, ranged from 0.15 to 0.73.

Ji et al. introduced a methodology to enhance technical indicators by employing denoising techniques on various wavelet basis functions and selecting the most optimal feature from the feature set [23]. The utilization of denoised stock price information in this study enabled a more effective calculation of the technical indicators. The data set included four stock markets, and a total of 18 technical indicators were assessed. Time windows of varying sizes, specifically 3, 5, 10, 15, 30, 45, and 60, were utilized. The findings indicated an enhancement in model accuracy after implementing denoising techniques.

Peng et al. examined a set of 124 technical analysis indicators. Three selected features were utilized to reduce the size of the feature set and evaluate the predictive capabilities of these features [24]. Deep neural networks with varying structures and regularization parameters were employed. Daily data spanning from 2008 to 2019 from seven global market indicators were utilized. Sequential forward selection (SFS) and sequential backward selection (SBS) procedures were performed using logistic regression as the fitting model and Akaike criteria to assess the features. The data set included stock data from seven companies, spanning from 2008 to 2019. It was divided into two sections for feature selection and prediction. Wrapper feature selection methods, specifically sequential forward floating selection (SFFS), tabu search (TS), and Lasso, were employed for feature selection. The selected features were then used to train the models using a second section of data. These models consisted of 3, 5, and 7 hidden layers, with the activation function being the sigmoid function. The networks were trained using the Adam optimization algorithm, with 400 epochs and mini-batches with a size of 128. The precision of prediction for all seven markets and 48 compositions ranged from 50% to 65%.

Hosseinzadeh and Haratizadeh proposed a framework implemented to predict the movement direction of various indices such as RUSSELL, DJI, NYSE, NASDAQ, and S&P 500, based on CNN [25]. This prediction was made using different sets of initial variables. The main objective of the 2D-CNNpred framework is to establish a general model that can link the historical data of a market's forthcoming variations. Hence, to derive the intended mapping function, the model must undergo training with data from various markets. The 2D-CNNpred approach adheres to this overarching principal methodology but also incorporates various other variables and the market history as input data. In this framework, all the aforementioned information is compiled and inputted as a 2D tensor into a specifically designed CNN, hence referred to as 2D-CNNpred. In contrast, the alternative approach, 3D-CNNpred, argues that different models are required for prediction in distinct markets, but these models can utilize historical data from multiple markets. Unlike 2D-CNNpred, 3D-CNNpred does not engage in training a singular prediction model for each market based on its historical data but rather extracts characteristics from the historical data. The 2D-CNNpred and 3D-CNNpred data consist of layers that aim to merge the extracted features in the initial layer and produce more complex features that summarize the data over a specific period. During the prediction phase, a flat operation is applied to the features generated in the preceding layers, converting them into a one-dimensional vector. Subsequently, this is input into a fully connected layer which transforms the characteristics into a forecast.

Based on the reviews of the implemented hybrid models in the studies mentioned above, it is evident that most of them exhibit high error rates and struggle to achieve high accuracy for prediction. The primary goal of these endeavors is to reduce the error rate in accurately forecasting stock prices for the short-term future. Based on this observation, it becomes clear that a hybrid deep neural network may outperform an individual deep neural network. Therefore, our main objective is to achieve more accurate stock price movement prediction by proposing a new hybrid DL model that is inherently hybrid.

3. Methodology

3.1. CNN

CNNs, a type of deep feedforward network, surpass human performance and are capable of effectively learning image features, allowing for the accurate classification of new data using only a limited number of samples from the training set [26]. The structure of a CNN consists of three layers: input, convolutional, pooling, and fully-connected layers [27,28]. Forward propagation occurs through the convolution process, extracting features from the input data. The pooling layer intermittently intervenes between the convolution layers to reduce the input data dimension. The fully connected layer establishes connections across different levels. Through the iterative process of convolution or pooling layers, CNNs effectively learn the inherent characteristics of the input data, accurately predicting new data. The structure of a typical CNN is illustrated in Figure 1.

3.2. LSTM

RNNs are an advanced iteration of conventional feedforward neural networks, specifically designed to handle sequential data, which incorporate gates to retain prior inputs and effectively process data sequences [29]. Despite the theoretical capacity of RNNs to handle sequential data, they encounter practical challenges in maintaining long-term memory.

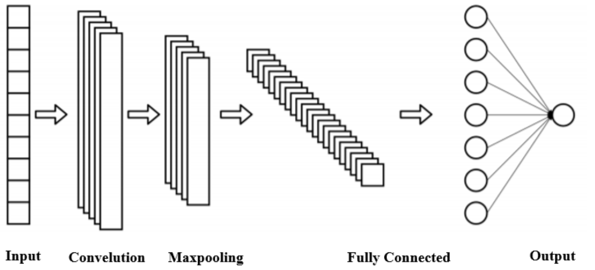

Figure 1. Structure of the CNN.

These challenges include the vanishing gradient problem, where RNNs struggle to retain information over long intervals, and the exploding gradient problem, where disproportionately high values are assigned to matrix weights during training [30]. To address these issues, LSTM models, which are enhanced RNN models with long and short-term memory capabilities, have been introduced. The LSTM architecture includes three gates (Figure 2): the input gate, output gate, and forget gate. By incorporating these gates, LSTM models can effectively capture long-term dependencies within a sequence. The forget gate (f_t) controls which information should be forgotten from the LSTM's memory, the input gate (i_t) determines whether new information should be added to the LSTM's memory, and the output gate (o_t) determines whether the LSTM's current state should be outputted.

The LSTM equations as Eq. (1) to (6) are as follows.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot \tanh(C_t) \tag{6}$$

The candidate value at time t, represented as $\tilde{C}_t$, is computed based on the cell state C_t and the input vector x_t, using the element-wise Hadamard product symbolized by ⊙. The final output h_t is determined by the weight matrices W^* and bias vectors b^*. As mentioned, the gates play a crucial role in determining which information is forgotten and which is retained.

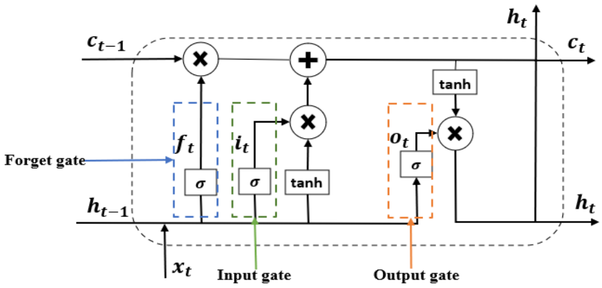

Figure 2. Structure of the LSTM.

3.3. GRU

The GRU, a simplified variant of the LSTM model with fewer parameters, employs only an update gate and a reset gate, unlike LSTM, which includes four gates. By combining the forget and input gates into a single update gate and reducing the number of tensor operations, GRU enables faster training compared to LSTM. While GRU requires fewer parameters during computation, simplifying the convergence process and facilitating model stability, LSTM demonstrates superior performance in large data sets. However, GRU excels at capturing temporal interdependence among time steps within time series data, addressing challenges such as gradient explosion more effectively [16].

By effectively combining cell and hidden states, GRU is well-suited for sequence learning tasks and mitigates issues like vanishing or exploding gradients encountered in vanilla RNNs when capturing long-term dependencies [31]. Furthermore, GRU tends to outperform LSTM when trained on limited data, while LSTM is more proficient at retaining information from longer sequences [32,33]. The GRU structure is depicted in Figure 3.

