Journal of Mathematics and Modeling in Finance (JMMF)
Vol. 3, No. 2, Summer & Autumn 2023
Research paper

A new hybrid method of dynamic mode decomposition and long short-term memory for financial market forecasting

Roya Karimkhani[1], **Yousef Edrisi Tabriz**[2], **Ghasem Ahmadi**[3]

[1] Department of Mathematics, Payame Noor University, Tehran, Iran
karimkhani@student.pnu.ac.ir

[2] Department of Mathematics, Payame Noor University, Tehran, Iran
yousef_edrisi@pnu.ac.ir

[3] Department of Mathematics, Payame Noor University, Tehran, Iran
g.ahmadi@pnu.ac.ir

Abstract:
Forecasting price trends in financial markets is of particular importance for traders because price trends are inherently dynamic and forecasting these trends is complicated. In this study, we present a new hybrid method based on combination of the dynamic mode decomposition method and long short-term memory method for forecasting financial markets. This new method is in this way that we first extract the dominant and coherent data using the dynamic mode decomposition method and then predict financial market trends with the help of these data and the long short-term memory method. To demonstrate the efficacy of this method, we present three practical examples: closing price of US Dollar to Iranian Rial, closing prices of zob roy Isfahan stock, and also closing prices of siman shargh stock. These examples exhibit bullish, bearish, and neutral behaviors, respectively. It seems that the proposed new method works better in predicting the financial market than the existing long-short-term memory method.

Keywords: Dynamic mode decomposition, Long short-term memory, Financial market forecasting.
Classification: 37E99, 68T07, 91G15.

1 Introduction

Modeling and forecasting the price of the financial market is one of the challenging goals of researchers due to its importance and significant impact on the countries economic progress. The purpose of using the price prediction algorithm is to suggest a set of financial market trends that maximizes the investors return. The investor has a finite amount of money and wants to maximize her or his return on investment. Algorithms based on mathematics predict the behavior of the financial market

[2]Corresponding author
Received: 24/07/2023 Accepted: 20/10/2023
https://doi.org/10.22054/JMMF.2023.74998.1099

according to its history. This could then be used to select a portfolio of financial markets with some idea of future performance.

Long short-term memory (LSTM) is a type of neural network for ordinal data that emerged in 1997 to develop the recurrent neural network (RNN) [7] and was later improved in 2000 by Felix Gers and others [2]. Short-term memory refers to the internal states of cells, and long-term memory refers to learned weights [7]. LSTM network can detect time dependencies and also the ability to detect complex transactions of financial data [15]. The LSTM network was used to predict the price in the short term, which shows that the LSTM prediction was better than the random prediction, but it needs many improvements [18].

The seminal work done by Bernard Koopman in 1931 on nonlinear dynamical systems is where the dynamical mode decomposition (DMD) approach began [9]. However, theoretical developments were limited mainly due to the lack of computer resources at the time. Pietraschmid and others first defined DMD as an algorithm in 2008 and 2010, respectively [13, 14]. In 2013, Jonathan et al. presented a paper titled dynamic mode decomposition (Theory and Applications), which presents a theoretical framework that extends DMD to non-sequential time series [16]. In 2016, Jordan Mann and his colleagues used the DMD algorithm for financial trading strategies [11]. Nathan Kotz et al. developed the essential theoretical foundations of DMD and the Koopman operator [10]. They also created many innovations and new algorithms that expanded the scope of applications of the method. In 2019, Steven Brunton and his colleagues published a book entitled Machine Learning and dynamic systems and control on the application of the dynamic mode decomposition algorithm to predict, identify, and control dynamic systems [1].

The DMD algorithm decomposes state information and geographical and temporal patterns to comprehend global dynamic behavior. Then it uses these states to reveal coherent and dominant structures in the data. This algorithm is an equation-free and this is one of the its best features. In fact, in this method, instead of problem modeling, we use data-driven methods to predict the future states.

The aim of this study is to increase the accuracy of predicting financial market trend by combining DMD algorithm with LSTM network. Our proposed method utilizes the DMD algorithm to extract coherent and dominant structures from the financial market data, which are then input into the LSTM network for prediction. We evaluate the performance of our DMD-LSTM method by investing for some time and calculating the percentage of profit and loss. We compare the profitability and forecasting improvement of our method with the traditional LSTM method and also evaluate the results with real financial market data.

The written order of the paper is as follows. In section 2, DMD and LSTM are described. In section 3, a new method based on the combination of DMD and LSTM methods is introduced. Next, in section 4, we examine three cases to demonstrate the performance of the LSTM and DMD-LSTM approaches. Finally, section 5 presents the conclusion based on the performance comparison of the two methods.

