

Research Article

Machine Learning-Based Decision-Making for Stock Trading: Case Study for Automated Trading in Saudi Stock Exchange

Mohammad Alsulmi  and **Nourah Al-Shahrani**

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia

Correspondence should be addressed to Mohammad Alsulmi; malsulmi@ksu.edu.sa

Received 26 February 2022; Revised 18 June 2022; Accepted 27 August 2022; Published 22 September 2022

Academic Editor: Rahman Ali

Copyright © 2022 Mohammad Alsulmi and Nourah Al-Shahrani. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Stock markets are becoming the center of attention for many investors and hedge funds, providing them with a wide range of tools and investment opportunities to grow their wealth and participate in the economy. However, investing in the stock market is not trivial. Stock traders and financial advisors are required to frequently monitor market actions, search for profitable companies, and analyze stock price movements to generate various trading ideas (e.g., selecting a stock symbol and making the decision when to enter or exit a trade), potentially leading to investment returns. Therefore, this study aims to address this challenge through exploring the adaptation of machine learning methods combined with risk management techniques to develop a framework for automating the task of stock trading. We evaluated our framework by creating a diverse portfolio containing several companies listed on the Saudi Stock Exchange (Tadawul) and using the simulated trading actions (executed by the framework) to estimate the portfolio's returns for 3.7 years. The findings show that in terms of investment returns, the proposed framework is very promising; it has generated over 86% returns and outperformed almost all hedge funds by the top investment banks in Saudi Arabia.

1. Introduction

Financial markets are becoming an important part of the global economy, playing a crucial role in the growth and prosperity of many countries around the world [1, 2]. These markets provide investors with free and open platforms for exchanging different commodities and securities (e.g., stocks) from a variety of sectors, including healthcare, manufacturing, financial and banking services, information technology, and telecommunications. Financial markets, and particularly stock exchanges, are known to produce higher investment returns relative to other types of investments [3–5]; this makes them the center of attention for many investors and hedge funds and provides them with a wide range of tools and opportunities to make investments and grow their wealth. However, one of the challenges is that investing in stock markets is a complex and time-consuming task; it requires investors and stock market traders to continuously monitor market actions and, frequently,

analyze the price movements of stocks, looking for potentially profitable trades (i.e., by selecting a stock symbol and making the decision when to commit or abandon a trade) that increase their chances of generating investment returns and reduce the risk of losses [6, 7].

With recent advancements in artificial intelligence (AI) and machine learning (ML) techniques, many aspects of them have been widely adopted to address some of the limitations and challenges in various domains, including the financial analytics sector [6–9]. Therefore, in this study, we develop a framework that relies on ML, as well as risk control and management principles, to automate the decision-making process for the stock market trading task (i.e., deciding when to purchase, monitor, and sell stocks), aiming to reduce its complexity. The proposed framework is applied to the Saudi Stock Exchange (Tadawul) by creating a portfolio consisting of 10 stock symbols and conducting various trading actions. It is applied over a period of 3.7 years (starting from January 2018 and ending in August 2021) so

that the performance of the portfolio is estimated over the entire period of time. The findings of our study suggest that the framework is very promising for real-life trading scenarios, generating investment returns of up to 86% for that period of time. Moreover, after comparing the framework to several hedge funds managed by top investment banks in Saudi Arabia, it is shown that its performance significantly exceeds the performance of almost all of these funds. Also, it is shown that the active trading strategy adopted by the framework can lead to investment returns that are higher than the typical passive buy-hold strategy.

The remainder of this paper is organized as follows. In Section 2, we conduct a literature review and examine prior studies related to stock market predictions. Section 3 introduces our methodology for developing our trading framework, explains its implementation, and examines the application of model learning to predict the stock price direction. Section 4 evaluates our learning model and presents a case study for conducting trading by applying the proposed framework. Finally, in Section 5, we summarize the study and provide concluding remarks.

2. Related Work

The importance of financial markets (and specifically stock exchange markets) and their significant impact on the global economy [1, 2] have motivated researchers worldwide to examine a wide range of computational intelligence methods to study and analyze stock market behavior and movements. They explore various ML methods to forecast future stock and index values and predict stock price movements [6–16]. For instance, in the study presented in [9], the authors explored time-series forecasting relying on neural networks, including the multi-layer perceptron (MLP) and dynamic architecture for artificial neural networks (DAN2). Their aim was to predict future values of the NASDAQ market index, and their findings suggest that a simple MLP is sufficient for forecasting the NASDAQ index value. Likewise, Chiang et al. [10] explored the MLP model combined with particle swarm optimization to forecast the direction of several U.S. market indices, including NASDAQ and SP500. Unlike Güreşen et al. [9], this study attempted to forecast the direction of index movement rather than its value and use it to signal entry and exit trading actions.