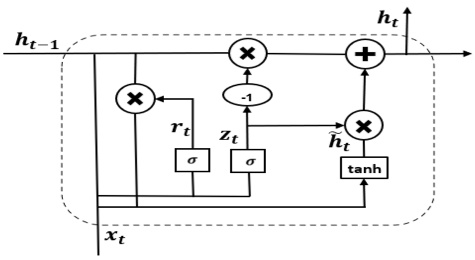

Figure 3. Structure of the GRU.

Eq. (7) to (10) below illustrate the update of memory cells at each hidden layer during each time step.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{7}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{8}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (9)$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t\, h_{t-1}, x_t]) \quad (10)$$

Where r_t is the reset gate, x_t represents the input, z_t is the update gate, h_t is the candidate activation, W_r , W_z , and W_h are the weight matrices.

3.4. Proposed 1D-CNN-GRU-LSTM model

Merging multiple forecasting models introduces bias, ultimately resulting in a reduction of variance and consequently leads to better performance compared to individual models [34]. The proposed hybrid model merges a 1D-CNN with a GRU and an LSTM, leveraging the strengths of each component to enhance prediction accuracy. The 1D-CNN layer effectively filters noise and identifies essential features from sequential data, reducing parameter complexity. The 1D-CNN-GRU-LSTM model comprises a 1D-CNN layer with 32 filters of size 3 and stride 1, activated by the Relu function, followed by a max-pooling layer with a pool size of 2. Subsequently, a GRU layer with 128 units activated by the tanh function and an LSTM layer with 64 units activated by Relu are incorporated. Additionally, a dense layer with 32 units activated by Relu and a final dense layer with 1 unit activated by Relu complete the model architecture. The block diagram of the proposed 1D-CNN-GRU-LSTM model is illustrated in Figure 4.

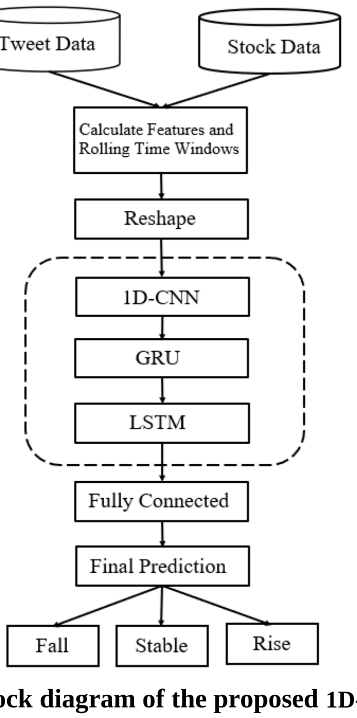

Figure 4. The block diagram of the proposed 1D-CNN-GRU-LSTM model

The primary objective of this study is to evaluate the effectiveness of a novel hybrid architecture that integrates multiple DL models for predicting stock price movements. In this hybrid model, data sets comprising tweets related to various companies, their corresponding stock prices, and calculated technical indicators are crucial for prediction accuracy. The model combines CNN, GRU, and LSTM models, leveraging the strengths of each to overcome individual model limitations and achieve superior performance in stock market analysis. The hybrid model operates in stages: initially, the CNN model processes input data and extracts features, which are then transferred to the GRU layer. The GRU layer, being simpler than LSTM, quickly learns sequential patterns and enhances network performance with less training time. Subsequently, the output from GRU is passed to the LSTM layer, which preserves temporal sequence information and conducts final analysis based on past stage data. Finally, the stock price movement is categorized into three classes.

Structure of proposed 1D-CNN-GRU-LSTM model is shown in Figure 5. The operations and equations of the proposed method with details used are as below:

In 1D-CNN layer; X is the input data to the 1D-CNN layer, and W_i be the filter weights with i ranging from 1 to 32. The output of the 1D-CNN layer is obtained by convolving the input X with each filter W_i and applying the Relu activation function that shown as Eq. (11).

$$Y_{cnn} = Relu\left(X * W_i + b_i\right) \quad (11)$$

where * denotes the convolution operation, and b_i is the biases.

Deep learning models do not need a separate step to extract, and CNN layers are able to learn implied raw data representations in network training.

The 1D-CNN with 32 filters effectively captures various features of the input data, especially in time series or sequential data. Each filter, tailored with specific sizes and features, aims to identify patterns present in the data. For stock data, these filters can detect crucial temporal patterns such as market price fluctuations, trends, trading volume variations, and market behavior shifts. These patterns provide insights into market dynamics, future movement probabilities, and specific market trends, including bullish, bearish, or sideways trends.

The Pooling layer input is a convolution layer output and is used to reduce the number of network parameters.

In 1D-Max-Pooling layer, Y_{cnn} is the output of the 1D-CNN layer. The max-pooling operation selects the maximum value from each pool of values in Y_{cnn} that is shown as Eq. (12).

$$Y_{maxpool} = MaxPool\left(Y_{cnn}\right) \tag{12}$$

The output $Y_{maxpool}$ of the max-pooling layer is obtained by selecting the maximum value from each pool of values in Y_{cnn}.

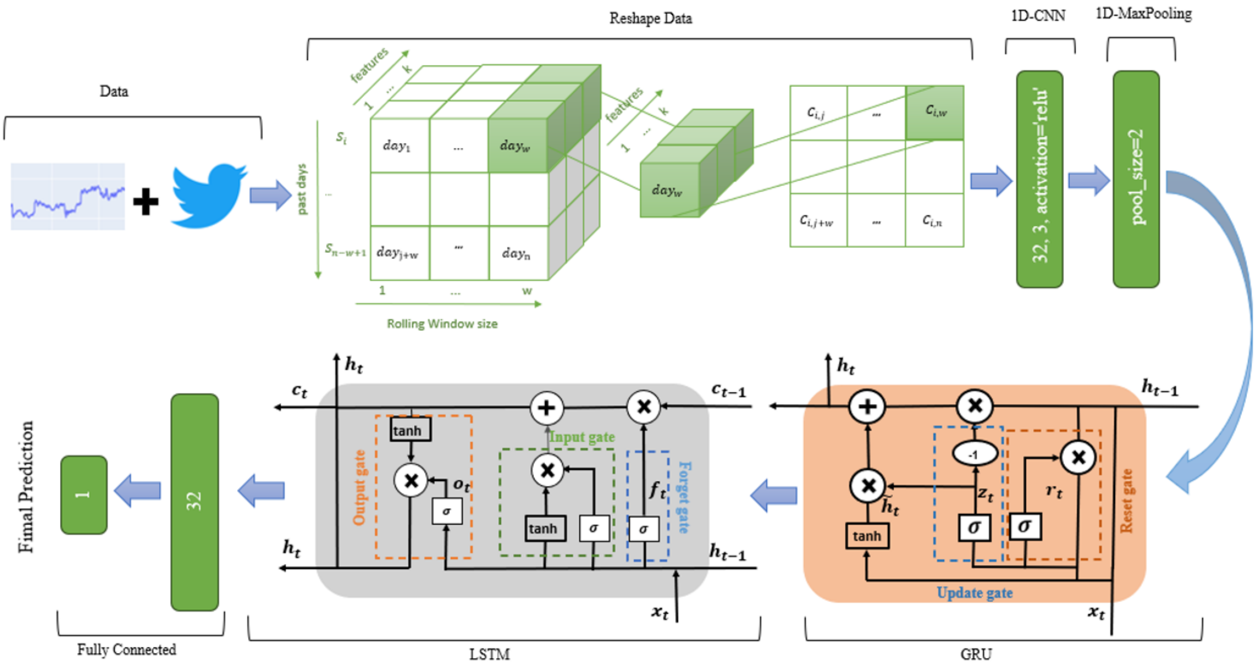

Figure 5. Structure of proposed 1D-CNN-LSTM model.