2 Prerequisites

This section begins with a detailed description of the dynamic mode decomposition method used to extract dominant and coherent financial market data. Subsequently, we describe the long-short term memory method used for prediction.

2.1 Dynamic mode decomposition

Decomposition of matrices in linear algebra has been of interest to researchers for a long time. The intended applications of matrix decomposition have caused increasing attention to it. In the sciences related to matrices, it is crucial to emphasize the significance of identifying similar matrices as a means to simplify intricate computations. Here we want to examine the method of dynamic mode decomposition, which reduces the computation significantly by finding a matrix with dimensions smaller than the original matrix.

Here we are going to simulate a dynamic system with the help of DMD method and then predict its future states. For this purpose, it is necessary to have data from this dynamic system at different times, which we obtain by taking snapshots of this system. DMD is data-driven and requires the collection of pairs of system snapshots in the first step. The time step Δt is short enough to produce the highest dynamic frequencies and these snapshot pairs can be expressed as $\{x(t_j), x(t'_j)\}_{j=1}^{m}$.

A snapshot, like previously, might represent the state of a system. For instance, given a three-dimensional fluid velocity field collected at several discretized sites, a high-dimensional column vector may be produced [17]. Then, these photos are arranged in the form of two data matrices X and X' as follows

$$X = \begin{bmatrix} x(t_1) & x(t_2) \cdots & x(t_m) \end{bmatrix}, \tag{1}$$

$$X' = \begin{bmatrix} x(t'_1) & x(t'_2) \cdots & x(t'_m) \end{bmatrix}. \tag{2}$$

The DMD approach seeks the leading spectral decomposition (i.e., eigenvalues and eigenvectors) of the best-fit linear operator A that joins the two snapshot matrices in time

$$X' = AX. \tag{3}$$

The best linear operator A for a dynamical system produces the best forward motion in time by measuring snapshots. If we assume uniform sampling across time, this becomes

$$X_{j+1} \approx AX_j. \tag{4}$$

The optimal A matrix is provided by

$$A \approx X'X^{\dagger}, \tag{5}$$

where $X^\dagger$ is the MoorePenrose pseudoinverse matrix of X [3].

The DMD approach does not need any explicit computations utilizing A directly; instead, it makes use of dimensionality reduction to determine the dominating eigenvalues and eigenvectors of A. Specifically, the singular value decomposition (SVD) of the matrix X is used to compute the pseudoinverse $X^\dagger$ in equation (4). There are normally m nonzero singular values and accompanying singular vectors since this matrix contains $m \ll n$ columns and n rows. The matrix A will thus have a maximum rank of m. We compute A projection onto these leading singular vectors rather of computing A directly, which produces a tiny matrix $\tilde{A}$ with a dimension of no more than $m \times m$.
The reduced matrix $\tilde{A}$ and the data matrix X approximate high-dimensional DMD modes (eigenvectors A) without using A. In certain circumstances, the DMD algorithm can be used to approximate the eigenvectors of a complete matrix A. There are two common expressions for the DMD algorithm, the first one was introduced by Schmid [14] and the second one, called the exact DMD algorithm, was introduced by Tu et al [16]. We describe the steps of the exact DMD algorithm in Algorithm 2.1.

The DMD algorithm

(i) Take the reduced SVD of X

$$X \approx U\Sigma V^*, \tag{6}$$

where $U \in \mathbb{C}^{n\times r}$, $\Sigma \in \mathbb{C}^{r\times r}$, $V \in \mathbb{C}^{m\times r}$ and $r \leqslant m$ specifies the X data matrix's SVD approximation rank.

(ii) Construct the low-dimensional matrix $\tilde{A}$ which models the dynamic system linearly. To do this, first calculate the pseudoinverse of X to get the entire matrix A

$$A = X'V\Sigma^{-1}U^*. \tag{7}$$

But we only care about A leading r eigenvalues and eigenvectors, so we may project A onto U on the proper orthogonal decomposition (POD) modes as follows

$$\tilde{A} = U^*AU = U^*X'V\Sigma^{-1}. \tag{8}$$

The critical finding is that both the complete matrix A and the reduced matrix $\tilde{A}$ have nonzero eigenvalues.

(iii) Now, instead of A, find the eigenvalues and eigenvectors of $\tilde{A}$

$$\tilde{A}W = W\Lambda. \tag{9}$$

Therefore, the eigenvalues of the matrix A are represented by the elements of the diagonal matrix Λ. The diagonalization of the matrix is accomplished via the columns of W, which are eigenvectors of $\tilde{A}$.

