

Prediction of Iran Stock Market Using LSTM Network and DTW Algorithm

Abbas Zare, Zahra Rezaei

Department of Computer Engineering, Marvdasht Branch, Islamic Azad University, Marvdasht, Iran

Abstract—The fluctuations, noise, and information load of the stock market necessitate efficient forecasting methods. Due to the nonlinear and non-stationary nature of time-series data generated from the stock market, predicting index prices is a complex task. In this dynamic market, intelligent forecasters provide analytical tools and predictive models that enable investors and traders to make informed decisions and reduce financial risks. Stock market data is considered time-series data as it is generated regularly. Long Short-Term Memory (LSTM) networks are particularly effective for time-series forecasting. In this study, the trend of the Tehran Stock Exchange index and the *Shapna* stock has been predicted using an LSTM network. For data classification, a comparison is made between the price of each day and the previous day. If the price increases or remains relatively stable compared to the previous day, it is assigned to class (1); if the price decreases, it is assigned to class (-1). The DTW algorithm is used to compare the predicted results with actual values. By employing two-class classification and tuning the parameters of the LSTM network, model accuracy improved. Additionally, removing parts of the price chart affected by market excitement—considered as outliers—played a key role in enhancing the prediction accuracy of the model.

Index Terms— Stock market, LSTM network, prediction, classification, recurrent neural network parameters, activation function

I. INTRODUCTION

Financial markets, particularly stock markets, have always attracted the attention of both retail and institutional investors. One of the key challenges faced by investors is predicting future stock prices. In financial terms, the method of analyzing historical market data—primarily price and volume—to forecast price movements is referred to as technical analysis. Technical analysts argue that supply and demand ultimately determine price discovery. By examining and comparing past price charts and identifying different recurring patterns, they predict the future price of an asset. The stock market is highly volatile, and due to its uncertainty, it is impossible to predict stock prices with absolute accuracy. However, stock investors and traders can utilize such models to make informed decisions about buying, holding, or investing

in stocks. Financial institutions can also leverage these models to manage risk and optimize their clients' investment portfolios.

The fluctuations, noise, and information load of the stock market require efficient prediction methods. The nonlinear and non-stationary nature of time-series data generated by the stock market makes predicting index prices highly complex. In recent years, financial activities have grown significantly, and with rapid economic development, their changes have become increasingly intricate. Understanding the patterns of financial activities and predicting their progression and changes are key research priorities in both academic and financial circles [1]. Forecasting financial data can aid in understanding the development and transformation of the financial market on a macro level and serve as a foundation for making investment decisions and maximizing profits for organizations. However, because economic data is often complex, incomplete, and ambiguous, predicting its trends is exceedingly challenging [1]. The performance of an intelligent stock recommendation system assists in making decisions for buying or selling stocks and generating higher profits in stock trading. However, due to the nonlinear nature of stock prices, developing such systems is highly challenging. Moreover, the knowledge of recommendation systems regarding the dynamics of the stock market is limited.

Traditional recommendation systems are primarily based on technical analysis. However, recent studies in this field suggest that recommendation systems based on soft computing/data mining approaches can provide profitable trading recommendations [2]. These systems can identify stock price activity patterns and significantly enhance the decision-making process for stock traders. Such systems are particularly valuable for individuals who lack the expertise or skills of experienced traders [3]. Becoming a successful trader in the stock market requires substantial experience in stock trading and the ability to identify stock price movement trends. These systems can act as experts, making them highly valuable for non-professionals aiming to profit from stock market investments [3]. The Efficient Market

Hypothesis (EMH) was introduced by Fama (1965). According to this theory, historical stock market data contains practical information that must be taken into account [4]. There are two primary methods for stock prediction: fundamental analysis and technical analysis.

Fundamental analysis utilizes factors related to a company's well-being, such as its revenues and expenses, market position, and annual growth rate. While technical analysts believe that all information necessary for predicting price changes is embedded in historical prices, fundamental analysts prefer to study the underlying companies to determine their intrinsic value. This allows them to predict whether a stock's price will increase or decrease in the future [5]. Ratios are often employed to compare the value of different companies within the same sector. The key financial ratios used in fundamental analysis include:

- Profitability Ratios: Measure the company's earning power.
- Liquidity Ratios: Assess the company's ability to pay off its immediate obligations.
- Debt Ratios: Evaluate the firm's capacity to meet long-term debt obligations.
- Asset Utilization Ratios: Determine how effectively a company uses its assets.
- Market Value Ratios: Reflect the market's valuation of the company.

These ratios provide a comprehensive perspective on a company's financial health and performance, helping investors make informed decisions.