In GRU layer, X' is the input data to the GRU layer, and W_j be the weight matrices with j ranging from 1 to 128. The GRU layer performs linear transformations on the input X' using the weight matrices W_j shown as Eq. (13).

$$Y_{gru} = tanh\left(X'.W_j + b_j\right) \tag{13}$$

In LSTM layer; X'' is the input data to the LSTM layer, and W_k is the weight matrices with k ranging from 1 to 64. The LSTM layer performs linear transformations on the input X'' using the weight matrices W_k that is shown as Eq. (14).

$$Y_{lstm} = Relu\left(X''.W_k + b_k\right) \tag{14}$$

where W_k are the weight matrices and b_k are the biases.

In Dense layer 1, X''' is the input data to the dense layers, and W_m is the weight matrices with m ranging from 1 to 32. The dense layers perform linear transformations on the input X''' using the weight matrices W_m that is shown as Eq. (15).

$$Y_{dense} = Relu\left(X'''.W_m + b_m\right) \tag{15}$$

In Dense layer 2 (Output layer); X'''' is the input data to the output layer, and W_n is the weight matrices. The output layer performs a linear transformation on the input X'''' using the weight matrices W_n that is shown as Eq. (16).

$$Y_{Output} = Relu\left(X''''.W_n + b_n\right) \tag{16}$$

The technical indicators utilized in this study are outlined in Appendix I. It's noteworthy that all indicators are derived from the Pandas TA library, ensuring consistency and reliability in the analysis. The process involves calculating technical indicators and creating features in Appendix I from the tweet data set after preprocessing. Subsequently, the data set containing technical indicators and sentiment analysis is inputted into the hybrid DL model, which categorizes them based on stock price movement direction (Figure 4).

3.5. Evaluation criteria

3.5.1. Evaluation criteria for stock price prediction

In DL models, the evaluation of predictions often involves the utilization of loss error, which pertains to the disparity between the actual value and the predicted value. A multitude of tools can be employed to assess the loss error. We used the MSE, MAE, and MAPE to evaluate the proposed model. The calculation formulas for these metrics are presented as Eq. (17), (18), and (19).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{17}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{18}$$

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{19}$$

It is evident that the lower the value of the above criteria, the predictions will be more considerable and accurate [35].

3.5.2. Classification evaluation criteria for stock price movement

Different criteria exist for evaluating the classification of stock price movements, with the most common ones being Accuracy, and F1-score. They can be calculated as Eq. (20) and (21).

$$Accuracy = \frac{True\ Positive + \ True\ Negative}{Total\ Predictions} \tag{20}$$

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{21}$$

4. Experiments

The paper introduces a hybrid deep neural network and evaluates its performance using data obtained from various stock price indices. Before delving into the details of the proposed model, it is essential to describe the preprocessing steps for both the stock data and tweets data.

4.1. Details of stock data preprocessing

The time series data was retrieved from Yahoo Finance, spanning from 01/01/2015 to 12/31/2019. This data set comprises close, high, low, adjusted close, and volume data for four companies: Microsoft, Amazon, Apple, and Tesla. The closing price serves as the dependent variable, while the other features were used to compute technical indicators, subsequently removed from the data set (except for volume).

4.2. Details of tweets data preprocessing

The tweets data set encompasses tweets related to the four selected companies, spanning the same period from 01/01/2015 to 12/31/2019 [36]. The corpus contains 3,717,964 distinct tweets, distributed as follows: 1,425,013 tweets about Apple, 718,715 tweets about Amazon, 375,711 tweets about Microsoft, and 1,096,868 tweets about Tesla.

Tweets preprocessing involves eliminating punctuation marks, URLs, and symbols deemed non-essential. Additionally, words in the tweets undergo processing to remove stop words and irrelevant symbols. The internal sentiment analyzer, VADER, from NLTK, is utilized to classify each tweet into negative, neutral, or positive categories. The features of the Tweets Data set are outlined in Appendix I.

Features derived from sentiment analysis using VADER include [37]: the count of positive, negative, and neutral comments for each day in the tweet data set, the average score of positive, neutral, and negative comments per day, the volume of tweets per day, the ratio of positive and negative tweets [38], the count of retweets, positive retweets, negative retweets, and neutral retweets, as well as the count of likes, positive likes, negative likes, and neutral likes.

4.3. Details of implementation and training process

4.3.1. Detail of simulation

(

With the parameters setting for each component of the hybrid 1D-CNN-GRU-LSTM model through experimentation, we optimize the performance of the model for stock price prediction. The number of filters for CNN (1D-CNN) defines how many different features the model will learn at each layer. For time-series data, we used 32 and 64. The kernel size determines the size of the window that the CNN uses to scan over the input data. We used kernel sizes ranging from 2 to 5. The stride determines how much the filter moves at each step. A stride of 1 ensures the filter captures every small detail. We can apply max pooling or average pooling after each CNN layer to reduce the dimensionality. For 1D-CNN, a pool size of 2 is typical to down-sample the data, which also helps in reducing computation and mitigating overfitting. ReLU (Rectified Linear Unit) is commonly used as the activation function in CNN layers because it helps the model converge faster and introduces non-linearity. The number of units (neurons) in the GRU layer controls the model's ability to learn patterns from the time-series data. We experiment

with 64, 128 and 256 units, and adjust based on the performance during model evaluation. Similar to GRU, the number of units controls the capacity of the LSTM to learn long-term dependencies. We experiment with 64, 128 and 256 units for the LSTM.
Training Parameters (for the entire model) are shown in Table 1. The batch size here is 128. We used 100 epochs. Early Stopping performance is used to prevent overfitting issues.

4.3.2. System specifications

Proposed models in the Jupiter Notebook software with Python version 3.9.13 and with the Scikit-Learn Library version 1.3.0, NumPy version 1.22.4, Pandas version 1.4.4, TensorFlow and Keras version 2.10.0 and NLTK version 3.7 were trained and analyzed. All experiments were performed using an Intel Corei7, NVIDIA GeForce processor and 12GB of RAM.

Table 1. Hyperparameter setting for models.

Hyperparameter	Value
Number of epochs	100
Learning rate	0.0001
Batch size	128
Loss function	MSE
Optimizer	Adam
Activation function	ReLU

4.3.3. Details of data normalization and training process

Normalization of the raw data is performed using the MaxAbsScaler tool from Scikit-Learn. This scalar function scales and shifts each individual feature, ensuring that the largest absolute value of each feature in the training set is equal to 1.0 [39]. The formula for the MaxAbsScaler function is provided in Eq. (22).

$$x^i = \frac{x^i}{\left|x^i_{\max}\right|} \tag{22}$$

Where x^i is the ith feature of the data set and x^i_{max} represents the maximum value of the ith feature in the data set.
Approximately 20% of the data should be retained for the testing set [40]. By this recommendation, 80% of the initial data set was allocated for training purposes, while the remaining 20% was designated for testing. The initial 80% of the training set was used for training, leaving the remaining 20% for validation.