(iv) In accordance with the time-shifted snapshot matrix X' and the reduced system's eigenvectors W, the high-dimensional DMD modes Φ are rebuilt

$$\Phi = X' V \Sigma^{-1} W. \tag{10}$$

As a result, approximated answers can be obtained at all future times from the following formula.

$$X\left(t\right) = \sum_{j=1}^{n} \phi_j \exp\left(\omega_j t\right) b_j = \Phi \exp\left(\Omega t\right) b, \tag{11}$$

where b_j are related to the linear combination coefficients of the starting value of the data from the DMD modes, Φ is a matrix whose columns are the DMD eigenvectors ϕ_j, and $\Omega = diag(\omega)$ is a diagonal matrix whose entries are the eigenvalues ω_j. The state X and the eigenvectors ϕ_j have the same size, and the vector b consists of the coefficients b_j.

2.2 Long short-term memory (LSTM)

LSTM is a particular type of recurrent neural network that can learn long-term dependencies. In many problems, information must be held in memory while running. This advantage of LSTM is beneficial in solving these problems [4].

In a simple recurrent neural network, a run of values used as input $x^{(1)}, \ldots, x^{(n)}$ are processed sequentially to generate a corresponding sequence of hidden states $h^{(1)}, \ldots, h^{(n)}$, output values $o^{(1)}, \ldots, o^{(n)}$, and model predictions $y^{(1)}, \ldots, y^{(n)}$. At each time step $1, \ldots, n$, the beginning state $h^{(0)}$ of the RNN is applied to a series of equations to update the hidden layer, output layer, and predictions [6]:

$$\begin{cases} h^{(t)} = \tanh(b + W h^{(t-1)} + U x^{(t)}), \\ o^{(t)} = c + V h^{(t)}, \\ y^{(t)} = g(o^{(t)}), \end{cases} \tag{12}$$

where, in turn, the letters U, V, and W stand for the weight matrices for input-to-hidden, hidden-to-output, and hidden-to-hidden connections, respectively. Furthermore, the bias vectors b and c are present, as well as the softmax function $g(z)$, which is specified as

$$g(z_i) = \frac{e_i^z}{\sum_i e_j^z}. \tag{13}$$

RNNs have trouble with longer sequences because early in the sequence back-propagated error signals tend the propensity to vanish or inflate [7]. By including the memory cell in the RNN design, this is known as vanishing gradient, which improves on short-term memory networks. These cells enable the network to store information and delete it after usage. These cells are linked periodically and replace the recurrent networks hidden units. The input, forget, and output gates are the three gates that each cell possesses to regulate how information enters and leaves the cell.

Typically, the following formulae are used to define these cells:

$$f_i^{(t)} = \sigma\Big(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\Big), \tag{14}$$

$$g_i^{(t)} = \sigma\Big(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\Big), \tag{15}$$

$$q_i^{(t)} = \sigma\Big(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\Big). \tag{16}$$

Each of these gates is composed of a sigmoid activation function that outputs a value between zero and one, representing the amount of information to let through the gate. The bias vectors b, weight matrices U, V, and W all contain superscripts that indicate which gate they belong to, such as b^o for an output gate and U^f for a forget gate. The (z) sigmoid activation function has a definition

$$\sigma(z) = \frac{1}{1+e^{-z}}. \tag{17}$$

For memory cell i at time step t, the LSTM cells internal state $s_i^{(t)}$ is updated by

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\Big(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\Big). \tag{18}$$

The LSTM cell calculates the output $h_i^{(t)}$ using:

$$h^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)}. \tag{19}$$

A standard artificial neuron unit is used to compute an input characteristic. The value of the input can be added to the state if the sigmoid input gate allows it. The linear inner self-loop of the memory cells is managed by a forget gate, and the output gate can completely disable the output of the cells.

The overall design of LSTM is comparable to a straightforward recurrent neural network, but LSTM memory cells are used instead of hidden layers. With the help of this structure, LSTM networks may find long-term correlations in time series data [5].

3 Hybrid method of DMD and LSTM

It is evident that financial markets lack a convolution among their signal samples, which can impede the accuracy of LSTM models in predicting market trends. To address this issue, a combined DMD-LSTM method has been proposed as a more practical approach for forecasting the stock prices.