Technical analysis methods aim to identify and utilize significant patterns in stock price movements. These approaches are typically used for short-term predictions. On the other hand, fundamental analysis approaches focus on critical factors related to the companies whose stocks are to be predicted. The main idea is to estimate the intrinsic value of a company and compare it with its current stock price. For instance, if an investor believes that a stock is currently undervalued, they might invest in it, expecting the stock price to rise in the future. Fundamental analysis approaches are generally used for long-term predictions [5]. This technic aims to predict future values using historical stock prices. The primary assumption motivating technical analysis is the existence of patterns that can reliably be used for forecasting future values [5]. Technical analysis involves making stock trading decisions based on historical stock market data. The premise of its application in the investment industry is that using past data can yield above-average returns. A meta-analysis conducted by Park and Irwin (2004) indicates that most studies on technical analysis demonstrate

profitability. However, such analyses should be interpreted with caution [4]. Clark et al. (2001) and Grig and Menkhoff (2006) found that technical analysis remains widely used in today's investment industry, even with the persistence of the Efficient Market Hypothesis (EMH). However, in most cases, this term refers to simpler approaches within technical analysis. Numerous established techniques exist in the field of technical analysis [5]. The main idea is to analyze data to identify trends and capitalize on them. There are three fundamental principles:

- **Market Action Discounts Everything**
Murphy asserts that everything that could potentially affect prices—fundamentally, politically, psychologically, etc.—is already reflected in the market price. This assumption implies that all fundamental information influencing prices is embedded in historical prices.

- **Prices Move in Trends**

The core idea here is that if a trend is currently in motion, it is more likely to continue than to reverse. This assumption is critical because generating profits through short-term or long-term trades would not be feasible without it.

- **History Repeats Itself**

Murphy suggests that technical analysis heavily relies on human psychology and assumes that human behavior remains consistent over time. This assumption enables analysts to use similar patterns to identify bullish and bearish markets.

This article investigates the performance prediction of the *Shapna* stock and the Tehran Stock Exchange index. Stock price prediction has been performed with relatively satisfactory accuracy using step-by-step modeling and LSTM. For the prediction section, two models have been considered. The first model is based on data labeling, which is calculated according to the previous day's price. In the second model, the LSTM network is trained based on the close price to predict the stock price trend for the coming days. The subsequent sections of this article include the literature review, the proposed method, simulation results, and the conclusion. The remainder of this paper discusses the literature review, the proposed method, simulation results, and the conclusion.

2. LITERATURE REVIEW

In the work of Yu and Yan [1], forecasting for 1, 2, 5, 10, 15, 20, and 30 days was explored using various machine learning techniques to predict future values of stock market groups. Methods such as decision trees, bagging, random forests, adaptive boosting (AdaBoost), gradient boosting, XGBoost, artificial neural networks (ANN), recurrent neural networks (RNN), and extended short-term memory networks

(LSTM) were applied. Ten technical indicators were selected as inputs for each algorithm. Among the compared algorithms, LSTM demonstrated superior accuracy. The report by De Rossi et al. [6] aimed to equip portfolio managers with a tool to narrow down their extensive stock lists by conducting in-depth stock analyses. The proposed stock recommendation and prediction model were developed based on observed features and the past behavior of investors. An empirical study used a large set of global actively managed funds between 2005 and 2016. The results showed that the proposed system could effectively predict future buy transactions.

Patel et al. [7] addressed the challenge of forecasting price direction changes for 23 stocks in Indian markets. Four predictive models were used: artificial neural networks (ANN), support vector machines (SVM), random forests, and Naive Bayes. The findings showed improved predictive performance for all proposed models. In a study by Nir et al. [3], an optimized genetic algorithm was proposed to develop a stock trading recommendation system that extracts temporal association rules from stock price data. The system was tested on 12 datasets and demonstrated significantly better performance than passive buy-and-hold strategies, showing potential for successful investment in capital markets. Vismaya et al. [2] developed stock trading recommendation systems based on a novel classification method, using historical stock price data and technical indicators as input features. An empirical evaluation was provided for India, the sixth-largest economy in the world, and the Bombay Stock Exchange (BSE). The performance of the recommendation system for each stock, based on classification accuracy, indicated that the proposed method could successfully generate profitable trading recommendations. Extensive research has been conducted on stock performance prediction. The results indicate that data from textual sources related to the stock market can be successfully used for predictions. Most existing approaches focus on short-term predictions, employing relatively simple sentiment analysis techniques or limited available data. Boone [5] utilized over a decade of stock data and proposed a solution that integrates annual textual features and quarterly archives with fundamental factors to predict long-term stock performance. Additionally, a text feature extraction method was developed. Results showed that feature selection significantly enhanced test performance and reliability compared to baseline models. Text-based prediction approaches using machine learning models have also been applied, leveraging news articles to provide new insights rather than relying solely on historical data for

predicting stock prices [4]. Deep learning has gained attention in recent years due to its advanced computational capabilities and layered models. Financial data, characterized by complexity, incompleteness, and ambiguity, presents significant challenges for trend forecasting. Economic data volatility is influenced by thousands of continuously changing factors, making financial data analysis a nonlinear, time-dependent problem. Deep neural networks (DNN) combine the advantages of deep learning (DL) and neural networks to address these challenges.