4.4. The Rolling Window for prediction

The rolling window technique is employed to enhance the accuracy of long-term time series predictions by dividing the data into various intervals for training [41]. In this study, a variable-length rolling window approach was adopted, as depicted in Figure 6. These windows progressively advance, allowing for the assessment of different window sizes. Rolling windows of sizes 1, 14, 30, and 60 days were evaluated to determine the optimal window size for predicting stock price movement.
By implementing this approach, the model can adapt to varying patterns and fluctuations in the data, thereby improving its predictive capabilities for different time horizons.
The rolling window can be expressed mathematically as Eq. (23):

$$s_i = \left\{day_j, day_{j+1}, \ldots day_{w+j-1}\right\}, \quad i = j-1, \; j = 1, \ldots, n-w+1 \tag{23}$$

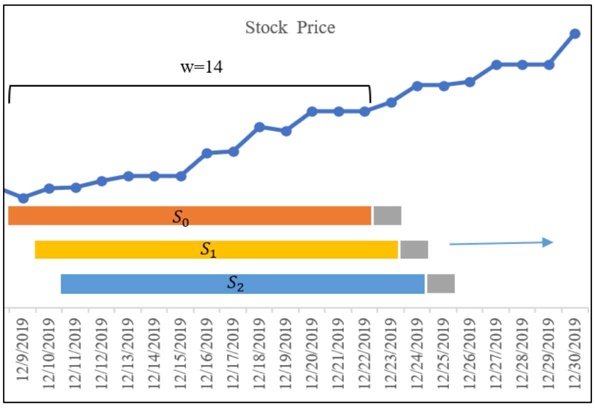

Figure 6. Structure of rolling windows.

Here, s_i represents the ith rolling window, w is the size of the rolling window, n is number of days in the data set, and day_j denotes the jth day in the data set. Generally, the input data for the ith rolling window, s_i, consists of day_j to day_{j+w-1}, forming the training data for a window of size w, and the output is indicated by day_{j+w}. The input data for the next rolling window, s_{i+1}, is from day_{j+1} to day_{j+w}, and the output is represented by day_{j+w+1}. This sequential progression occurs at each stage until the time series forecasting extends [42]. A subset of 191 data points has been selected for testing. Subsequently, the remaining data is partitioned into a training set comprising 80% and a validation set comprising 20%.
The data corresponding to each rolling window has been transformed into a data point called C_i, according to Eq. (24). C_i includes values of indicators and sentiment analysis and various aspects extracted from tweets on day jth within the

rolling window s_i .

$$C_{i,j} = \left\{ F_{1,1}^{(i)}, F_{2,1}^{(i)}, \ldots, F_{k,j}^{(i)} \right\} \tag{24}$$

Here, k represents the number of features. $F_{k,j}^{(i)}$ denotes the value of the kth feature for the jth day in the ith rolling window [43].

The choice of stocks is based on the big tweet dataset used for sentiment analysis. Time windows of different sizes (1, 7, 14, 21, 30, 45, 45 and 60 days) have been studied, which we reported a few in the article to prevent the tables from growing.

Model performance at different time windows allows investors to choose optimal time windows for their trading strategies, depending on how far they want to move prices.

5. Results

5.1. Stock price prediction results

The forecasting results for the stock prices of four companies by the 1D-CNN-GRU-LSTM and baseline models are presented in Table 2. The results are evaluated using windows of 1, 14, 30, and 60 days with three error-based metrics.

Based on Table 2, the 1D-CNN-GRU-LSTM model emerged as the best model for forecasting stock prices. Here are the MSE values for the stock price predictions by the proposed model:

-For MSFT stock data, with a 14-day rolling window, the MSE was 0.00099.

-For AMZN stock data, with a 60-day rolling window, the MSE was 0.00108.

-For TSLA stock data, with a 14-day rolling window, the MSE was 0.00136.

-For AAPL stock data, with a 1-day rolling window, the MSE was 0.00060.

Lower MSE values indicate better performance because the model's predictions are closer to the actual values. The MSE values reported for the stock price predictions suggest how well the model has captured the underlying trend in the stock prices over the respective time windows. Given that stock prices can be very volatile, a low MSE indicates that the model can predict prices with relatively small errors, which is crucial for financial forecasting where precision is key to minimizing risks and maximizing returns.

Additionally, the average squared error (MSE) of the three models for TSLA stock with a 30-day rolling window is depicted in Figure 7.

It's impressive to note the low MSE values achieved, indicating that the model's predictions are generally close to the actual stock prices. This demonstrates the effectiveness of the 1D-CNN-GRU-LSTM hybrid model in capturing the complex patterns and dynamics present in stock price data. Additionally, the visualization of the average squared error (MSE) for TSLA stock with a 30-day rolling window in Figure 7 provides further insights into the performance of the model over time. This allows for a more comprehensive understanding of how the model's predictive accuracy evolves over different time horizons. Overall, these results validate the efficacy of the proposed hybrid model for stock price prediction and underscore its potential utility in real-world financial applications.

Based on Figure 7, the models have shown better optimization in the early stages of the learning period. With the progress of time and the number of epochs, learning has reached better stages, indicating improvement and enhancement of the models' performance.

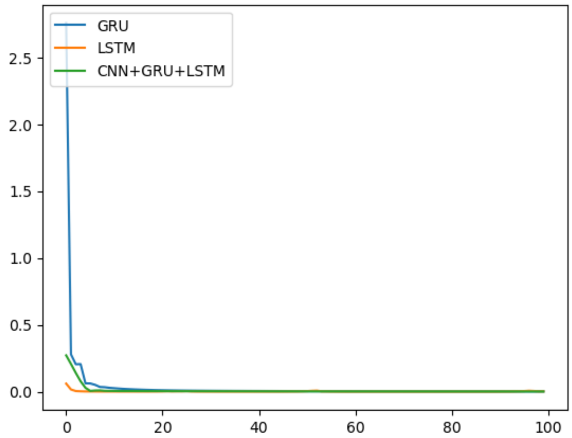

Figure 7. MSE of the models on TSLA stock with w=30.

5.2. Stock price movement prediction

Here, the movement of stock prices is classified into three categories: up, down, and stable. To create a classification variable for predicting stock price movements in a specific rolling window, we use three classes:

- Rise: If the stock price increases compared to the beginning of the period.
- Fall: If the stock price decreases compared to the beginning of the period.
- Stable: If the stock price remains constant compared to the beginning of the period.

The direction of stock price movement is expressed by Eq. (19).

$$\nabla x_t = x_{t+1} - x_t \begin{cases} \nabla x_t > 0 & class = Rise \\ \nabla x_t < 0 & class = Fall \\ \nabla x_t = 0 & class = Stable \end{cases} \tag{19}$$

A new time series, denoted as ∇x_t , has been created to represent changes in stock prices over a specific period. Here, x_t represents the value of the time series (stock price) at period *t*.

The results of predicting stock price movements for four companies using the 1D-CNN-GRU-LSTM model and baseline models, based on the discussed classification criteria, are presented in Table 3. These results have been evaluated using rolling windows of 1, 14, 30, and 60 days.

