Improved optimization model for forecasting stock directions (FSD)

Noura Metawa, Iman Akour, Zahra Tarek & Mohamed Elhoseny

Published online: 21 Jun 2023.

Submit your article to this journal ↗

Article views: 161

View related articles ↗

View Crossmark data ↗

Routledge
Taylor & Francis Group

Improved optimization model for forecasting stock directions (FSD)

Noura Metawa[a,b] (iD), Iman Akour[c], Zahra Tarek[d] and Mohamed Elhoseny[c,d] (iD)

[a]College of Business Administration, University of Sharjah, Sharjah, UAE; [b]Faculty of Commerce, Mansoura University, Mansoura, Egypt; [c]College of Computing and Informatics, University of Sharjah, Sharjah, UAE; [d]Faculty of Computers and Information Systems, Mansoura University, Mansoura, Egypt

ABSTRACT

The study of stock market price predictions is very important. The Recurrent Neural Network (RNN) has shown excellent results with this issue. There are two significant problems with using this strategy. One is that it constantly struggles with extensive neural network construction efforts and hyper-parameter adjustments. Two, it often fails to come up with a superior answer. The suggested model is proposed to optimize the network topology and hyper-parameters of the RNN model. RNN is utilized for effective forecasting of stock directions in this research. Additionally, the Improved Differential Evolution (IDE) method is used to tune the RNN's hyperparameters to their best potential. Utilizing the IDE method helps in achieving the best stock direction prediction results possible. The direction of Stock Prediction (SP) changes has been accurately predicted by the proposed model that is being presented. A series of tests on two popular benchmark datasets (AAPL and FB) revealed the superiority of the proposed model over the other strategies with accuracy of 99.02% and the loss close of 0.1% for training and testing.

1. Introduction

In last many years, the viable contrasts in stock price (SP) can't be recognized in earlier times. Irregular Walk (Javed Awan et al., 2021), along with Efficient Market Hypothesis (EMH) uncovered that a market might be taken into account as hearty and proficient based on ongoing data, in the event that it isn't achievable to distinguish the market stream because of the arbitrariness of SPs, then, at that point, it is like the dangers which can be recuperated, practical increase can't be upgraded. The errand of stock market price (SMP) identification is exceptionally perplexing and it gives a greatest return from expanding measure for hazard expectation. Afterward, the Wisdom of
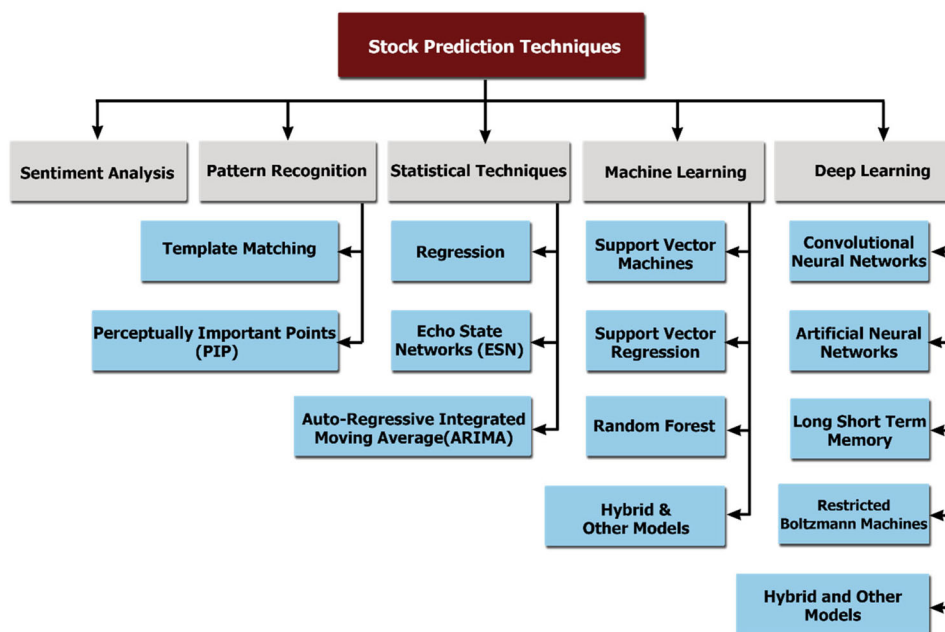
Figure 1. Classification of stock prediction techniques.
Source: the Authors.

Crowd system stated that different people can give precise assessment of data was recuperated fittingly (Metawa & Mutawea, 2022). This turn out to be obscure and it isn't appropriate to identify the aftereffect of stock markets (SM); notwithstanding, a portion of the people and authoritative investors are fit for outstanding the market to create better gains (Thakkar & Chaudhari, 2021). A fragmented location is improved because of assorted abnormalities present and in view of the presence of huge boundaries that impacts the market esteem each day. Subsequently, SMs are exceptionally defenseless for contrasts for conspicuous for making irregular struggles in SPs.

Different prudent as well as measurable experts have conjured to allude that the SMP is anticipated somewhat. Afterward, an original affordable master has pointed the psychological and social components of stock-price calculation and unpredictability (Javed Awan et al., 2021). Next, just couple of experts have expressed the deciding examples would empower the investor to achieve most extreme addition from hazard-based values (Pustokhin & Pustokhina, 2022). As of late, SM discovery techniques are conveyed utilizing Data Mining (DM) as well as Machine Learning (ML) procedures. A portion of the connected works are characterized in the accompanying. Prescient methodologies are exploited to conjecture the upcoming examples in SM tasks that offer a technique to further develop the deciding capacities and predetermine productive market system as well as dissemination plans (Goh et al., 2021). ML, additionally that is group of present methodologies. At long last, various strategies are applied to gauge SPs like DNN, SVM, NB, RF, and so on to accomplish the strategy for consistency with most extreme productivity. DM techniques are exceptionally utilized with normal stock information. Figure 1 displays the classifications of stock prediction techniques.