Jiajing Wang et al. [16] emphasized that fine-tuning hyperparameters and strategically configuring neurons in hidden layers significantly improve prediction accuracy, aligning forecasted stock trends with actual data. Deep-layered models for time series analysis have broad applications and are recognized as a challenging research area. Deep neural networks can learn complex time-series correlations, making them suitable for predicting price trend changes based on the slope of other stocks' trends. The study by Moves [4] used the S&P 500 stock index as a test dataset. Models developed based on previous stock trend slopes were tested, demonstrating their ability to forecast trend changes effectively. A recent study by Song Jintong et al. [17] used a multi-modal self-attention method to improve the prediction accuracy of deep learning models in stock prediction. This research introduced an innovative gold price forecasting model based on a Bayesian-optimized LSTM network, achieving improved prediction accuracy by optimizing the model's hyperparameters. Khanesha et al. [8] proposed a profitable portfolio allocation strategy using reinforcement learning. They developed a novel risk index based on innovative money flow behavior to determine optimal buying and selling times. The results showed that the model outperformed all baseline strategies, including traditional buy-and-hold approaches.

Kalra et al. [9] introduced a novel hybrid Bi-directional LSTM model. This model combined incremental learning and deep learning techniques to forecast real-time index prices. Its implementation in a live trading system demonstrated the method's effectiveness in predicting major global stock indices. Varadarajan et al. [10] employed a combination of Long Short-Term Memory networks and Recurrent Neural Networks (LSTM-RNN) to predict Amazon's stock closing prices. They studied the effects of various hyperparameters in their proposed model to identify factors influencing predictive performance. However, due to market uncertainties, further research is recommended to reassess the model's accuracy. Safari and Badamchi Zadeh [11] designed a multi-

modal deep learning model for predicting stock prices and performing intelligent trade analysis. Their proposed model incorporated bi-directional LSTM networks for analyzing Amazon's features and was compared with other models, including KNN, LSTM, RNN, CNN, and ANN.

III. PROPOSED METHODOLOGY

The data collected from the stock exchange website is first divided into training and testing datasets. The model is trained using the training data and then tested with the testing data to make predictions under various parameters. The predicted results are compared with the actual results, and a confusion matrix is generated to evaluate the model's performance.

A. Data Collection and Preprocessing

The data is obtained in CSV format using the *TseClient* software provided by the stock exchange organization. The data can include stock market indices (e.g., stock index, OTC index, equal-weighted index, and sector indices such as automotive, refinery, banking, etc.) or individual stock symbols (e.g., *Shapna*, *Khodro*, *Palayesh*, *Ghamino*, etc.). During data acquisition, configurations such as the output file type, start and end dates of the transactions, and other settings can be applied. In the preprocessing stage, the collected data is prepared for subsequent steps. Initially, unnecessary features are removed, and records lacking essential information such as opening, closing, high, or low prices—caused by reasons such as stock suspension, holding a general meeting, or other factors—are excluded. Next, the date format is adjusted to a compatible format, and if necessary, the data for the target symbol is adjusted (e.g., to account for events like capital increases). Adjusting stock prices ensures that the resulting charts are plotted more smoothly and continuously. Table 1 presents the preprocessed data for the *Shapna* stock after completing these preprocessing steps.

TABLE I
PREPROCESSED DATA FOR THE *SHAPNA* STOCK

index	Closing Price
0	1263.06
1	1257.12
2	1302.84
3	1310.09
4	1338.66
...	...
697	7090.0
698	6870.0
699	6950.0
700	7060.0
701	6970.0

B. Preparation of Training and Testing Data

In this test, the training data accounts for approximately 80% of the total data, while the test data comprises 20% of the total data. The training and test data are placed in separate arrays. The training parameters are set as follows: the loss function is of the Mean Squared Error type, and the optimizer parameter is set to Adam. The loss function (or cost function) essentially displays the error in each iteration of the neural network for the training data. In other words, in machine learning, the loss function is used to determine the error or deviation during the learning process [12]. Optimization is a critical process in machine learning, which adjusts input weights by comparing predictions with the loss function [13].

C. Data Classification

Since stock market data is time-series and lacks labels, it is necessary to label (classify) the data first [14]. For the first day's data (representing the stock price or index), the class is assigned as 0. If the price increases on the next day (or record), it is assigned to class 1, and if the price decreases, it is assigned to class -1. If the price change is negligible, the class for that record is assigned as 0. The threshold for determining negligible changes is defined by the *FixedPercent* parameter, which is set by default to 0.01. This means that changes within 1% are considered as no change, or the price is regarded as constant. Figure 1 shows the classification of a portion of the *Shapna* stock data.