Based on Table 3, the 1D-CNN-GRU-LSTM model performed the best for forecasting stock price movements. Here are the results for stock price movement forecasting by the proposed model:

Table 2. Result of stock price prediction for variate rolling windows.

W	Evaluation Criteria	Model	MSFT	AMZN	TSLA	APPL
		1D-CNN-GRU-LSTM	0.00251	0.00162	0.00172	**0.00060**
	MSE	GRU	0.00287	0.00126	0.00248	0.00078
		LSTM	0.00181	0.0010	0.00213	0.00061
		1D-CNN-GRU-LSTM	0.05307	0.03828	0.05038	0.02981
1	MAPE	GRU	0.05443	0.03264	0.06655	0.03160
		LSTM	0.04315	0.02851	0.05908	0.02694
		1D-CNN-GRU-LSTM	0.04324	0.03352	0.03139	0.02022
	MAE	GRU	0.04546	0.02849	0.04053	0.02199
		LSTM	0.03543	0.02513	0.03781	0.01940
		1D-CNN-GRU-LSTM	**0.00099**	0.00156	**0.00136**	0.00082
	MSE	GRU	0.00323	0.02882	0.00540	0.00229
		LSTM	0.00782	0.00242	0.00193	0.00205
		1D-CNN-GRU-LSTM	0.03098	0.03634	0.04744	0.03280
14	MAPE	GRU	0.05598	0.15753	0.09502	0.05287
		LSTM	0.09554	0.04655	0.05753	0.05284
		1D-CNN-GRU-LSTM	0.02563	0.03192	0.02877	0.02321
	MAE	GRU	0.04689	0.13890	0.05990	0.03804
		LSTM	0.07834	0.04038	0.03536	0.03635
		1D-CNN-GRU-LSTM	0.00137	0.00174	0.00153	0.00153
	MSE	GRU	0.00200	0.00296	0.01163	0.00224
		LSTM	0.00345	0.00622	0.00480	0.00146
		1D-CNN-GRU-LSTM	0.03477	0.03761	0.04746	0.04265
30	MAPE	GRU	0.04214	0.04940	0.14014	0.05181
		LSTM	0.05342	0.06771	0.09937	0.03889
		1D-CNN-GRU-LSTM	0.02941	0.03350	0.02942	0.03117
	MAE	GRU	0.03541	0.04423	0.08403	0.03824
		LSTM	0.04600	0.06103	0.05567	0.02912
		1D-CNN-GRU-LSTM	0.00380	**0.00108**	0.00172	0.00118
	MSE	GRU	0.00254	0.00640	0.00454	0.00520
		LSTM	0.01170	0.05052	0.00229	0.00087
		1D-CNN-GRU-LSTM	0.06768	0.02934	0.05655	0.03784
60	MAPE	GRU	0.04770	0.06610	0.08521	0.08619
		LSTM	0.11640	0.24831	0.05587	0.03226
		1D-CNN-GRU-LSTM	0.05755	0.02621	0.03379	0.02816
	MAE	GRU	0.03963	0.05880	0.05292	0.06009
		LSTM	0.10028	0.22271	0.03612	0.02368

-For MSFT stock data, with a 60-day rolling window, the Accuracy was 0.86408.

-For AMZN stock data, with a 30-day rolling window, the Accuracy was 0.86425.

-For TSLA stock data, with a 30-day rolling window, the Accuracy was 0.92760.

-For AAPL stock data, with a 60-day rolling window, the Accuracy was 0.93717.

High accuracy in predicting the direction of stock prices (Rise**,** Fall**,** Stable), indicates that the model is correctly identifying the trend most of the time. The reported accuracy values suggest that the model is highly effective in predicting the direction of stock movements over different time periods. High accuracy in predicting stock price direction is particularly valuable in trading and investment strategies, where knowing the likely direction of price movement can directly influence buy/sell decisions and portfolio management.

Table 3. Result of stock price movement prediction for variate rolling windows.

W	Evaluation Criteria	Model	MSFT	AMZN	TSLA	APPL
		1D-CNN-GRU-LSTM	0.448	0.5	0.488	0.45
	Accuracy	GRU	0.548	0.484	0.476	0.524
		LSTM	0.536	0.488	0.512	0.5
1		1D-CNN-GRU-LSTM	0.45207	0.50057	0.48833	0.45777
	F1	GRU	0.55076	0.48440	0.47588	0.52662
		LSTM	0.53984	0.488	0.512	0.50276
		1D-CNN-GRU-LSTM	0.65401	0.76793	0.83544	0.75105
	Accuracy	GRU	0.66245	0.72574	0.68354	0.71730
		LSTM	0.52321	0.78903	0.81857	0.85654
14		1D-CNN-GRU-LSTM	0.68397	0.77010	0.83515	0.77964
	F1	GRU	0.69568	0.72642	0.68256	0.75233
		LSTM	0.56873	0.79096	0.81780	0.86542
		1D-CNN-GRU-LSTM	0.78733	**0.86425**	**0.92760**	0.85520
	Accuracy	GRU	0.76923	0.71041	0.77376	0.80090
		LSTM	0.65158	0.76923	0.80543	0.90045
30		1D-CNN-GRU-LSTM	0.80983	0.86315	0.92786	0.86072
	F1	GRU	0.79927	0.71034	0.77402	0.81621
		LSTM	0.69928	0.76904	0.80530	0.90575
		1D-CNN-GRU-LSTM	**0.86408**	0.85340	0.87435	**0.93717**
	Accuracy	GRU	0.85864	0.77487	0.86911	0.82199
		LSTM	0.86911	0.78010	0.86911	0.93241
60	F1	1D-CNN-GRU-LSTM	0.88042	0.85156	0.87808	0.94294
		GRU	0.88084	0.77845	0.86484	0.86802
		LSTM	0.89631	0.78230	0.86859	0.92989

5.3. Comparison with individual models

Our proposed hybrid model, which combines 1D-CNN, GRU and LSTM, shows significant improvements to the separate LSTM and GRU models. Here are the advantages of the hybrid model in terms of performance:

- Performance on predictive criteria (MSE and precision): The proposed hybrid model for stock price predictions shows less MSE than separate LSTM or GRU models. According to the results (Table 2), the MSE hybrid model is 0.00092 for MSFT with a 7-day window, which is less than the MSE of LSTM and GRU models. This decrease in MSE indicates a greater accuracy of the hybrid model in capturing complex and non-linear relationships in stock prices.

Individual LSTM models, though strong in long-term dependencies, may not record short-term patterns well and as a result their MSE is slightly higher than the hybrid model. Also, individual GRU model, being simpler than LSTM, tends to underperform slightly in capturing both short-term fluctuations and long-term dependencies. Its MSE was also consistently higher than the hybrid model's MSE.

Also for accuracy in predicting the stock price movement, in predicting the direction of stock prices (Rise**,** Fall**,** Stable), the hybrid model also works better than individual LSTM and GRU models. According to the results (Table 3), the hybrid model obtained 93.71% accuracy for AAPL with a 60-day window, which is more accurate than LSTM and GRU models. This higher accuracy means the model's ability to make reliable decisions about the stock movement direction, which is very important in real financial scenarios.