Initially, financial market data is collected, and then the DMD algorithm is implemented to extract coherent and dominant structures from the data [11]. However, a rank mismatch issue may arise with the DMD response since the number of days (columns) is greater than the number of data types (rows) in the data matrix [16]. To address this issue, the matrix can be augmented using delay coordinates [8]. These coordinates refer to an augmented vector that contains states at the current time along with copies of states at future times. This augmentation can be done using past or future measurements and changing the order will not affect the results of the DMD method. The following is the process of constructing the augmented data matrix

$$X = \begin{bmatrix} x_1 & x_2 \dots & x_{m-n} \\ \\ x_2 & x_3 \dots & x_{m-n+1} \\ \vdots & \vdots & \vdots \\ x_n & x_{n+1} & x_{m-1} \end{bmatrix}, \tag{20}$$

$$X' = \begin{bmatrix} x_2 & x_3 \dots & x_{m-n+1} \\ \\ x_3 & x_4 \dots & x_{m-n+2} \\ \vdots & \vdots & \vdots \\ x_{n+1} & x_{n+2} & x_m \end{bmatrix}, \tag{21}$$

where $x_i = x(t_i)$.

After completion the construction of the augmented data matrix, we proceed to execute the DMD algorithm, as outlined in Section 2.1. The result of this algorithm led to the estimation of the system response, which includes the coherent and dominant structures observed in the data. We extract this data and use them in LSTM method to predict future states.

To build an LSTM neural network and train and then predict the data, we first define the target signal:

$$\begin{aligned} x &\longrightarrow [1 : end - 1]\,, \\ y &\longrightarrow [2 : end]\,, \end{aligned}$$

where x is the input and y is the desired output. In this way, the desired output will be one step (one day) ahead of the network input. Figure 1 describes this better.

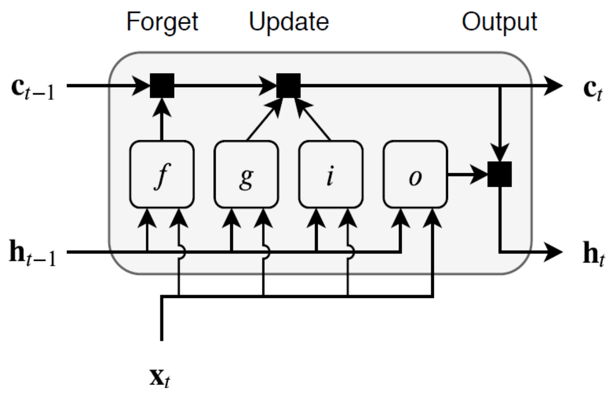

Figure 1: LSTM network (the figure is taken from MathWorks)

Recurrent neural networks have weaknesses in their training procedure due to their recursive nature, which may lead to divergence. Thus, to ensure an excellent statistical distribution of the data, normalization of the data is employed utilizing the formula below

$$X_{Normal} = \frac{X - \bar{X}}{S},$$

where $\bar{X}$ is the mean and S is the standard deviation. Then at the end of the forecast, we return the data back to its true range and mean.

The DMD-LSTM algorithm

(i) We apply the DMD algorithm 2.1 to the available data.

(ii) We construct LSTM network structures as follows:

- Input layer: Which is a three-dimensional input that includes samples, time steps, and features. Here we choose the number of features equal to 1.
- LSTM layer: Which is an RNN layer that learns long-term dependencies between time steps in time series and sequence data.
- Output layer: which is a regression layer that maps the outputs of the intermediate layers using the end layers connected to the desired output layer. We consider the number of neurons in the output layer (the number of regression layer) to be 1, because the desired output is a variable.

(iii) We set the network parameters.

(iv) We train the network. We consider 90% of the data for this purpose.

(v) We test the network. We consider 10% of the data for predication.

After training the network, 10% of the data is predicted and compared to the original data. A target signal that enables the network to anticipate the following input based on the past input must be provided to train the recurrent neural network. The training signals are then defined and utilized to construct an LSTM recurrent neural network, which comprises a sequence input layer followed by LSTM layers. The number of LSTM layers affects the performance of the network and its execution time. Since the current problem is a regression task, the output of the LSTM layers, which is a combination of a fully connected layer and a regression layer, are mapped to a variable that acts as the final layer of the network [12]. The LSTM network used here consists of one input and one output as well as 200 hidden layers.

Error scales

We compare the two methods using a robust benchmark called root mean square error (RMSE) that is as follows

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{pred} - y_i^{actual}\right)^2}.$$

Also, we use 95% confidence interval(CI). This scale is aften used in statistical sampels. For large n, based on a random sample from a distribution with mean μ and standard deviation σ, an approximate 95% confidence interval for μ is given by

$$\bar{x} \pm 1.96\frac{s}{\sqrt{n}},$$

where $\bar{x}$ and s are the observed value of the sample mean and standard deviation respectively.