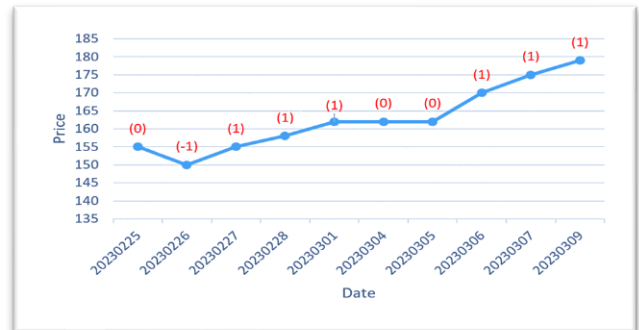


Fig.1. The classification of a portion of the *Shapna* stock data

Machine learning and deep learning are powerful tools for extracting meaningful insights from large and complex stock market data generated at regular intervals [9]. Among these methods, Long Short-Term Memory (LSTM) networks are particularly effective for time-series forecasting. For stock price prediction, LSTM networks often outperform models like Autoregressive Integrated Moving Average (ARIMA), as they can capture complex and nonlinear patterns in time-series data. However, real-time updates of these models are crucial for making

accurate predictions [9]. The LSTM algorithm addresses one of the significant challenges in deep learning architectures: the problem of vanishing gradients. This issue arises when gradient values become increasingly small as they propagate backward through the network, causing minimal updates to weights. This slows down the training process significantly and halts it entirely in extreme cases. LSTM networks, a specialized type of recurrent neural network (RNN), are designed to learn long-term dependencies over time [15]. By introducing memory gates into the deep learning architecture, the LSTM algorithm effectively resolves the vanishing gradient problem. The figure below illustrates the functionality of memory blocks within the LSTM algorithm [16, 17].

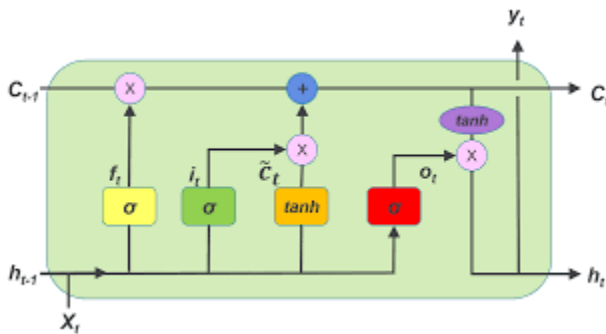


Fig.2. LSTM structure with the memory block

D. Evaluation Metrics

The confusion matrix serves as the basis for measuring the quality of a classification algorithm. This matrix includes True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Metrics such as Accuracy, Precision, Recall, and F1-Score are calculated based on the confusion matrix.

1) Accuracy

The accuracy metric is defined as the number of correct predictions (True Positive) divided by the total number of predictions made. Essentially, it measures how many of the model's predictions were correct out of all the predictions it made.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2) Precision

Precision refers to the proportion of correct predictions made by the model for a specific class out of all predictions it made for that class.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

3) Recall

Recall refers to the proportion of instances from a specific class in the dataset that the model correctly identified.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

4) F1-Score

When you want the evaluation metric to represent a balance between Recall and Precision, you can use the harmonic mean of these two metrics, which is referred to as the F1-Score.

$$\text{F1_score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

5) DTW

In time-series analysis, Dynamic Time Warping (DTW) is an algorithm that measures the similarity between two-time sequences [18]. The time complexity of the DTW method is $O(MN)O(MN)O(MN)$, where MM represents the number of predicted data points (days) and NN represents the number of price fluctuations. For example, if the model makes predictions for 20 days and the index (stock) price fluctuates 15 times during this period, DTW will operate with a time complexity of 600 ($20 \times 1520 \times 1520 \times 15$). To measure similarity in stock market data, most algorithms use methods that calculate the distance between two sequences using Euclidean measures. However, in specific scenarios, this approach may lead to errors. For instance, two identical sequences that differ only in timing may result in a significantly considerable distance when calculated using Euclidean methods, causing mistakes in similarity measurement, even though the sequences are very similar. In Figure 3, the difference in similarity measurement between the Euclidean distance method and the Dynamic Time Warping method is visually apparent.

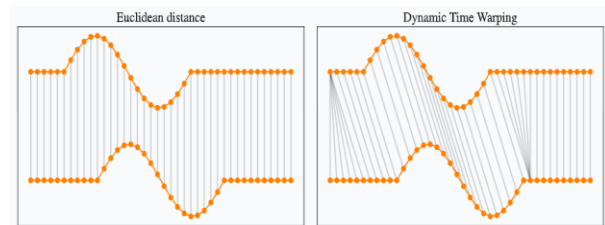


Fig.3. Comparison of Euclidean distance and dynamic time warping

E. Evaluation of the Proposed Model

The proposed method assumes that stock price changes each day can only fall into three categories: the price increases, decreases, or remains unchanged compared to the previous day. Therefore, if the stock price increases on a given day and the algorithm correctly predicts the growth (or conversely, the price decreases and the algorithm predict a decrease), this is considered a True Positive according to the concepts

of the confusion matrix, which impacts the model's accuracy.