For the most part, Autoregressive Integrated Moving Average (ARIMA) structure (Tuarob et al., 2021) exploited for discovering and recognizing contrasts in time sequence. In shutting prices are utilized for investigating the firm of 3 M that have information from the time period 2008–August 2013 (Mokhtari et al., 2021). Monstrous techniques are utilized for investigator draws near and executed to gauge the component of stocks on compelling day's information test. It is additionally equipped for foreseeing the sum for future n days (Ananthi & Vijayakumar, 2021). It can be expressed that US SM is to some extent powerful, which means specialized also as basic investigation can't be utilized and achieve higher increase (Shilpa & Shambhavi, 2021). In any case, the delayed prescient approach offers exactness that had outperformed assuming the time window is 44 days. Next, at that point, SVM has arrived at ideal exactness.

This study introduces an improved differential evolution with recurrent neural network based forecasting stock directions (FSD), named IDERNN-FSD model. The proposed model intends to identify the direction of SP changes effectively. The model follows a two-stage process namely prediction and hyperparameter tuning. For the effectual prediction of stock directions, recurrent neural network (RNN) model is applied. RNN has the benefit of taking data context into account during training, processing time series data, and being ideally suited for market prediction. because changes in stock prices frequently have a link to historical trends. As, the network will remember the prior data and use it to calculate the current output, which means that the nodes between the hidden layers are now connected and that the input of the hidden layer not only contains the output of the input layer. Additionally, it contains the concealed layer's prior output (Zhu, 2020). Followed by, the hyperparameters involved in the RNN model are optimally tuned by the design of improved differential evolution (IDE) algorithm. The remainder of this paper is structured as: Section 2 discusses some of related work. The proposed model is presented in Section 3. Section 4 provides the experimental results and discussion. Finally, Section 5 provides the conclusions.

2. Related work

Khan and others (Khan et al., 2022) utilized an algorithm on social network and financial news information to determine the effect of this information on SM predictive performance for ten following days. To optimize quality and performance of prediction, spam tweets reduction and FS are executed on the datasets. Furthermore, performed an experiment to discover this SM which is problematic to forecast and more inclined by financial news and social media. Hiransha and others (Hiransha et al., 2018) designed four kinds of DL method that is RNN, MLP, CNN and LSTM for predicting the SP of corporation according to the past prices.

Pang et al. (2020) presented the LSTM-NN with automated encoder and the deep LSTM with embedded layer to estimate the SM. In this algorithm, we employ the presented method, correspondingly, to vectorize the information, in a bid to prediction the stock through LSTM-NN. Some researchers (Nti et al., 2021) developed a multisource data-fusion SP predictive method-based hybrid DNN (CNN) and

(LSTM) called IKN-ConvLSTM. Exactly, we designed a prediction architecture for integrating stock-based data in 6 heterogeneous sources. Next, constructed a base method with CNN, and random searching method as a feature selective to optimize this first trained parameter. The authors (Hussain et al., 2022) presented a new prediction technique in the NN architecture based fuzzy time series, weighted average (WA), and induced ordered weighted average (IOWA). The presented method is very effectual when compared to present difficulty dealing with fuzzy time series predictive method and other conventional time series predictive techniques. The presented technique accommodates the weighted average, IOWA operator, and related amount of all the concepts in a certain issue for a fuzzy non-linear predictions. Authors (Houssein et al., 2022) presented a hybrid model that dependent upon the SVR technique with equilibrium optimizer (EO) to predict the closing price of Egyptian Exchange (EGX). In 3 indices are employed and modelled: EGX 50 EWI, EGX 30 and EGX 30 capped. The efficacy of the statistical measure and technical indicator in the prediction method is estimated.

A hybrid prediction model of extreme learning machine (ELM) with the differential evolution (DE) algorithm was suggested (Tang et al., 2020). The hybrid model used EEMD technology on the residual item after applying VMD technology to the original stock index price sequence to generate various modal components and residual items. The hybrid model then superimposed the DE-ELM model's prediction results for each residual item and the modal component to produce the final prediction outcomes. The empirical findings demonstrated that the suggested hybrid model had the greatest prediction performance across all prediction situations. Authors (Albahli et al., 2022) utilized autoencoder and 1D DenseNet to forecast closing stock prices based on ten years of Yahoo Finance data of ten notable stocks and STIs. In order to anticipate closing stock values for long-, medium-, and short-term perspectives, the 1D DenseNet's resulting features were then sent into the softmax layer. The experimental findings demonstrated that the suggested strategy surpassed the cutting-edge methodologies by reaching a minimal MAPE value of 0.41. Researchers (Saud & Shakya, 2020) compared the effectiveness of three deep learning approaches (Vanilla RNN, LSTM, and GRU) for forecasting stock prices of two most well-known and powerful listed banks on NEPSE. They investigated the parameter look-back period utilized in recurrent neural networks. According to the results of the study, GRU was the best at predicting stock prices. The study also recommended appropriate look-back time settings that might be employed with GRU and LSTM for improved performance in stock price prediction. A model that predicts future stock market prices for both the GOOGL and NKE asset was proposed using recurrent neural networks (RNN), particularly the Long-Short Term Memory model (LSTM). The results of the testing support the fact that the proposed model can track the change of opening prices for both assets (Moghar & Hamiche, 2020).

3. The proposed model

It is necessary to select a specific machine learning algorithm for creating an efficient prediction model. Because stock price values are time-series data, a Recurring Neural

Network (RNN) was selected for this study as its recurrent properties outperform all other machine learning algorithms at forecasting time-series data. This study has developed a novel model for effectual forecasting of stock directions. The presented model has determined the direction of SP changes efficiently. The proposed model has applied a RNN model for the effectual prediction of stock directions. Besides, the IDE algorithm is employed for optimally tuning the hyperparameters involved in the RNN.

Everyday market statistics for Apple Inc. (AAPL) are included in the historical stock dataset. The daily high, low, average, and closing stock values as well as the overall amount of sold stocks are among the characteristics. In this paper, stock statistics were obtained using the AAPL and FB. Two preprocessing stages are applied to the data: (i) data cleansing, which deals with absent and incorrect values; and (ii) data normalization, which improves the performance of machine learning models. To scale the stock price data to the range [0, 1], preprocessing and normalization are required for the stock price list to arrange data for more accurate forecast outcomes. The following equation is used:

$$Z = \frac{(X - Xmin)}{(Xmax - Xmin)}$$

where x is the value of price related variable (*open*, *close*, *high*, *low*, …), min and max are the minimum and maximum of a given time, and Z is the normalized variable value.