The performance of the combined DMD-LSTM approach is evaluated using three cases in the next section, and the percentage of investment profit and loss in a certain period of time is calculated using the DMD-LSTM method. This percentage is then compared with the profit and loss percentage of the LSTM method and outstanding market data.

4 Examples

In this section, we want to study three practical examples of financial market forecasting with three different behaviors: bullish, bearish, and neutral. We obtained these results using the MATLAB R2019b software on a PC with an Intel Core i5 CPU and 4 GB of RAM.

Example 4.1. In this example, we use the data of the closing price of US Dollar (USD) to Iranian Rial (IRR) in 1602 days from February 2017 to January 2023.

The graph of these data is in Figure 2. What can be seen in the chart shows the upward trend of the price of the USD to IRR in the entire period.

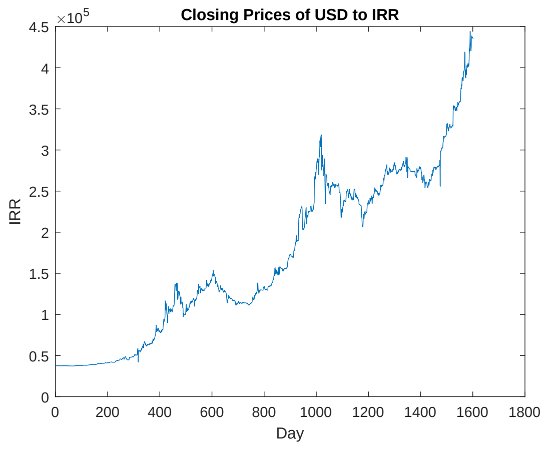

Figure 2: 1602 days daily closing prices USD to IRR for example 4.1.

For prediction, we first feed the real market data to the LSTM method. To build the neural network and train it, we consider the closing prices of USD to IRR during 1442 days for training and keep the 160 days for testing. As we can see in Figure 3, the prediction of last 160 days prices by LSTM method, compared to the real market data, does not apply even in terms of whether it is bullish or bearish. According to the LSTM prediction chart, we do not have a relatively good trend. On days when the signal increases, on some days our forecast decreases and on some days when the signal decreases, our forecast increases.
Now, we extract the dominant and consistent daily Dollar closing price data with the DMD algorithm and then feed the pre-processed data to the LSTM network.

To do this, first we put all the data, including the closing prices of the USD to IRR, in the augmented data matrix X as equation (20), then we put this data in the X' matrix by moving one day forward. Then we find the SVD decomposition of X' matrix and calculate the $\tilde{A}$ matrix according to equation (8). The next step is to find the eigenvalues and eigenvectors of $\tilde{A}$, which we extract from these values and equation (10) eigenmodes of the matrix. Finally, using equation (11), we calculate the DMD prediction data and use this data in the LSTM algorithm.

Similarly to the previous case, we train 1442 days of data and keep 160 days for testing. Figure 4 shows that the predicted data for the last 160 days is similar to the actual data itself in terms of upside and downside. Our data indicate a positive correlation between the signal and forecast days: when the signal increases, forecast days increase, and when the signal decreases, forecast days decrease. Additionally, based on obtained results in Table 1, RMSE ratio of LSTM method to DMD-LSTM method is almost equal to 14. This shows that DMD-LSTM method can reduce

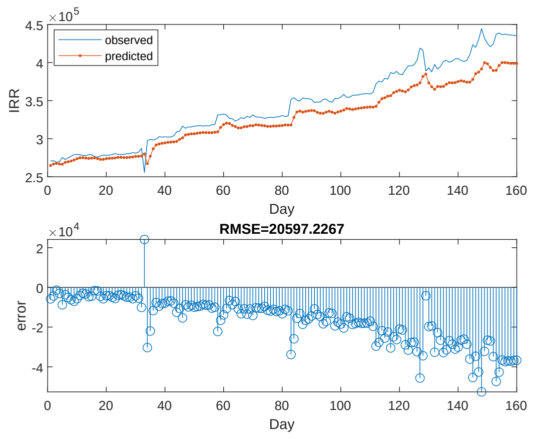

Figure 3: Forecasting of 160 days of daily closing prices USD to IRR using LSTM method for Example 4.1.