Initially, the price change for each day compared to the previous day is analyzed. If the price increases, it is assigned to the price increase class (1); if it decreases, it is assigned to the price decrease class (-1). A tolerance for price stability can also be defined using a parameter called FixedPercent. For example, if the tolerance for price changes is set to ± 0.005 , this means that if the stock price on the previous day was 1,000 rials, any price change within the range of 995 to 1,005 rials is considered stable, and the stable class (0) is assigned.

After classifying the data, plotting the confusion matrix, and calculating the model's accuracy, we observed that the model's accuracy was very low, dropping to as low as 30% in some cases. Therefore, instead of using three classes, we switched to two classes (1 and -1), where a price increase was assigned to class 1 and both a price decrease and no price change were assigned to class -1. By adopting this approach, we observed a significant improvement in the model's accuracy. Although at first glance, a three-class classification might seem more precise than a two-class classification—since, on some days, the price changes are minimal, making it difficult to determine which class the data belongs to—the results indicate that a two-class classification ultimately achieves higher accuracy than a three-class classification. With these changes, the model's accuracy increased to a maximum of 55% (Figures 4 and 5).

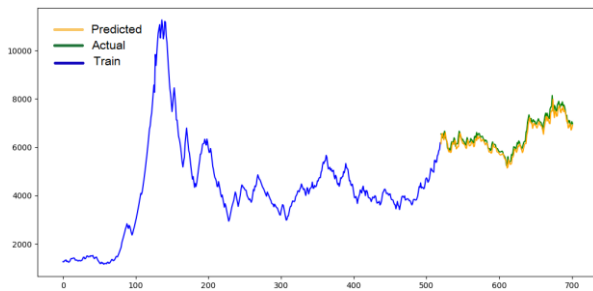


Fig.4. *Shapna* stock forecast chart

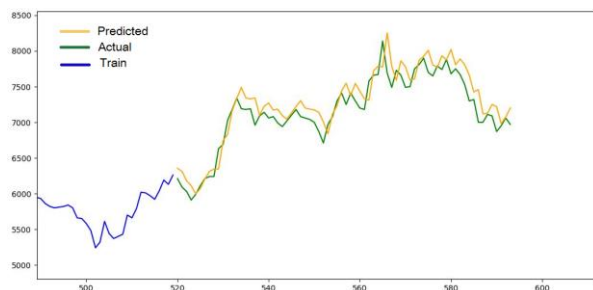


Fig.5. Magnified *Shapna* stock prediction chart

By comparing the actual labels with the predicted labels, we observed that almost all the actual and predicted classes—except for a few—matched with a time lag of one unit. Therefore, if we shift the predicted values back by one unit, the model's accuracy is expected to improve. After performing this adjustment, plotting the confusion matrix, and recalculating the model's accuracy, we observed that the accuracy increased to 93.2%, which is satisfactory.

IV. SIMULATION AND RESULTS

A. Prediction Using Labeled Data

In the first stage of simulation, 702 records of *Shapna* stock and 672 records of the total index were tested after the preprocessing steps. Of these, 520 samples were used as the training set, and the remaining records were used as the test set. Figure 6 shows the prediction chart of *Shapna* stock in the first stage of testing. Some initial simulation settings are listed in Table 2.

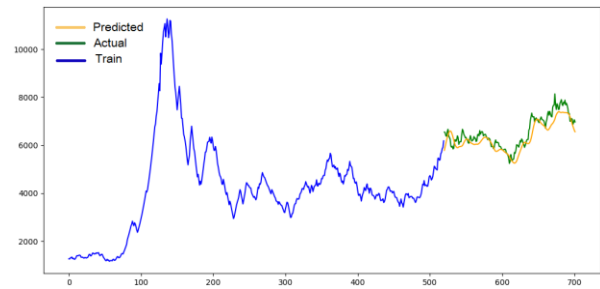


Fig.6. *Shapna* stock prediction chart with tanh activation function

TABLE II

SETTINGS AND SIMULATION RESULTS FOR *SHAPNA* STOCK AND THE TOTAL INDEX IN THE FIRST STAGE.

Acc	DTW	T	Af	Train	n	stock
38.46	2408	6.5	tanh	520	702	<i>Shapna</i>
40.65	18325	80	sigmoid			
38.46	2490	63	ReLU			
61.84	280013	7.4	tanh	520	672	<i>total index</i>
66.44	2876597	87	sigmoid			
61.18	331525	63	ReLU			

*Acc: Accuracy, T: Time, Af: Activation function, n: number of samples

According to the results shown in Table 2, at first glance, it appears that using the sigmoid function can lead to better accuracy. However, upon examining the

prediction chart for the sigmoid function (Figure 7), it becomes evident that the predicted vector does not resemble the actual vector at all, and the relatively better accuracy is merely artificial. The DTW value obtained with the sigmoid function is also significantly higher than the other two functions, indicating a considerable distance between the predicted and actual vectors. Therefore, we excluded the sigmoid function from further computations.