The data set is normalized in this flow chart so that the data can be trained. The suggested IDERNN model with historical data as input is used as the benchmark to examine the effects of technical indicators on stock price trend prediction. For the purpose of the experiment in this paper, the dataset is split into a training set and a test set. For model training, the first 70% of the examples are used, and the final 30% are used for model verification. The threshold for early stopping is 35 epochs, which indicates that the training is terminated if the loss of the validation set does not improve after 35 epochs. In the validation group, the model is saved with the lowest loss for testing. For the best outcome, certain crucial parameters need to be fine-tuned. The IDERNN setup is followed by the definition of placeholders to store the appropriate values, an LSTM cell to handle short-term and long-term memory, and dynamic RNN and loss to determine the Mean Squared Error (MSE). After analyzing the MSE, the model chooses whether to repeat the session in order to fine-tune the RNN setup's parameters or run it one more to obtain the final result for the anticipated direction. The model can forecast the stock direction very precisely and reliably after the sufficient iterations.

The first stage in the proposed IDERNN is to initialize the population and assign a value per each chromosome. The network layers number, the hidden neurons number, the chromosomes genes, and the RNN iterations number are all factors that influence neural networks. The neural network is then trained to compute the fitness value. As a fitness function, model prediction accuracy is utilized. Adaptive crossover and mutation procedures on solved individuals result in the creation of new individuals. Repeat the preceding stages until the maximum iterations number is reached, and
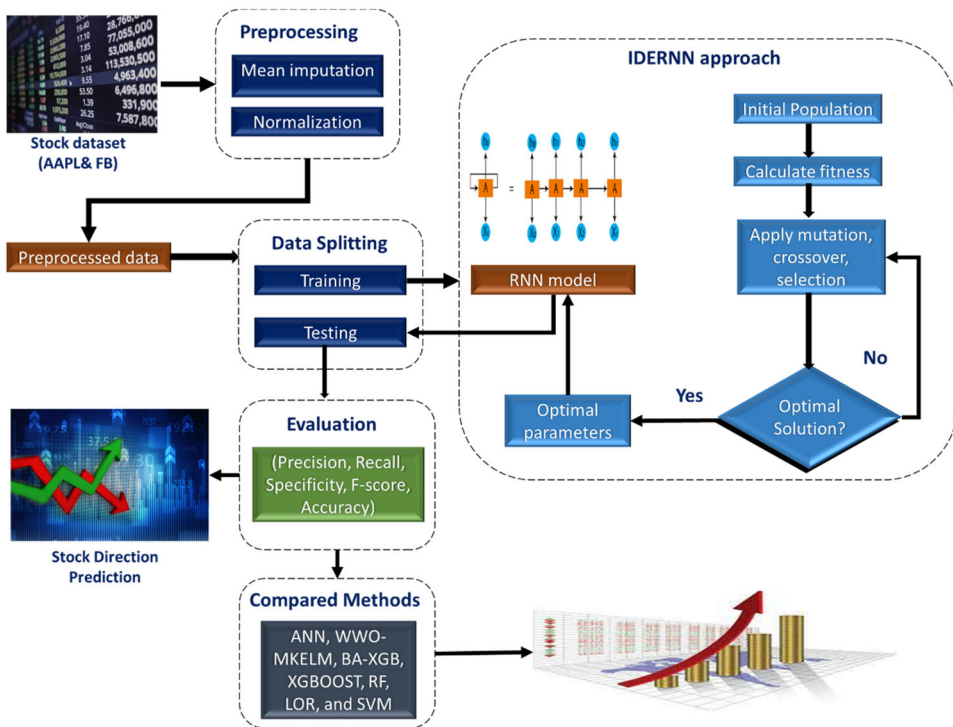
Figure 2. Proposed IDERNN-FSD methodology.
Source: the Authors.

the best individual emerges from among them. Finally, the RNN neural network is trained with the optimal hyper-parameter combination. The predictions on training and test sets are produced and print the root mean square error on both sets and output prediction is compared with true values. The evaluation metrics in this paper are accuracy, F score, recall, precision, and specificity. To demonstrate the improved results of the proposed method, a series of experiments on AAPL and FB datasets highlighted the supremacy of the suggested model over the other techniques, e.g., Artificial Neural Network (ANN), water wave optimization with multi-kernel extreme learning machine (WWO-MKELM), Boosting Algorithm- eXtreme gradient boosting (BA-XGB), XGBOOST, Logistic Regression (LOR), Random Forest (RF), and Support Vector Machine (SVM) Algorithm. The exact steps of the proposed approach are depicted in Figure 2.

3.1. Module I: RNN based forecasting model

Primarily, the RNN model can be employed for forecasting the direction of SP movements. RNN is a type of ANN which extend the convention FFNN with loop from the connection. Different FFNN, an RNN is ability to procedure the sequential input with containing a recurrent hidden state (HS) if activation at all the steps dependent upon that preceding step (Lin et al., 2021). During this approach, the network is demonstrated dynamic temporal behavior. Figure 3 demonstrates the framework of RNN.
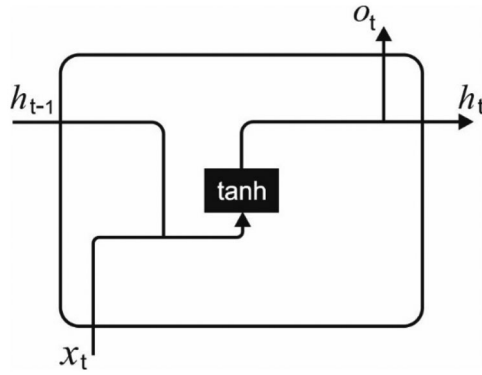
Figure 3. Structure of RNN.
Source: the Authors.