- Combining the strengths of CNN, GRU and LSTM: The 1D-CNN layer helps the hybrid model extract meaningful features and patterns from stock data, especially in the identification of short-term trends and fluctuations. LSTM and GRU models alone do not have this feature. Adding CNN to the model allows for better preprocessing and filtering of input data, thereby improving the performance of RNN layers (GRU and LSTM).

The hybrid model uses the strengths of both GRU and LSTM. GRU is more efficient in computing and works better for smaller data, while LSTM can better manage long-term dependencies. By combining the two, the hybrid model can balance the performance and computing time, which makes it more suitable for different market conditions (short-term fluctuations and long-term trends).

Individual GRU and LSTM models may develop more or less, depending on the complexity of the stock market movements. The LSTM may face the problems of the vanishing gradient for short-term patterns, while GRU, despite being lighter in computing, may not fully record long-term dependencies. The hybrid model reduces these two weaknesses.

- Computational efficiency: The hybrid model is more computationally efficient compared to running deep LSTM or GRU networks on their own. CNN helps reduce the dimensionality of input data, and GRU reduces the computational burden, leading to faster training and inference times. This is particularly beneficial when scaling to larger datasets or when deploying the model in real-time trading systems.

- Handling noisy data: Stock market data often contains noise and irrelevant fluctuations. The 1D-CNN in the hybrid model acts as a feature extraction layer that helps in filtering out noise and extracting relevant patterns from the raw data. In contrast, individual LSTM and GRU models can sometimes get influenced by this noise, leading to higher prediction errors.

- Handling temporal dependencies: The hybrid model effectively captures both short-term (via CNN and GRU) and long-term dependencies (via LSTM), making it better suited for time series data like stock prices, which often have patterns occurring at multiple time scales. GRU and LSTM, when used individually, tend to focus on either short-term or long-term dependencies, but not both as efficiently as the hybrid model. Therefore, the hybrid 1D-CNN-GRU-LSTM model consistently outperforms the individual models in terms of predictive accuracy and computational efficiency. Its combination of CNN for feature extraction and the complementary strengths of GRU and LSTM allows it to handle both short- and long-term dependencies in stock market data more effectively.

5.4. Calculation of stock returns in a specific period (Rate of Return)

One of the most important tasks in financial markets is analyzing the historical returns of various investments. The Rate of Return (RoR) represents the net profit or loss of an investment over a specified period, indicated as the initial investment cost percentage. When calculating the rate of return, we determine the percentage change from the beginning to the end of the period. This simple rate of return is sometimes called the base rate of growth or, in other words, Return on Investment (ROI). The daily return rate indicates the percentage change in stock price over a specific day.

Relative return is a measure used in investing that estimates the return of an asset or any other portfolio relative to a theoretical benchmark over a specific period. This return is calculated as the difference between the absolute return obtained by the asset or portfolio and the return obtained by the benchmark. Mathematically, this return can be expressed as Eq. (25):

$$ROR_W = \frac{Close_{j+w-1} - Close_j}{Close_j} \times 100 \qquad (25)$$

Here, ROR_W represents the relative rate of return over period w. $Close_j$ is the closing price of stocks at the beginning of the period, and $Close_{j+w-1}$ is the closing price of stocks at the end of the specified period.

To compare stock returns over longer periods, we compute weekly or monthly returns and compare them, which is useful for demonstrating long-term patterns and market trends.The relative rate of return for rolling windows of various sizes for the four specified companies has been calculated using the MSE error metric and is presented in Table 4.

Table 4. Evaluate of ROR with MSE criteria.

W	MSFT	AMZN	TSLA	APPL
1	**0.000562**	**0.000541**	**0.002209**	**0.001545**
14	0.004536	0.004772	0.026022	0.008149
30	0.004704	0.012425	0.038577	0.014107
60	0.003368	0.002079	0.008164	0.001940

Based on Table 4, the relative rate of return for the four examined companies has been calculated with very low error rates. The MSE error rates for companies AMZN and MSFT are 0.000541 and 0.000562, respectively, which are very close to zero.

Comparison of daily stock returns is typically done by examining daily price changes and comparing daily stock returns (percentage changes) over a specific period. Therefore, comparing stock returns over different periods can aid in understanding market performance, analyzing market trends, and making appropriate financial decisions.

The forecasting results of the proposed model compared with past works discussed in this paper for stock price movement prediction and stock price prediction are presented in Tables 5 and 6. In all cases, the proposed model outperformed other models.

Based on Tables 5 and 6, comparing the new results with past works indicates an improvement in the performance of the new model compared to previous approaches. The improvement of the proposed hybrid model for stock price movement prediction over existing models (compared to reference [23]) is 12% for accuracy and F1-score metrics.

6. Discussion and Conclusion

In this paper, a new hybrid DL model for predicting stock price movements was proposed, which yielded very good results. The proposed model was evaluated on the stocks of four major companies. Technical indicators and sentiment analysis of relevant tweets about the companies were utilized within the specified time intervals. In comparing the proposed model with individual models, namely LSTM and GRU, the best model for stock price prediction was the 1D-CNN-GRU-LSTM hybrid model. Stock price prediction by the proposed model with MSE criteria yielded values of 0.00099 for MSFT with a 14-day rolling window, for AMZN with a 60-day is 0.00108, TSLA with a 14-day is 0.00136 and for AAPL with a 1-day window size is 0.00060. Additionally, the proposed hybrid model was the best for predicting stock price movements. Stock price movement prediction by the proposed model achieved an accuracy of 0.86408 for MSFT 60-day, for AMZN with 30-day is 0.86425, for TSLA with 30-day is 0.92760 and for AAPL with 60-day a window size is 0.93717. Also, the relative rate of return for the four examined companies was calculated with very low error rates. The results obtained from the proposed deep hybrid model in this study have practical significance in predicting market trends and can assist in decision-making in the stock market.

Table 5. Comparison of Proposed Model with Existing work for stock price movement prediction.

Model	Accuracy	F1-score
1D-CNN-GRU-LSTM	**0.93717**	**0.94294**
[23]	0.822	0.821
[24]	0.6418	0.6180
CNN-LSTM-News Headlines [21]	0.7476	-
CNN-LSTM-OHLC [21]	0.8873	-
2D-CNNpred [25]	-	0.5562
3D-CNNpred [25]	-	0.5787

Table 6. Comparison of Proposed Model with Existing work for stock price prediction.

Model	MSE	MAPE	MAE
1D-CNN-GRU-LSTM	**0.00060**	**0.02981**	**0.02022**
EMD-CNN-LSTM [20]	-	0.611	12.04
CEEMD-CNN-LSTM [20]	-	0.536	10.58
CNN-LSTM [17]	0.0078	-	0.0742
BiGRU-CNN-FS [19]	-	0.01488	-
CNN-GRU-ATTENTION [16]	-	-	0.86
CNN-LSTM-ATTENTION [16]	-	-	1.04

Future works

The potential of using the proposed deep hybrid model in the future is significant, and further improvement in results can be achieved by adding more layers and additional financial data sets. In the future, we will develop a stronger model by combining conventional and DL models. For future work, we can extend the model to predict other financial metrics such as bond prices, commodity prices (e.g., oil, gold), or currency exchange rates. Also incorporating other forms of financial data, such as macroeconomic indicators, interest rates, or global market indices, can enhance the model's predictive power. By integrating a wider array of financial indicators, the model could potentially capture more complex relationships within financial markets. Also, we suggest the incorporation of More Advanced Techniques such as Attention Mechanisms, Transformer Models, and Enhanced Sentiment Analysis.