Other international markets have also been studied, and several approaches have been proposed, as presented in [11–16]. For example, Peng et al. [11] examined Chinese stock markets and proposed a long short-term memory (LSTM) neural network model incorporating a few input indicators to forecast stock price values. In [12], the authors explored the adaptation of two ML methods—i.e., MLP and support vector machines (SVMs)—to predict stock price directions for the Karachi Stock Exchange in Pakistan. Moreover, Hiransha et al. [13] and Agrawal et al. [14] proposed several deep learning models, including MLP and LSTM, and relied on a few technical indicators, such as the moving average (MA) and moving average convergence divergence (MACD), to predict stock prices for the Indian national stock market. All the aforementioned approaches

produced promising results considering the stock trading task.

For the Saudi Stock Exchange (Tadawul), a few studies have attempted to examine stock price movements [15, 16]. Alturki et al. [15] examined LSTM with several indicators, including stock price, moving average, and MACD, to predict buy-and-sell signals. Using accuracy and investment returns for evaluation, the proposed approach is shown to produce results that are comparable to those of the buy-hold strategy. Another study by Alsubaie et al. [16] explored various technical indicators in addition to several learning models including MLP, SVM, and naïve Bayes. They used these models to predict stock price movements for the Saudi Stock Exchange and evaluated their effectiveness using accuracy and by simulating an investor’s trading actions. Their findings indicated that the naïve Bayes model outperforms other learning models, resulting in higher investment returns.

To enhance the effectiveness of model learning and reduce the complexity of decision-making during the trading process, we introduce in this study a framework for conducting automated trading actions through incorporating model learning that relies on various technical indicators. Unlike prior studies (especially for the Saudi Stock Exchange), we further explore the adaptation of various ML models and apply them in a progressive learning manner. Moreover, we study the impact of adopting several risk control management techniques on the effectiveness of our framework. Lastly, we conducted a comprehensive study to evaluate the proposed methods by considering a longer testing period and making a more realistic analysis of the performance of the framework.

3. Methodology

Initially, we address the research problem by proposing a tool for automated stock market trading. The tool allows stock market investors to create investment portfolios and select a set of companies (i.e., symbols) from a target stock exchange market. Then, it automatically performs trading actions by monitoring stock and making the decisions on when to purchase and sell shares. This tool was developed through several steps that are described as follows.

First, we collect historical data from a target stock market and perform data preprocessing to prepare the data for an ML task. Then, after defining our learning as a supervised ML task, we build a model to learn from historical data and predict stock price movements (whether a price is moving up or down for the next day). Finally, we implement a trading strategy that relies on the stock price movement signals produced by the learned model to conduct trading actions through buying and selling stocks. All of these steps are explained in the following sub-sections 3.1–3.3.

3.1. Data Acquisition and Preprocessing. The target stock exchange market for this study is the Saudi Stock Exchange (Tadawul). We rely on historical market data released by the Saudi Capital Market authority [17]. We selected data

TABLE 1: Sample of historical Saudi stock exchange data. Stock prices are reported in Saudi Riyal (SAR).

Company name	Ticker ID	Date	Open price	Highest price	Lowest price	Close price	Volume
Sabic	2010	20180101	101.8	102.0	100.8	101.4	4162501
Sabic	2010	20180102	101.4	101.6	100.8	100.9	5639868
Sabic	2010	20180103	101.0	102.0	101.0	101.5	7413881
Jarir	4190	20180101	146.0	146.6	146.0	146.6	5669
Jarir	4190	20180102	146.6	146.6	145.0	145.5	8050
Jarir	4190	20180103	145.4	146.0	145.0	145.3	10371
Alrajhi	1120	20180101	64.6	65.2	64.1	65.0	2788920
Alrajhi	1120	20180102	65.1	65.3	64.6	64.6	2605433
Alrajhi	1120	20180103	64.6	66.3	64.6	65.9	3992279
STC	7010	20180101	68.5	68.5	68.1	68.1	208241
STC	7010	20180102	68.2	68.2	66.9	67.1	178184
STC	7010	20180103	67.5	67.5	66.9	67.4	164584

instances from the last 12 years, spanning a period from January 2010 to August 2021. The data consist of daily trading information for all 209 companies listed in Tadawul. Each instance is represented by several attributes, including company name (i.e., the listed name in the market), company ticker (i.e., unique identification of the company), trading date, open price, highest price, lowest price, close price, and volume of shares traded for that day. Table 1 shows a sample of historical Saudi Stock Exchange data included in this study. Once we acquire the data, and because the data need preparation to be ready for the task of learning to predict stock price movements, we apply three stages of preprocessing. First, we label and annotate the data instances with the ground-truth labels; then, we extract and generate feature values for each instance. Finally, we normalize and put the data into the correct scale, making it ready for supervised model learning.