RMSE by 7.1% and the confidence interval for the error of DMD-LSTM method is superior to that of LSTM. Additionally, during the execution of code in MATLAB software, the DMD method takes less time than the LSTM method. The utilization of low-dimensional modes in the DMD method is what leads to the reduction in time for this phenomenon.

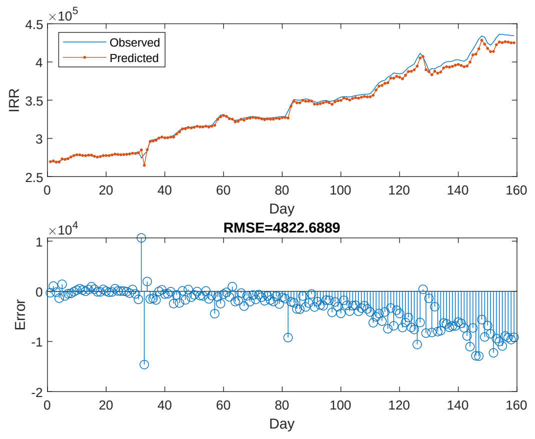

Figure 4: Forecasting of 160 days of daily closing prices of USD to IRR using DMD-LSTM method for Example 4.1.

Example 4.2. In this example, we extract the data about the proportion of Siman Shargh stock in the Tehran Stock Exchange, specifically consisting of the closing

Table 1: Error scales and running time for Example 4.1

Method	RMSE	Confidence Interval	Running Time (Second)
LSTM	20597.2267	$[0.106171, 36857.53]$	304.493364
DMD-LSTM	4822.6889	$[0.014079, 4862.918]$	248.675378

prices of shargh stock over 1400 days from January 2017 to February 2023. The corresponding data is presented in Figure 5. The chart of this stock is neutral except for a short period when it rises and falls.

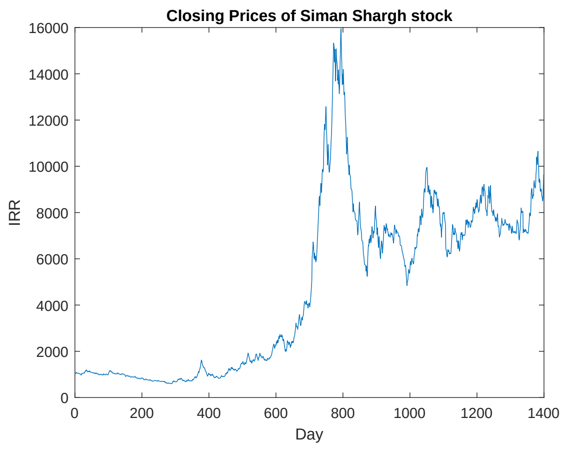

Figure 5: 1400 days of closing prices of Siman Sharq stock for Example 4.2

The original data are integrated into the LSTM network, which is then subjected to a training phase consisting of 1260 days of stock market data, with the remaining 140 days reserved for testing purposes. The findings presented in Figure 6 show that the forecasts made by LSTM method for the 140-day future data are characterized by different patterns of bullish and bearish trends compared to the actual stock market data. In addition, the RMSE error shown in this graph is relatively large.

The next method consists of extracting the dominant and consistent daily data of the closing price of Siman Shargh stock using the DMD algorithm and then transferring this pre-processed data to the LSTM network. The training phase includes 1460 days of stock market data and 140 days for testing, according to the formula. The findings presented in Figure 7 show that the predicted performance of the next 140 days shows a similar pattern of up and down trends to the original data. According to the results in Table 2, the RMSE and the confidence interval highlights a substantially lower error rate when compared to the LSTM method. It is worth noting that this trend remains constant and balanced, and the increase in

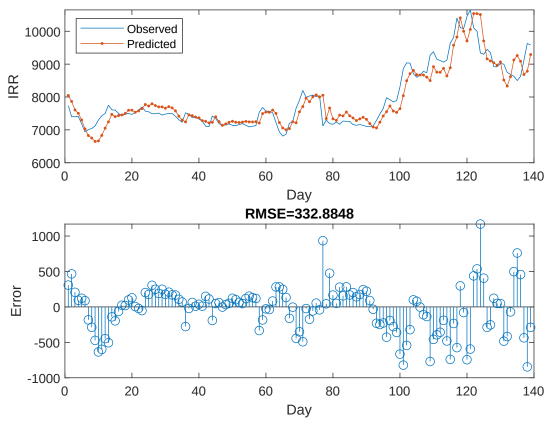

Figure 6: Forecasting 140 days of daily closing prices of Siman Shargh stock using LSTM method for Example 4.2.

stock price is accompanied by a corresponding increase in the forecast trend, and vice versa for a decrease in stock price.