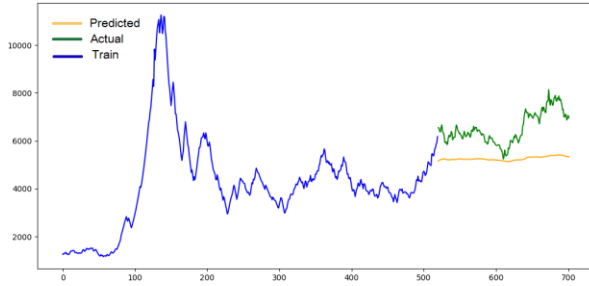


Fig.7. Prediction chart of Shapna stock with sigmoid activation function

We selected tanh between the tanh and ReLU functions because the DTW values and accuracy are very similar. Still, the prediction time with the ReLU function is approximately 9 to 10 times longer than with the tanh function. For this reason, we chose Tanh for subsequent simulations. Figure 8 shows the prediction chart of the *Shapna* stock using the ReLU activation function.

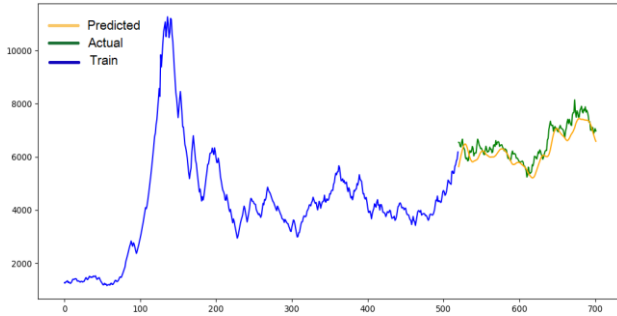


Fig.8. Prediction chart for *Shapna* stock using the ReLU activation function

In the previous stage, the prediction accuracy for *Shapna* stock was approximately 38%, while the total index reached about 62%, which is not acceptable. We hypothesized that better results might be achieved by adjusting the epoch parameter and increasing the number of learning iterations. However, despite this expectation, no significant improvement was observed in this simulation phase. Moreover, increasing the epoch parameter only resulted in longer prediction

times. Upon reviewing the results and analyzing the confusion matrix, we suspected that one of the reasons for the low prediction accuracy in different models was the use of three classes. When the number of classes increases, the probability of correct predictions decreases, resulting in lower model accuracy. Therefore, we repeated the simulation by changing the classification from three classes (1, 0, -1) to two classes (1, -1). The result using tanh for 2 and 3 classes shown in Table 3.

TABLE III
SIMULATION SETTINGS AND RESULTS FOR SHAPNA STOCK AND THE TOTAL INDEX AFTER SWITCHING FROM THREE CLASSES TO TWO CLASSES

Acc	DTW	T	C	Train	n	stock
37.36	1222	15.9	3	520	702	<i>Shapna</i>
48.9	1745	22.7	2			
62.5	274819	19.7	3	520	672	<i>total index</i>
60.5	128584	20.4	2			

*Acc: Accuracy, T: Time, C:class, n: number of samples

For *Shapna* stock, accuracy increased significantly from 37% to approximately 49%. However, despite a substantial reduction in DTW from 270,000 to 128,000 for the total index—indicating a closer alignment between the predicted vector and the actual vector—there was little change in accuracy. This raises questions: Why is there a significant difference (over 11%) in model accuracy between the total index and *Shapna* stock? And why does changing the classification from three classes to two classes lead to a substantial improvement in accuracy for *Shapna* stock (over 25%) but not for the total index (where accuracy decreased by 2%)?

By comparing the prediction charts for *Shapna* stock and the total index in Figure 9, we observe that the total index follows an almost upward trend with less volatility compared to *Shapna* stock. This may explain why the change in classification had a more significant impact on *Shapna* stock, which is more volatile.

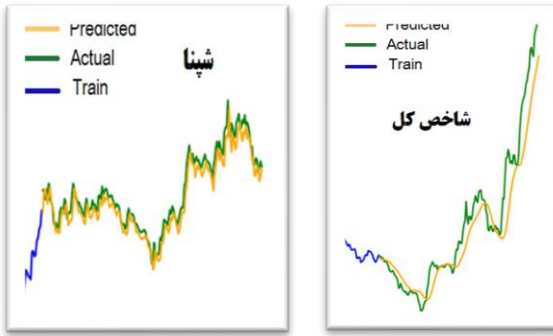


Fig.9. Comparison of the growth trend of *Shapna* stock and the total index

By examining the chart for *Shapna* stock, the area marked with a red box appears to deviate from the usual trend of *Shapna* stock, shown in the purple box. This suggests that the outlier data might have caused incorrect training and impacted the model’s accuracy. To address this, we removed the outlier data and repeated the simulation.



Fig.10. Forecasting *Shapna* stock with out-of-range data

After removing the outliers, the model's accuracy improved slightly from 51% to 57%, but it was still far from satisfactory. By zooming in on the output plot and examining the result in Figure 11, we observed that the predicted vector was very similar to the actual vector. This raised the question: Why is the model’s accuracy low despite such a high degree of similarity?