Data Availability and Access Dataset is available for research purposes from the first author.

Declarations of interests

Competing Interests 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.

Ethical and informed consent for data used The data used is already available.

Authors contribution The authors confirm contribution to the paper as follows: study conception and design: First author; data collection: First author; analysis and interpretation of results: First author, Second author; evaluation: First author, Second author, Third author, Fourth author; draft manuscript preparation: First Author. All authors reviewed the results and approved the final version of the manuscript.

References

[1] G. A. Pérez, "Forecasting and optimization of stock trades," University of Seville, 2021.

[2] J. M. Polanco-Martínez, "Dynamic relationship analysis between NAFTA stock markets using nonlinear, nonparametric, non-stationary methods," *Nonlinear Dynamics*, vol. 97, no. 1, pp. 369-389, 2019/07/01 2019, doi: 10.1007/s11071-019-04974-y.

[3] E. F. Fama, "The behavior of stock-market prices," *The journal of Business,* vol. 38, no. 1, pp. 34-105, 1965.

[4] X. Zhang, X. Liang, A. Zhiyuli, S. Zhang, R. Xu, and B. Wu, "At-lstm: An attention-based lstm model for financial time series prediction," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 569, no. 5: IOP Publishing, p. 052037.

[5] R. Shields, S. Ajour El Zein, and N. Vila Brunet, "An analysis on the NASDAQ's potential for sustainable investment practices during the financial shock from Covid-19," *Sustainability,* vol. 13, no. 7, p. 3748, 2021.

[6] M. K. Daradkeh, "A hybrid data analytics framework with sentiment convergence and multi-feature fusion for stock trend prediction," *Electronics,* vol. 11, no. 2, p. 250, 2022.

[7] S. Chen and H. He, "Stock prediction using convolutional neural network," in *IOP Conference series: materials science and engineering*, 2018, vol. 435: IOP Publishing, p. 012026.

[8] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," *The Journal of Supercomputing,* vol. 76, pp. 2098-2118, 2020.

[9] J. Cao and J. Wang, "Stock price forecasting model based on modified convolution neural network and financial time series analysis," *International Journal of Communication Systems,* vol. 32, no. 12, p. e3987, 2019.

[10] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. K. Menon, and K. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *2017 international conference on advances in computing, communications and informatics (icacci)*, 2017: IEEE, pp. 1643-1647.

[11] D. M. Nelson, A. C. Pereira, and R. A. De Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *2017 International joint conference on neural networks (IJCNN)*, 2017: Ieee, pp. 1419-1426.

[12] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach," *Applied Soft Computing,* vol. 70, pp. 525-538, 2018.

[13] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, K. R. Dahal, and R. K. Khatri, "Predicting stock market index using LSTM," *Machine Learning with Applications,* vol. 9, p. 100320, 2022.

[14] M. Faghihi Nezhad and B. Minaei Bidgoli, "Development of an ensemble learning-based intelligent model for stock market forecasting," *Scientia Iranica,* vol. 28, no. 1, pp. 395-411, 2021.

[15] J. Zhang, S. Cui, Y. Xu, Q. Li, and T. Li, "A novel data-driven stock price trend prediction system," *Expert Systems with Applications,* vol. 97, pp. 60-69, 2018..

[16] C. Chen, "Stock Price Prediction Based on the Fusion of CNN-GRU Combined Neural Network and Attention Mechanism," in *2023 6th International Conference on Electronics Technology (ICET)*, 2023: IEEE, pp. 1166-1170.

[17] H. Song and H. Choi, "Forecasting stock market indices using the recurrent neural network based hybrid models: CNN-LSTM, GRU-CNN, and ensemble models," *Applied Sciences,* vol. 13, no. 7, p. 4644, 2023.

[18] M. E. Karim, M. Foysal, and S. Das, "Stock price prediction using Bi-LSTM and GRU-based hybrid deep learning approach," in *Proceedings of Third Doctoral Symposium on Computational Intelligence: DoSCI 2022*, 2022: Springer, pp. 701-711.

[19] Q. Zhou, C. Zhou, and X. Wang, "Stock prediction based on bidirectional gated recurrent unit with convolutional neural network and feature selection," *PloS one,* vol. 17, no. 2, p. e0262501, 2022.

[20] H. Rezaei, H. Faaljou, and G. Mansourfar, "Stock price prediction using deep learning and frequency decomposition," *Expert Systems with Applications,* vol. 169, p. 114332, 2021.

[21] S. Gite, H. Khatavkar, K. Kotecha, S. Srivastava, P. Maheshwari, and N. Pandey, "Explainable stock prices prediction from financial news articles using sentiment analysis," *PeerJ Computer Science,* vol. 7, p. e340, 2021.

[22] A. Nayak, M. M. Pai, and R. M. Pai, "Prediction models for Indian stock market," *Procedia Computer Science,* vol. 89, pp. 441-449, 2016.

[23] G. Ji, J. Yu, K. Hu, J. Xie, and X. Ji, "An adaptive feature selection schema using improved technical indicators for predicting stock price movements," *Expert Systems with Applications,* vol. 200, p. 116941, 2022.

[24] Y. Peng, P. H. M. Albuquerque, H. Kimura, and C. A. P. B. Saavedra, "Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators," *Machine Learning with Applications,* vol. 5, p. 100060, 2021.

[25] E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN-based stock market prediction using a diverse set of variables," *Expert Systems with Applications,* vol. 129, pp. 273-285, 2019.

[26] N. A. Bakshi, P. R. Kolan, B. Behera, N. Kaushik, and A. M. Ismail, "Predicting Pregnant Shoppers Based on Purchase History Using Deep Convolutional Neural Networks," *Journal of Advances in Information Technology Vol,* vol. 9, no. 4, 2018.

[27] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision,* vol. 115, pp. 211-252, 2015.

[28] W. Chen, M. Jiang, W.-G. Zhang, and Z. Chen, "A novel graph convolutional feature based convolutional neural network for stock trend prediction," *Information Sciences,* vol. 556, pp. 67-94, 2021.

[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555,* 2014.

[30] X. Le, H. V. Ho, G. Lee, and S. Jung, "Application of long short-term memory (LSTM) neural network for flood forecasting," *Water,* vol. 11, no. 7, p. 1387, 2019.

[31] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, "Deep learning with gated recurrent unit networks for financial sequence predictions," *Procedia computer science,* vol. 131, pp. 895-903, 2018.

[32] Ł. Kaiser, and I. Sutskever, "Neural gpus learn algorithms," *arXiv preprint arXiv:1511.08228*, 2015.

[33] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," *arXiv preprint arXiv:1702.01923,* 2017.

[34] I. E. Livieris, E. Pintelas, S. Stavroyiannis, and P. Pintelas, "Ensemble deep learning models for forecasting cryptocurrency time-series," *Algorithms,* vol. 13, no. 5, p. 121, 2020.

[35] F. Moodi and H. Saadatfar, "An improved K-means algorithm for big data," *IET Software,* vol. 16, no. 1, pp. 48-59, 2022.