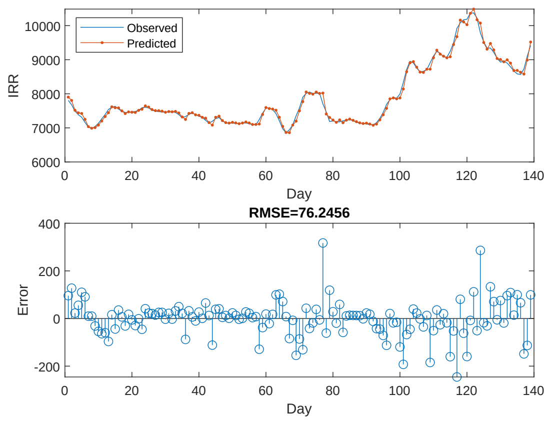

Figure 7: Forecasting 140 days of daily closing prices of Siman Shargh stock using DMD-LSTM method for Example 4.2.

Example 4.3. Similar to the previous two examples, this time we collect the stock data of zob roy Isfahan (Faroy) in Tehran Stock Exchange for 701 days from January 2020 to February 2023. The corresponding trend of the time signal is shown in Figure 8. As can be seen in the figure, the trend of the stock can be considered downward.

The LSTM network is first integrated with the original data and subsequently

Table 2: Error scales and running time for Example 4.2

Method	RMSE	Confidence Interval	Running Time(Second)
LSTM	332.8848	$[0.074844, 597.4653]$	310.018435
DMD-LSTM	76.2456	$[0.028251, 224.5398]$	246.175938

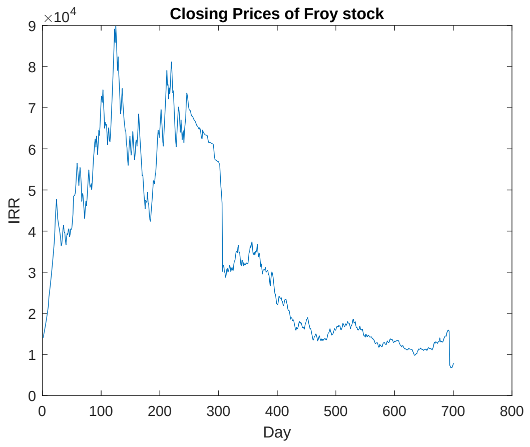

Figure 8: 701 days of daily closing prices of Froy stock for Example 4.3.

undergoes a training phase that includes 631 days of data, while the remaining 70 days are reserved for testing purposes. Figure 9 shows that the results obtained from the LSTM method have a high error. In order to improve the error, we can use the combined method of DMD and LSTM, the results of using this method are clear in Figure 9 and it shows the high accuracy of the method and the improvement of the error to the optimum level.
To implement the new method, we first use the DMD algorithm to extract dominant data and then enter them into the LSTM network. Figure 10 shows that the 70-day stock price forecasts produced using this method are very close to stock market prices. It's worth noting that, based on Table 4, the RMSE and confidence interval associated with this method are less than LSTM method, and during the execution of code in MATLAB software, the DMD method takes less time than the LSTM method. This is because in the DMD method, dominant data is extracted from the original data and inputted into LSTM, resulting in time savings.

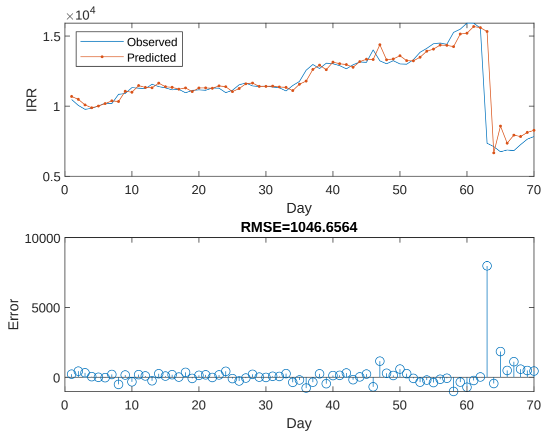

Figure 9: Forecasting 70 days of daily closing prices of Froy stock using LSTM method for Example 4.3.