To provide a sequence data $x = (x_1, x_2, \ldots, x_T)$, whereas x_i refers the data at i^{th} time step, an RNN upgrades their recurrent HS h_t as:

$$h_t = \begin{cases} 0, & if \ t = 0 \\ \phi(h_{t-1}, x_t), & otherwise \end{cases} \tag{1}$$

In which, ϕ represents the non-linear function as hyperbolic tangent function or logistic sigmoid function. The RNN could have an output $y = (y_1, y_2, \ldots, y_T)$ optionally. For some tasks like hyperspectral image classifier, it can be required only one outcome, i.e., y_T. During the typical RNN technique, the upgrade principle of recurrent HS in Equation (1) has generally executed as:

$$h_t = \phi(Wx_t + Uh_{t-1}) \tag{2}$$

whereas W and U are the co-efficient matrices to the input at the current stage and to the activation of recurrent hidden unit at preceding stage correspondingly. Actually, an RNN is method of probability distribution for the following element in a sequence of data, provided their current state h_t, by capture a distribution on sequence data of variable length. Assume $p(x_1, x_2, \ldots, x_T)$ be the sequence probability that is decomposed as:

$$p(x_1, x_2, \ldots, x_T) = p(x_1) \cdots p(x_T | x_1, \ldots, x_{T-1}). \tag{3}$$

Afterward, all the conditional probability distribution is demonstrated with the following recurrent network.

$$p(x_t | x_1, \ldots, x_{t-1}) = \phi(h_t) \tag{4}$$

In which, h_t has attained in Equations (1) and (4). Our stimulus in this case is obvious at this point: a hyperspectral pixel performs as sequential data rather than feature vector, and therefore recurrent network is implemented for modeling the spectral sequence.

3.2. Module II: IDE based hyperparameter optimizer

For the RNN technique's hyperparameters to be tuned as well as possible, the IDE is employed to it. Differential Evolution (DE) is a metaheuristic approach depending upon group searching. This is identical to the genetic algorithm (GA). It is depending upon three stages: selection, mutation, and crossover. But the variance is that the mutation stage of DE is depending upon distinct strategies. The approach has higher performance and could efficiently jump out of local optimization (Deng, Shang, et al., 2021; Deng, Xu, et al., 2021). The mutation is the vital stage to differentiate DE from the GA. The major portion is the variance vector, comprised of the vector variation of two distinct random vectors, i.e., later superimposed by the separate vector to attain a vector with volatility:

$$V_i^t = X_{r1}^t + F \cdot \left(X_{r2}^t - x_{r3}^t \right) \tag{5}$$

where X_{r1}^t, X_{r2}^t, and X_{r3}^t shows three distinct individuals in the population and F indicates the difference factors. The common value range is 0-2, that control the size of variance vector.

Cross-function on the population maintains the variety of the population:

$$u_i^t = \begin{cases} \nabla_{ij}^t, & CR \geq rand(0,1) \ or \ j = rand(1,n) \\ X_{ij}^t, & CR \leq rand(0,1) \ and \ j \neq rand(1,n) \end{cases} \tag{6}$$

where, $rand()$ denotes an arbitrary value within $[0, 1]$; $(0, n)$ denotes an arbitrary value among 0 and n; CR indicates the crossover factor, i.e., commonly among 0 and 1. The values of CR affect the optimization of method. Once the CR values get larger, the crossover possibility of the approach gets larger, the globality is stronger, and the variety of the population is maintained. Once the CR value is smaller, the crossover possibility of the approach is smaller making it easier to the approach for falling into local optimization.

The selection operation of DE approach is accomplished according to the greedy approach. With the comparison of fitness initialized individual of mutation and crossover individual, the optimum individual is carefully chosen for entering the second iteration:

$$X_i^{t+1} = \begin{cases} X_i^t, f\left(X_i^t\right) < f\left(u_i^t\right) \\ u_i^t, \ otherwisw \end{cases} \tag{7}$$

The IDE basic steps:

Step 1: Set the relevant parameters to their initial values. For instance, the population size is set to S, the mutation factor is set to F, the cross factor is set to CR, the spatial dimension is set to D, and the evolutionary algebra is set to $t = 0$.

Step 2. Set up the parent population: X(t)= $\{x_1^t, \ x_2^t, \ \ldots \ldots x_s^t\}$, $x_i^t = \left(x_{i1}^t, \ x_{i2}^t, \ \ldots, x_{iD}^t \right)^T$

Step 3: Determine the fitness value for every individual which is the model's objective function.

Step 4. Carry out the mutation procedure: the mutant individuals are produced by the difference between the parental individuals x_i^t, $v_i^t = (v_{i1}^t, \ v_{i2}^t, \ \ldots, v_{iD}^t)^T$. Equation (5) shows the mutation function.

Step 5. Perform Crossover operation: a crossover procedure is performed between the mutant individual v_i^t and the parent individual x_i^t to develop a mixed individual, $u_i^t = (u_{i1}^t, \ u_{i2}^t, \ \ldots, u_{iD}^t)^T$. Equation (6) depicts the crossover process.

Step 6. Selection process: The differential evolution method retains great individuals while eliminating inferior ones through continual development. As the next generation individual $x_i^{t+1} = \{x_1^{t+1}, \ x_2^{t+1}, \ \ldots \ldots x_s^{t+1}\}$, it chooses the best one with the highest fitness value from the parent individual x_i^t and the mixed individual u_i^t. The generations process is shown in Equation (7).

Step 7. Iteration termination: If the error criterion is fulfilled or the maximum iterations number in the created next-generation population is achieved, When the iteration is finished, the iteration is halted, and the best individual is output. Otherwise, activities such as mutation, crossover, and selection proceed until the iteration stop condition is reached.

The population created by initialized DE population has the disadvantage of lower randomness and ergodicity. The study employs an IDE algorithm with Henon map for initializing the DE population. The Henon map has higher ergodicity and is appropriate for the DE population initialization. It can be expressed in the following:

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases} \tag{8}$$

Whereas a and b denotes the system control parameter, χ and y denotes the input variable, and n indicates amount of iterations.

3.3. Construction of hybrid IDERNN model

To train the model, the data set must be split into two parts: training and testing. The training phase provides the foundation for the procedure's capacity to generalize, which is assessed by performance on the testing set. The DE method is used to optimize the input of the RNN model in order to enhance the training stability of the RNN model. Figure 4. Presented the structure of the hybrid IDERNN framework.

The specific steps of DEELM model

Step 1. The population of the RNN model is initialized by encoding the random input.

Step 2. Initialize the DE algorithm's pertinent parameters. The mutation factor is set to 0.5, crossover factor is set to 0.9, maximum iteration number is set to 30, each individual's dimension is 245 and the population size is set to10.