[36] M. Doğan, Ö. Metin, E. Tek, S. Yumuşak, and K. Öztoprak, "Speculator and influencer evaluation in stock market by using social media," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020: IEEE, pp. 4559-4566.

[37] F. Moodi, A. Jahangard-Rafsanjani, and S. Zarifzadeh, "A CNN-LSTM deep neural network with technical indicators and sentiment analysis for stock price forecastings," in *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 2024: IEEE, pp. 1-6.

[38] F. Moodi, A. Jahangard-Rafsanjani, and S. Zarifzadeh, "Improving stock price prediction using technical indicators and sentiment analysis," *TABRIZ JOURNAL OF ELECTRICAL ENGINEERING,* 2024.

[39] https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.MaxAbsScaler.html#sklearn.preprocessing.MaxAbsScaler

[40] C. W. Granger, "Strategies for modelling nonlinear time-series relationships," *Economic Record,* vol. 69, no. 3, pp. 233-238, 1993.

[41] E. Zivot, J. Wang, E. Zivot, and J. Wang, "Rolling analysis of time series," *Modeling financial time series with S-Plus®,* pp. 299-346, 2003.

[42] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," *arXiv preprint arXiv:2001.08317,* 2020.

[43] F. Moodi, A. Jahangard-Rafsanjani, S. Zarifzadeh, "Evaluation of feature selection performance for identification of best effective technical indicators on stock market price prediction," *arXiv Preprint,* 2023. https://doi.org/10.48550/arXiv.2310.09903

[44] M. Taghian, A. Asadi, and R. Safabakhsh, "A reinforcement learning based encoder-decoder framework for learning stock trading rules," *arXiv preprint arXiv:2101.03867,* 2021.

7. Appendix I. Description of features

Table A. Indicators Name.

No.	Name	No.	Name	No.	Name	No.	Name
1	aberration	32	ebsw	63	natr	94	sma
2	apo	33	er	64	ohlc4	95	slope
3	accbands	34	ssf	65	obv	96	smi
4	ad	35	eri	66	psar	97	squeeze
5	adosc	36	efi	67	pwma	98	squeeze_pro
6	alma	37	ema	68	ppo	99	stoch
7	aobv	38	fwma	69	pvo	100	stochrsi
8	aroon	39	fisher	70	pvi	101	supertrend
9	adx	40	hilo	71	pgo	102	sine wma
10	atr	41	hl2	72	pdist	103	swma
11	ao	42	hlc3	73	pvr	104	t3
12	bop	43	hwc	74	pvt	105	td_seq
13	bias	44	hma	75	pvol	106	tema
14	bbands	45	hwma	76	psl	107	thermo
15	brar	46	ichimoku	77	qstick	108	trima
16	cg	47	increasing	78	qqe	109	trix
17	cmf	48	amat	79	roc	110	true_range
18	cfo	49	kama	80	rsi	111	tsi
19	cksp	50	kdj	81	rsx	112	ttm_trend
20	cmo	51	kc	82	rvgi	113	ui
21	chop	52	kvo	83	rvi	114	uo
22	cci	53	kst	84	rma	115	vhf
23	coppock	54	linreg	85	zscore	116	vidya
24	cti	55	massi	86	stdev	117	vp
25	decay	56	mcgd	87	kurtosis	118	vwap
26	decreasing	57	midpoint	88	mad	119	vortex
27	dema	58	midprice	89	median	120	wcp
28	dpo	59	mom	90	quantile	121	willr
29	dm	60	mfi	91	skew	122	wma
30	donchian	61	macd	92	variance	123	zlma
31	eom	62	nvi	93	stc		

Table B. Tweet Features.

No.	Name	No.	Name
1	Count_retweet	11	Count_ comment
2	Count_Positive_retweet	12	Count_Positive_comment
3	Count_Negative_retweet	13	Count_Negative_comment
4	Count_ Neutral _retweet	14	Count_ Neutral _comment
5	Count_Positive	15	Score_mean_Positive
6	Count_Negative	16	Score_mean_Negative
7	Count_Neutral	17	Score_mean_Neutral
8	Count_like	18	Count_Positive_like
9	Count_Negative_like	19	Count_ Neutral _like
10	Count_ratio_NegPos	20	Tweet_vol

پیش‌بینی قیمت سهام پیشرفته با استفاده از مدل 1D-CNN-GRU-LSTM

فاطمه مودی، امیر جهانگرد رفسنجانی*، سجاد ظریف زاده و محمدعلی زارع چاهوکی

دانشکده مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران

ارسال ۲۰۲۴/۰۸/۰۵؛ بازنگری ۲۰۲۴/۰۸/۲۷؛ پذیرش ۲۰۲۴/۱۰/۰۹

چکیده:

این مقاله یک مدل ترکیبی عمیق جدید بر پایه ۳ معماری مختلف CNN و GRU و LSTM برای پیش‌بینی حرکت قیمت سهام ارائه می‌دهد. در اینجا ترکیب استخراج ویژگی و یادگیری توالی به همراه نقاط قوت تکمیلی می‌تواند عملکرد پیش‌بینی را بهبود بخشد. CNNها می‌توانند به طور مؤثر وابستگی‌های کوتاه‌مدت و ویژگی‌های مربوطه، مانند روند در قیمت سهام، را شناسایی کنند. GRUها برای مدیریت داده‌های متوالی طراحی شده‌اند و برای گرفتن وابستگی‌ها در طول زمان مفید هستند، ولی از نظر محاسباتی ارزان‌تر از LSTMها هستند. در مدل ترکیبی، GRU به حفظ اطلاعات تاریخی مربوطه در دنباله، بدون مشکل شیب ناپدیدشده، کمک می‌کند و آن را برای توالی‌های طولانی کارآمدتر می‌کند. LSTM به لطف ساختارِ سلول حافظه، در یادگیری وابستگی‌های طولانی‌مدت در داده‌های متوالی بسیار خوب عمل می‌کند. با حفظ اطلاعات در دوره‌های طولانی‌تر، LSTMها در مدل ترکیبی اطمینان حاصل می‌کنند که روندهای مهم در طول زمان از بین نمی‌روند و درک عمیق‌تری از داده‌های سری زمانی ارائه می‌دهند. نوآوری مدل ترکیبی جدید، در توانایی آن در ضبط همزمان الگوهای کوتاه‌مدت و وابستگی‌های بلندمدت در داده‌های سری زمانی نهفته است و پیش‌بینی دقیق‌تری از قیمت سهام را ارائه می‌دهد. داده‌ها شامل نشانگرهای فنی، تجزیه‌وتحلیل احساسات و جنبه‌های مختلف توییت‌ها است. حرکت قیمت به سه دسته طبقه‌بندی می‌شود: افزایش، کاهش و ثابت. ارزیابی این مدل در داده‌های ۵ ساله‌ی سهام، توانایی آن را برای پیش‌بینی حرکات قیمت سهام با دقت ۰/۹۳۷۱۷ نشان می‌دهد. بهبود مدل پیشنهادی برای پیش‌بینی حرکت سهام نسبت به مدل‌های موجود برای دقت و امتیاز F1، ۱۲ درصد است.

کلمات کلیدی: شبکه عصبی عمیق ترکیبی، شبکه 1D-CNN-GRU-LSTM، پیش‌بینی حرکت قیمت سهام، تجزیه و تحلیل احساسات توییت، نشانگرهای فنی