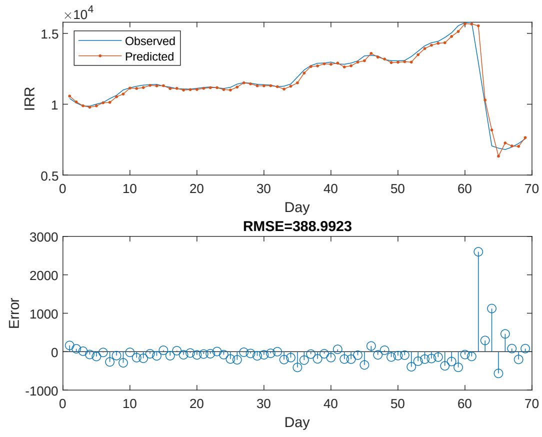

Figure 10: Forecasting 70 days of daily closing prices of Froy stock using DMD-LSTM method for Example 4.3.

Table 3: Error scales and runnig time for Example 4.3

Method	RMSE	Confidence Interval	Running Time (Second)
LSTM	1046.6564	$[0.214187, 2537.271]$	129.350504
DMD-LSTM	388.9923	$[0.091057, 1080.169]$	114.485250

5 Conclusion

In order to predict the financial markets, in this study, a new DMD-LSTM combined technique was proposed, and the performance of the proposed method was investigated in three different examples that had upward, neutral, and downward trends respectively. The results were promising and have better predictions than the existing LSTM method. In the future and to improve the method, the extended dynamic mode decomposition (EDMD) method can be used. In addition, the proposed approach can be used to forecast various time series in applied sciences.

Acknowledgement

Acknowledgements could be placed at the end of the text before the references.

Bibliography

[1] S.L. Brunton, J.N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*, Cambridge University Press, 2022.

[2] F.A. Gers, J. Schmidhuber, F. Cummins, *Learning to forget: Continual prediction with LSTM*, Neural Comput. 12 (2000), PP. 2451–2471.

[3] G. Golub, W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis 2(2) (1965), PP. 205–224.

[4] I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, MIT press, 2016.

[5] H.S. DiStefano, *Predicting long-term US housing price trends using a long short-term memory neural network*, Ph.D. Thesis, University of California, Los Angeles, 2022.

[6] A. Graves, *Long short-term memory*, Supervised sequence labelling with recurrent neural networks, Stud. Comput. Intell. 385 (2012), PP. 37–45.

[7] S. Hochreiter, J. Schmidhuber, *Long short-term memory*, Neural Comput. 9(8) (1997), PP. 1735–1780.

[8] J.N. Juang, R.S. Pappa, *An eigensystem realization algorithm for modal parameter identification and model reduction*, J. Guid. Control Dyn. 8(5) (1985), PP. 620–627.

[9] B.O. Koopman, *Hamiltonian systems and transformation in Hilbert space*, Proc. Natl. Acad. Sci. USA 17 (1931), PP. 315–318.

[10] J.N. Kutz, S.L. Brunton, B.W. Brunton, J.L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*, SIAM, 2016.

[11] J. Mann, J.N. Kutz, *Dynamic mode decomposition for financial trading strategies*, Quant. Finance 16 (2016), PP. 1643–1655.

[12] M. Paluszek, S. Thomas, *Practical matlab deep learning*, A Project-Based Approach, Apress Berkeley, CA, 2020.

[13] P.J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, J. Fluid Mech. 656 (2010), PP. 5–28.

[14] P.J. Schmid, J. Sesterhenn, *Dynamic mode decomposition of numerical and experimental data*, In Bull. Amer. Phys. Soc., 61st APS meeting, p. 208. San Antonio, 2008.

[15] S. Siami Namini, A. Siami Namini, *Forecasting economics and financial time series: ARIMA vs. LSTM*, arXiv:1803.06386 (2018).

[16] J.H. Tu, *Dynamic mode decomposition: Theory and applications*, Ph.D. Thesis, Princeton University, 2013.

[17] J.H. Tu, C.W. Rowley, D.M. Luchtenburg, S. L. Brunton, J.N. Kutz, *On dynamic mode decomposition: Theory and applications*, Journal of Computational Dynamics 1 (2014), PP. 391–421.

[18] S. Yao, L. Luo, H. Peng, *High-frequency stock trend forecast using LSTM model*, 13th International Conference on Computer Science & Education (ICCSE), IEEE, (2018), PP. 1–4.

How to Cite: Roya Karimkhani[1], Yousef Edrisi Tabriz[2], Ghasem Ahmadi[3], *A new hybrid method of dynamic mode decomposition and long short-term memory for financial market forecasting*, Journal of Mathematics and Modeling in Finance (JMMF), Vol. 3, No. 2, Pages:1–17, (2023).