Step 3. The fitness value is computed for each individual in the population to arrive the best accuracy of the predicted output of the RNN model.

Step 4. Sequential mutation, crossover, and selection processes are carried out, together with an iterative termination check. The optimal input is output if the
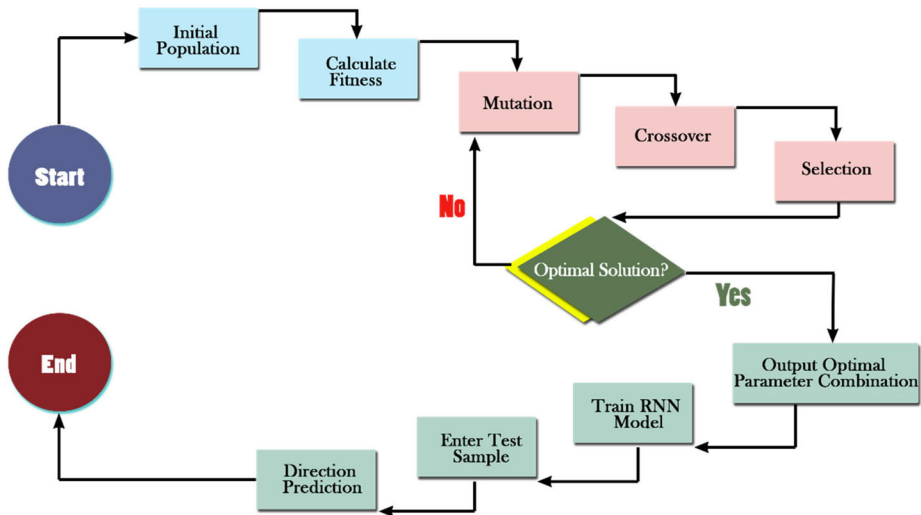
Figure 4. Architecture of hybrid IDERNN framework.
Source: the Authors.

termination condition is met; if not, the iteration is maintained until the optimal output is attained.

Step 5. Step 5: The RNN model's output is set to the ideal individuals optimized by the DE method, yielding the optimized and enhanced proposed DE-RNN model.

The purpose of optimization in the preceding phases of optimizing the RNN model is to maximize the accuracy as a function of fitness to enhance the overall prediction accuracy of the hybrid IDE-RNN model. The specific formula of accuracy can be found in Equation (13).

4. Results and discussion

The confusion matrix contains many equations that are used to assess the model's performance. In this work, we will assess the blended ensemble model using accuracy, recall, and F1-score. Precision measures how accurate the model is in terms of positive predictions by assessing how many anticipated positives are really positive. In this paper precision, F-score, recall, specificity, and accuracy are utilized to measure the prediction performance. These parameters can be defined as:

$$F_{score} = 2* \frac{(Prec_n * Reca_l)}{(Prec_n + Reca_l)} \tag{9}$$

$$Spec_y = \frac{TN}{TN + FP} \tag{10}$$

$$Prec_n = \frac{TP}{TP + FP} \tag{11}$$

Table 1. Various performance measures and trading windows of the proposed IDERNN-FSD.

Company Name	Trading Window	$Accu_y$	$Reca_l$	$Prec_n$	$Spec_y$	F_{Score}
AAPL Stock	3	70.70	77.13	72.86	62.62	73.29
	5	78.75	84.98	79.02	61.90	79.76
	10	83.31	87.84	88.41	82.43	86.60
	15	87.85	89.60	88.71	83.08	89.07
	30	90.35	93.14	92.10	85.82	91.89
	60	95.71	99.25	96.46	91.79	97.81
	90	99.02	99.15	99.28	98.30	98.97
FB Stock	3	71.15	77.58	71.47	67.80	76.35
	5	78.16	89.50	73.62	65.25	81.60
	10	85.07	95.36	85.37	73.66	88.68
	15	89.97	94.80	94.25	85.52	93.78
	30	91.90	99.47	96.53	95.32	97.30
	60	91.95	99.12	93.31	94.37	98.29
	90	98.71	99.24	98.28	96.41	99.34

Source: the Authors.

$$Reca_l = \frac{TP}{TP + FN} \tag{12}$$

$$Accu_y = \frac{TP + TN}{TP + TN + FN + FP}$$
$$= \frac{n_{correct}}{N} * 100\% \tag{13}$$

where TP denotes True Positive, TN indicates True Negative, FP refers False Positive, and FN is False Negative. $n_{correct}$ signifies the number of training days on which the anticipated price movement is right and N represents the total number of trading days.

In this section, the performance analysis of the IDERNN-FSD method is validated using two company stocks namely Apple's (AAPL) and FB. In all the experiments we performed, the trading window was set to 3, 5, 10, 15, 30, 60, and 90 days. Table 1 examines the prediction outcome of proposed technique with different measures and trading windows (TWs) on two datasets such as AAPL and FB stock datasets.

Figure 5 shows the prediction outcomes of the proposed model with AAPL stock dataset with distinct TWs. The figure reported that the proposed model has resulted to maximum outcome under every TW. For instance, with TW = 3, the proposed model has provided u_y, $reca_l$, $prec_n$, $spec_y$, and F_{score} of 70.70%, 77.13%, 72.86%, 62.62%, and 73.29% respectively. Along with that, with TW = 10, the proposed model has provided u_y, $reca_l$, $prec_n$, $spec_y$, and F_{score} of 83.31%, 87.84%, 88.41%, 82.43%, and 86.60% respectively. Moreover, with TW = 30, the proposed model has provided u_y, $reca_l$, $prec_n$, $spec_y$, and F_{score} of 90.35%, 93.14%, 92.10%, 85.82%, and 91.89% respectively. Furthermore, with TW = 90, the proposed model has provided u_y, $reca_l$, $prec_n$, $spec_y$, and F_{score} of 99.02%, 99.15%, 99.28%, 98.30%, and 98.97% correspondingly.