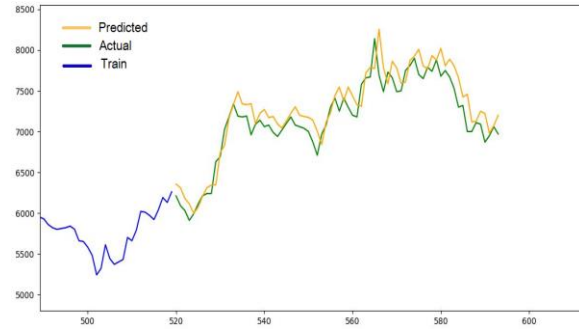


Fig.11. Magnified *Shapna* stock prediction chart

We compared the actual and predicted labels to answer this question. Figure 12 shows that nearly all the actual and predicted classes—except for a few—matched with a one-unit time lag. In other words, the predicted vector was phase-shifted by one unit compared to the actual vector.



Fig.12. Comparison of predicted classes and actual classes

Therefore, if we shift the predicted values back by one unit, we can expect the model’s accuracy to improve. We moved the first predicted label to the end of the label list to ensure the number of predicted labels remained equal to the number of actual labels.



Fig.13. Moving the first element of the class to the end of the list

After shifting the predicted values, plotting the confusion matrix (Figure 14), and recalculating the model's accuracy, we observed that the accuracy increased to **93.2%**, which is a satisfactory result.

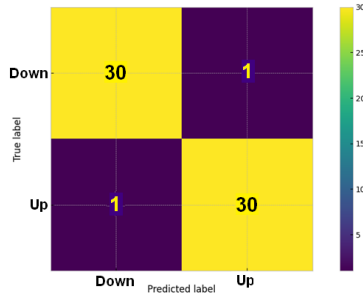


Fig.14. The confusion matrix of *Shepna's* contribution after shifting the predicted labels backward

B. Close Price Prediction

This section presents the prediction of the close price based on data obtained from the stock exchange website. Table 4 includes the layer type, output shape, and the number of parameters. The total number of parameters is 30,651 (119.73 KB), with 30,651 (119.73 KB) trainable parameters and zero non-trainable parameters. A sequential model was used. The number of epochs for training was set to 150, and the cost function was defined as Mean Squared Error (MSE).

TABLE IV
LSTM MODEL PARAMETERS

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 50)	10,400
dropout (Dropout)	(None, 60, 50)	0
lstm_1 (LSTM)	(None, 50)	20,200
dropout_1 (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51

The results were obtained using two activation functions: Sigmoid and ReLU (Rectified Linear Unit). Generally, the standard rule for data splitting is 80/20—where 80% of the data is used for training a model, while 20% is used for testing it. This rule depends on the dataset being used but is widely shared and effective for most datasets. Selecting an appropriate optimization algorithm for a deep learning model is crucial and significantly impacts the time required to achieve the desired results. The Adam optimizer is considered a generalized version of the Stochastic Gradient Descent (SGD) algorithm and has

recently been widely used for deep learning applications. The Root Mean Square Propagation (RMSprop) algorithm performs well in online and unstable problems, such as noisy data. In this method, the learning rates for each parameter are maintained, and these rates are adapted based on the average of recent gradient values associated with the weights. To predict the close price of *Shapna* stock without considering labels, the LSTM network was trained with different optimizer parameters. Table 5 shows the results of the close price prediction for *Shapna* stock.

TABLE V
THE RESULTS OF SHAPNA STOCK PREDICTION USING DIFFERENT OPTIMIZERS

Op	Af	TP	TN	FP	FN	Acc	Pre	Rec	F1
Adam	Sig	104	105	6	6	95.00	94.54	95.41	94.97
	Relu	105	106	5	4	95.90	95.45	96.33	95.89
SGD	Sig	103	106	7	4	95.00	93.63	96.26	94.93
	Relu	104	106	6	4	95.45	94.54	96.29	95.41
RMS prop	Sig	106	105	4	5	95.90	96.36	95.49	95.92
	Relu	105	106	5	4	95.90	95.45	96.33	95.89

*Op: Optimizer, Af: activation function, Sig: Sigmoid, Acc: Accuracy, Pre: Precision, Rec: Recall, F1:F1-Score

Based on the results in Table 5, the Adam optimizer with the ReLU function and the RMSprop optimizer with the Sigmoid and ReLU functions achieved the highest accuracy. However, Adam and SGD with the Sigmoid function had the lowest accuracy. RMSprop obtained the highest Precision and F1-Score with the Sigmoid function, whereas the lowest Precision was observed with SGD. Adam and RMSprop with the ReLU function achieved the maximum Recall, while Adam with the Sigmoid function recorded the minimum.