Figure 6 shows the prediction outcomes of the suggested model with FB stock dataset with distinct TWs. The figure reported that the proposed model has resulted to maximum outcome under every TW. For instance, with TW = 3, the proposed model has provided u_y, $reca_l$, $prec_n$, $spec_y$, and F_{score} of 71.15%, 77.58%, 71.47%, 67.80%, and 76.35% respectively. Along with that, with TW = 10, the proposed model has provided u_y, $reca_l$, $prec_n$, $spec_y$, and F_{score} of 85.07%, 95.36%, 85.37%, 73.66%, and 88.68% respectively. Moreover, with TW = 30, the proposed model has provided u_y, $reca_l$,
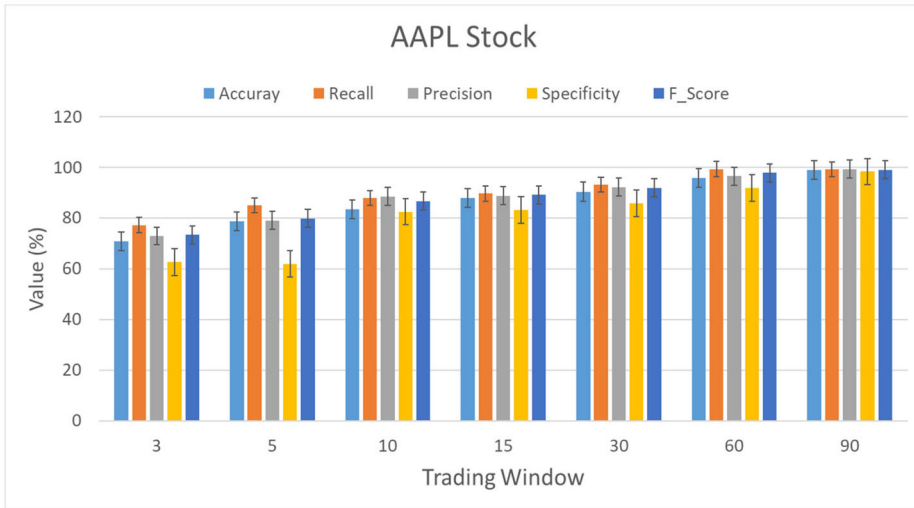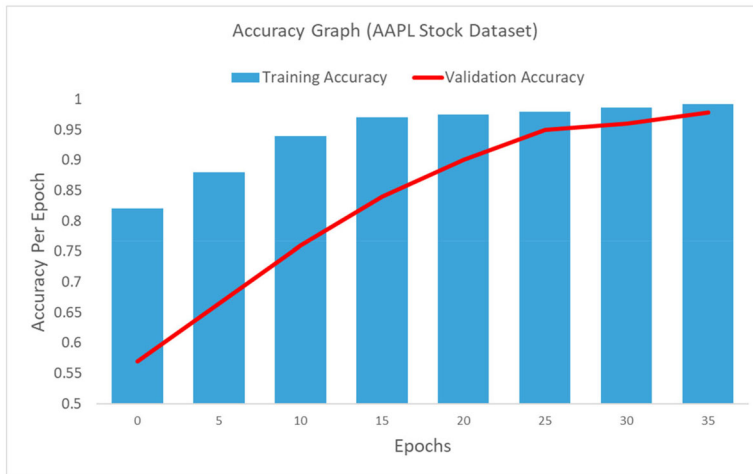
Figure 5. Result analysis of IDERNN-FSD technique on AAPL stock dataset.
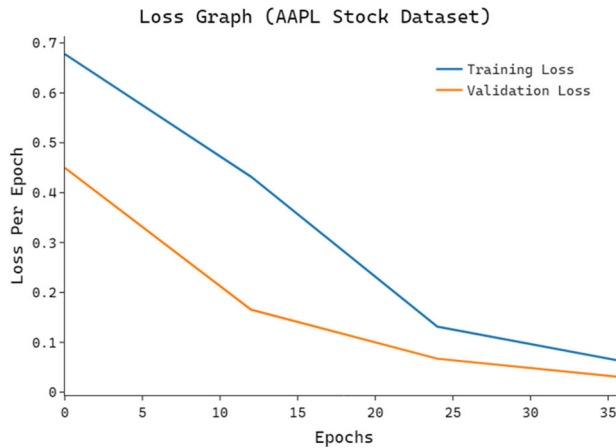Source: the Authors.
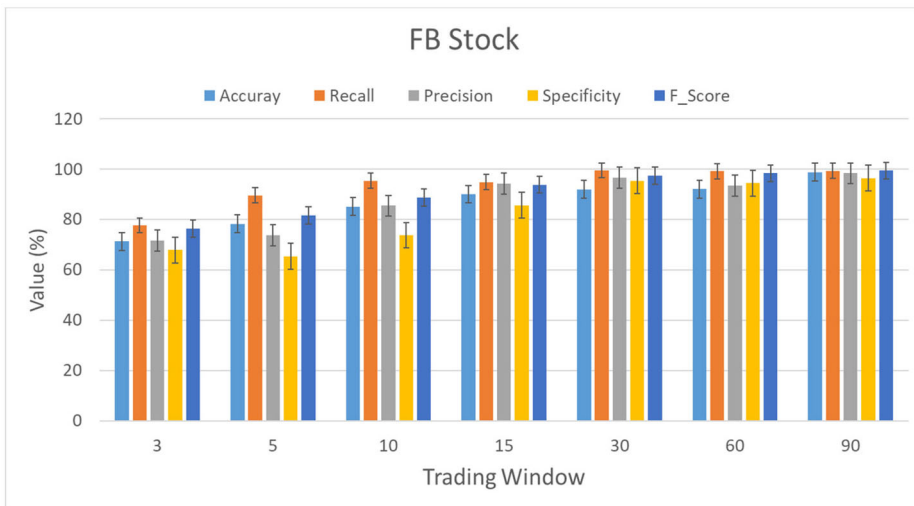
Figure 6. Result analysis of IDERNN-FSD technique on FB stock dataset.
Source: the Authors.

$prec_n$, $spec_y$, and F_{score} of 91.90%, 99.47%, 96.53%, 95.32%, and 97.30% respectively. Furthermore, with TW = 90, the proposed model has provided u_y, $reca_l$, $prec_n$, $spec_y$, and F_{score} of 98.71%, 99.24%, 98.28%, 96.41%, and 99.34% respectively.

On the basis of the AAPL stock dataset, Figure 7 verifies the proposed model's accuracy evaluation. The findings showed that the proposed model has the potential to achieve higher training and validation accuracy values. It is clear that the results for validation accuracy are somewhat greater than those for training accuracy. Figure 8 reports a brief training and validation loss provided by the proposed model using the test AAPL stock dataset. Results showed that the proposed model has been able to minimize training values and validation losses using data from the AAPL stock.
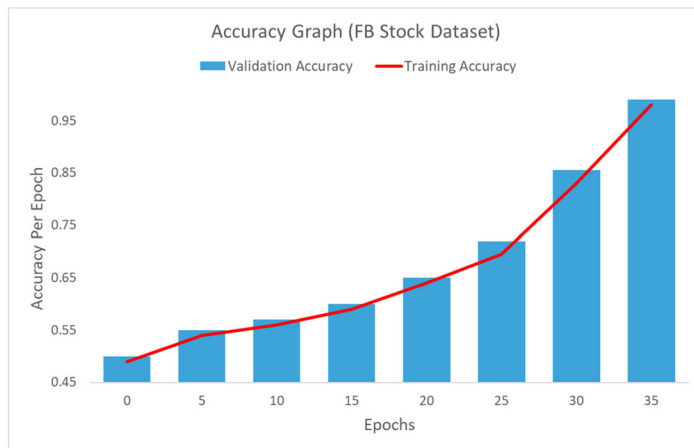
Figure 7. Accuracy analysis of IDERNN-FSD technique on AAPL stock dataset.
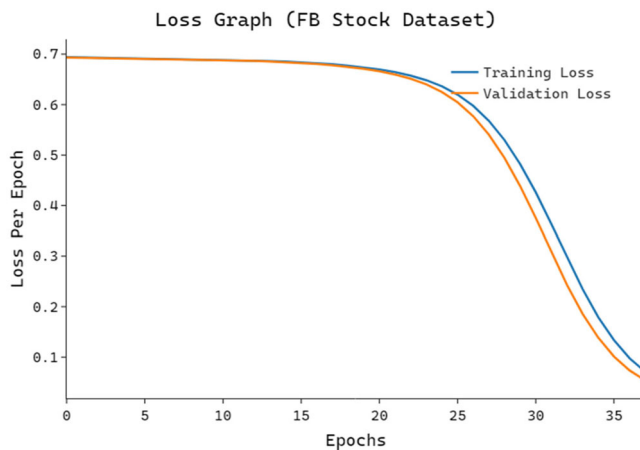Source: the Authors.

Figure 8. Loss analysis of IDERNN-FSD technique on AAPL stock dataset.
Source: the Authors.

On the basis of the FB stock dataset, Figure 9 verifies the accuracy evaluation of the proposed technique. The findings showed that the proposed model has the potential to achieve higher training and validation accuracy values. It is clear that the results for validation accuracy are somewhat greater than those for training accuracy. Figure 10 reports a brief training and validation loss provided by the proposed model using the test FB stock dataset. The results showed that, using the FB stock dataset, the proposed model was able to achieve lowest values of training and validation losses.

A series of experiments on AAPL and FB datasets highlighted the supremacy of the proposed model over the other techniques, e.g., Artificial Neural Network (ANN), water wave optimization with multi-kernel extreme learning machine (WWO-MKELM), Boosting Algorithm- eXtreme gradient boosting (BA-XGB), XGBOOST, Random Forest (RF), Logistic Regression (LOR), and Support Vector Machine (SVM) Algorithm in Table 2 and Figure 11 (Basak et al., 2019; Jeyakarthic & Punitha, 2020).

Figure 9. Accuracy analysis of IDERNN-FSD technique on FB stock dataset.
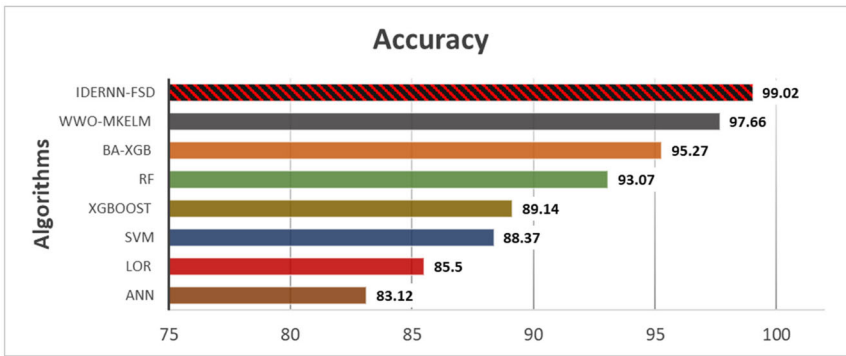Source: the Authors.

Figure 10. Loss analysis of IDERNN-FSD technique on FB stock dataset.
Source: the Authors.

Table 2. Comparative analysis of IDERNN-FSD technique with existing Methods.

Algorithms	Accuracy
Artificial Neural Network	83.12
LOR Algorithm	85.50
Support Vector Machine Algorithm	88.37
XGBOOST Algorithm	89.14
Random Forest Algorithm	93.07
BA-XGB Algorithm	95.27
WWO-MKELM Algorithm	97.66
Proposed Model	**99.02**

Source: the Authors.

The results indicated that the LOR and ANN models have obtained ineffective accuracy of 85.50% and 83.12% respectively. Followed by, the XGBOOST and SVM models have reached to slightly increased accuracy of 89.14% and 88.37% correspondingly. Also, the BA-XGB and RF approaches have reached to moderately improved

Figure 11. Comparative analysis of IDERNN-FSD technique with existing algorithms.
Source: the Authors.

accuracy of 95.27% and 93.07% respectively. Along with that, the WWO-MKELM model has resulted to reasonable accuracy of 97.66%. However, the proposed model has showcased maximum accuracy of 99.02%.