V. CONCLUSION

With data mining and artificial intelligence advancement, numerous tools and methods have emerged for predicting and discovering future stock prices. One such tool is artificial neural networks, including LSTM, which have proven effective for time-series data mining. In this study, we briefly explained the LSTM network and used data mining techniques to analyze and predict the performance of stocks and the Tehran Stock Exchange index. To begin, we implemented a simple LSTM network, achieving an initial prediction accuracy of approximately 30%. At this stage, data classification was performed using three classes. The choice of activation function plays a crucial role in predictions.

The results showed that the tanh activation function outperformed sigmoid and ReLU, offering both faster speed and higher accuracy in identifying the correct predicted classes. Adjusting the epoch parameter during simulations demonstrated that selecting the optimal number of training iterations significantly impacts model accuracy. Increasing the epoch parameter improves accuracy up to a certain point, beyond which further increases yield diminishing returns and only prolongs the prediction process. Additionally, selecting an appropriate amount of training data significantly affects the outcomes. Removing outliers before the training process reduces factors that cause errors, which can slightly improve accuracy. In conclusion, one of the most influential aspects of the LSTM network prediction process is selecting the appropriate classification method and adequately tuning the network parameters. In a subsequent experiment, the close price prediction was performed using unlabeled data from the stock exchange, yielding better results compared to previous tests. For future work, it is recommended to include data with key features for each stock, such as actual and legal purchase volumes, RSI, MACD, etc., which can be labeled based on profit levels. Additionally, feature selection techniques can be employed to choose the most essential features and achieve optimal classification outputs.

REFERENCES

1. Yu, P. and X. Yan, *Stock price prediction based on deep neural networks*. Neural Computing and Applications, 2020. **32**(6): p. 1609-1628.
2. Vismayaa, V., et al., *Classifier based stock trading recommender systems for Indian stocks: An empirical evaluation*. Computational Economics, 2020. **55**(3): p. 901-923.
3. Nair, B.B., et al., *A stock trading recommender system based on temporal association rule mining*. SAGE Open, 2015. **5**(2): p. 2158244015579941.
4. Möws, B., *Deep Learning for Stock Market Prediction: Exploiting Time-Shifted Correlations of Stock Price Gradients*. 2016.
5. Bohn, T.A., *Improving long term stock market prediction with text analysis*. 2017.
6. De Rossi, G., J. Kolodziej, and G. Brar, *A recommender system for active stock selection*. Computational Management Science, 2019: p. 1-31.
7. Patel, J., et al., *Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques*. Expert systems with applications, 2015. **42**(1): p. 259-268.
8. Khonsha, S., M. Agha Sarram, and R. Sheikhpour, *A Profitable Portfolio Allocation Strategy Based on Money Net-Flow Adjusted Deep Reinforcement Learning*. Iranian Journal of Finance, 2023. **7**(4): p. 59-89.
9. Kalra, R., et al., *An efficient hybrid approach for forecasting real-time stock market indices*. Journal of King Saud University-Computer and Information Sciences, 2024: p. 102180.
10. Varadharajan, V., et al., *Stock Closing Price and Trend Prediction with LSTM-RNN*. Journal of Artificial Intelligence and Big Data, 2024: p. 1-13.
11. Safari, A. and M.A. Badamchizadeh, *DeepInvest: Stock Market Predictions with a Sequence-Oriented BiLSTM Stacked Model—A Dataset Case Study of AMZN*. Intelligent Systems with Applications, 2024: p. 200439.
12. Janocha, K. and W.M. Czarnecki, *On loss functions for deep neural networks in classification*. arXiv preprint arXiv:1702.05659, 2017.
13. Bennett, K.P. and E. Parrado-Hernández, *The interplay of optimization and machine learning research*. The Journal of Machine Learning Research, 2006. **7**: p. 1265-1281.
14. Nanopoulos, A., R. Alcock, and Y. Manolopoulos, *Feature-based classification of time-series data*. International Journal of Computer Research, 2001. **10**(3): p. 49-61.
15. Hochreiter, S. and J. Schmidhuber, *Long short-term memory*. Neural Comput, 1997. **9**(8): p. 1735-80.
16. Gers, F.A. and E. Schmidhuber, *LSTM recurrent networks learn simple context-free and context-sensitive languages*. IEEE Transactions on Neural Networks, 2001. **12**(6): p. 1333-1340.
17. Graves, A. and A. Graves, *Long short-term memory*. Supervised sequence labelling with recurrent neural networks, 2012: p. 37-45.
18. Senin, P., *Dynamic time warping algorithm review*. Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, 2008. **855**(1-23): p. 40.
- [16] Wang, J., Hong, S., Dong, Y., Li, Z., & Hu, J. (2024). Predicting Stock Market Trends Using LSTM Networks: Overcoming RNN Limitations for Improved Financial Forecasting.
- [17] Song, J., Cheng, Q., Bai, X., Jiang, W., & Su, G. (2024). LSTM-Based Deep Learning Model for Financial Market Stock Price Prediction. Journal of Economic Theory and Business Management, 1(2), 43–50.