5. Conclusion

The reduction in prediction error rate significantly decreases the risk in investment process. This study has developed a novel model for effectual forecasting of stock directions. The proposed model has determined the direction of SP changes efficiently. The proposed model has applied a RNN model for the effectual prediction of stock directions. Besides, the IDE algorithm is employed for adjusting the RNN's hyperparameters in the best possible way. The exploitation of the IDE algorithm helps in accomplishing maximum stock direction prediction outcomes. For demonstrating the improved results of the proposed method, a series of experiments on benchmark datasets highlighted the supremacy of the proposed model over the other techniques with accuracy of 99.02%. It is also clear that the loss rate is essentially steady after the epoch 35 and reaches about 0.1%. In our upcoming work, we'll concentrate more on feature selection and engineering, particularly the use of technical indicators for stock forecasting. Many different features can be produced, and we can choose the more important technical features to enhance the performance of the stock prediction. Also, we'll think about enhancing the forecasting precision by utilizing other certain open datasets. Additionally, we will try estimate the performance by using parallel and incremental methodologies.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Noura Metawa http://orcid.org/0000-0002-2328-0813
Mohamed Elhoseny http://orcid.org/0000-0001-9659-1551

References

Albahli, S., Nazir, T., Mehmood, A., Irtaza, A., Alkhalifah, A., & Albattah, W. (2022). AEI-DNET: A novel densenet model with an autoencoder for the stock market predictions using stock technical indicators. *Electronics*, *11*(4), 611. https://doi.org/10.3390/electronics11040611

Ananthi, M., & Vijayakumar, K. (2021). Stock market analysis using candlestick regression and market trend prediction (CKRM). *Journal of Ambient Intelligence and Humanized Computing*, *12*(5), 4819–4826. https://doi.org/10.1007/s12652-020-01892-5

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance*, *47*, 552–567. https://doi.org/10.1016/j.najef.2018.06.013

Deng, W., Shang, S., Cai, X., Zhao, H., Song, Y., & Xu, J. (2021). An improved differential evolution algorithm and its application in optimization problem. *Soft Computing*, *25*(7), 5277–5298. https://doi.org/10.1007/s00500-020-05527-x

Deng, W., Xu, J., Song, Y., & Zhao, H. (2021). Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem. *Applied Soft Computing*, *100*, 106724. https://doi.org/10.1016/j.asoc.2020.106724

Freij, A., Walid, K., & Mustafa, M. (2021). Deep learning model for digital sales increasing and forecasting: Towards smart E-commerce. *Journal of Cybersecurity and Information Management*, *8*(1), 26–34. https://doi.org/10.54216/JCIM.080103

Goh, T. S., Henry, H., & Albert, A. (2021). Determinants and prediction of the stock market during COVID-19: Evidence from Indonesia. *The Journal of Asian Finance, Economics and Business*, *8*(1), 1–6.

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

Houssein, E. H., Dirar, M., Abualigah, L., & Mohamed, W. M. (2022). An efficient equilibrium optimizer with support vector regression for stock market prediction. *Neural Computing and Applications*, *34*(4), 3165–3200. https://doi.org/10.1007/s00521-021-06580-9

Hussain, W., Merigó, J. M., & Raza, M. R. (2022). Predictive intelligence using ANFIS-induced OWAWA for complex stock market prediction. *International Journal of Intelligent Systems*, *37*(8), 4586–4611. https://doi.org/10.1002/int.22732

Javed Awan, M., Mohd Rahim, M. S., Nobanee, H., Munawar, A., Yasin, A., & Zain, A. M. (2021). Social media and stock market prediction: A big data approach. *Computers, Materials & Continua*, *67*(2), 2569–2583. https://doi.org/10.32604/cmc.2021.014253

Jeyakarthic, M., & Punitha, S. (2020). An effective stock market direction prediction model using water wave optimization with multi-kernel extreme learning machine. *IIOAB J*, *11*, 103–109.

Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2022). Stock market prediction using machine learning classifiers and social media, news. *Journal of Ambient Intelligence and Humanized Computing*, *13*(7), 3433–3456. https://doi.org/10.1007/s12652-020-01839-w

Lin, J. C.-W., Shao, Y., Djenouri, Y., & Yun, U. (2021). ASRNN: A recurrent neural network with an attention model for sequence labeling. *Knowledge-Based Systems*, *212*, 106548. https://doi.org/10.1016/j.knosys.2020.106548

Metawa, N., & Mutawea, M. (2022). Multi-objective decision making model for stock price prediction using multi-source heterogeneous data fusion. *Fusion: Practice and Applications*, *9*(1), 59–69. https://doi.org/10.54216/FPA.090105

Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science*, *170*, 1168–1173. https://doi.org/10.1016/j.procs.2020.03.049

Mokhtari, S., Yen, K. K., & Liu, J. (2021). Effectiveness of artificial intelligence in stock market prediction based on machine learning. *ArXiv Preprint ArXiv:2107.01031*

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2021). A novel multi-source information-fusion predictive framework based on deep neural networks for accuracy enhancement in stock market prediction. *Journal of Big Data*, 8(1), 1–28. https://doi.org/10.1186/s40537-020-00400-y

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

Pustokhin, D. A., & Pustokhina, I. V. (2022). Statistical machine learning model and commodity futures volatility information for Financial Stock Market Forecasting. *American Journal of Business and Operations Research*, 7(2), 32–40. https://doi.org/10.54216/AJBOR.070203

Saud, A. S., & Shakya, S. (2020). Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE. *Procedia Computer Science*, 167, 788–798. https://doi.org/10.1016/j.procs.2020.03.419

Shilpa, B. L., & Shambhavi, B. R. (2021). *Combined deep learning classifiers for stock market prediction: Integrating stock price and news sentiments*. Kybernetes.

Tang, Z., Zhang, T., Wu, J., Du, X., & Chen, K. (2020). Multistep-ahead stock price forecasting based on secondary decomposition technique and extreme learning machine optimized by the differential evolution algorithm. *Mathematical Problems in Engineering*, 2020, 1–13. https://doi.org/10.1155/2020/2604915

Thakkar, A., & Chaudhari, K. (2021). Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions. *An International Journal on Information Fusion*, 65, 95–107. https://doi.org/10.1016/j.inffus.2020.08.019

Tuarob, S., Wettayakorn, P., Phetchai, P., Traivijitkhun, S., Lim, S., Noraset, T., & Thaipisutikul, T. (2021). DAViS: A unified solution for data collection, analyzation, and visualization in real-time stock market prediction. *Financial Innovation*, 7(1), 1–32. https://doi.org/10.1186/s40854-021-00269-7

Zhu, Y. (2020). Stock price prediction using the RNN model. *Journal of Physics: Conference Series*, 1650(3), 032103. https://doi.org/10.1088/1742-6596/1650/3/032103