3.1.1. Data Labeling. One of the challenges when dealing with stock market data is how to label each daily trading instance, so that it can be used effectively in a supervised learning setting. Several labeling methods have been explored previously to tackle this issue [13, 16, 18, 19]; nonetheless, we followed the approach presented in [18] because of its simplicity and effectiveness in producing good labels. This approach works by defining a window size n of the future stock prices and labeling a data instance (representing a day) based on whether the direction of future stock prices (within the defined time window n) tends to increase or decrease. Thus, to apply it to our problem, we defined two labels {UP, DOWN}; hence, our problem can be regarded as a binary classification problem. We use the close prices of a stock to label each instance such that an instance x (a day in the dataset) is labeled by looking forward to the average value of the close prices of n consecutive days. If the average is higher than the closing price of day x , then x is labeled as “UP,” indicating that the price of the stock rises shortly; otherwise, it is labeled as “DOWN” to indicate that the future stock price is either moving down or will have no significant movements.

Choosing a suitable window size n for computing the average future close prices is performed by exploring a range

of window size values and selecting the value that maximizes the given objective function f . We use the objective function as defined in [19], which maximizes trade profitability over a labeling sequence for a given stock. After exploring several values in the range between 5 and 9, we selected a window size of 8 as it maximized f for our dataset. The results of applying this labeling strategy to a sample of four companies (symbols) in our dataset are summarized in Table 2. From the table, we can see that such a labeling approach resulted in a well-balanced dataset and no further re-balancing is needed.

3.1.2. Feature Generation. After having labeled the data (as discussed in Section 3.1.1), we now describe the feature set that we use to represent the data instances. Generally, candidate features should be able to represent a set of properties of stock movement for a given period of time. They can be as simple as general statistics about the symbol on a trading day (e.g., open price, highest price, volume) or more complex performance indicators (known as technical indicators) that capture the trend of the symbol concerning previous trading days [19, 20]. These indicators are generally computed based on open, close, and highest price statistics, in addition to the volume for several days. It is also noteworthy that technical indicators are well known among stock investors and traders as they are one of the main tools they rely on deciding whether to invest in or trade a particular stock [20, 21].

We rely on the general statistics of the stock for a particular day (i.e., open price, highest price, lowest price, close price, and volume) that were originally part of the dataset provided by Tadawul. In addition, we computed a diverse set of technical indicators and used them along with general stock statistics. These indicators cover various technical analysis properties of stocks, such as momentum indicators (assessing the speed of stock value change), volume indicators (capturing the volume of buying and selling a stock), trend indicators (following the pattern of stock prices), and volatility indicators (evaluating the vacillation scope of stock prices) [20, 21]. Overall, we utilized 31 features listed in Table 3. Moreover, the detailed descriptions of the considered indicators can be found in [20, 21].

TABLE 2: The results of labeling data instances for the companies Sabic, Jarir, Alrajhi, and STC with UP and DOWN labels for the period from January 2010 to August 2021.

Label distribution (%)		Symbol	Ticker ID
UP (%)	Down (%)		
56.3	43.7	Sabic	2010
57.0	43.0	Jarir	4190
58.1	41.9	Alrajhi	1120
51.7	48.3	STC	7010

TABLE 3: The set of 31 learning features utilized by this study.

Feature	Description	Definition
Open	Open price	A stock opening price for a given day (P_{Open})
Highest	Highest price	A stock highest price for a given day (P_{High})
Lowest	Lowest price	A stock lowest price for a given day (P_{Low})
Close	Close price	A stock closing price for a given day (P_{Close})
Volume	Trading volume	Number of stock shares traded for a given day (Vol_t)
Direction	Previous day's direction	Previous day's price direction up or down
SMA	Simple moving average	$SMA(p_t) = p_1 + p_2 + \dots + p_t/n$, p_t is the price for day t , n represents periods
EMA	Exponential moving average	$EMA(p_t) = (p_t \times 2/1 + n) + EMA_{t-1} \times (1 - 2/1 + n)$
TEMA	Triple exponential moving average	$TEMA_t = 3 * EMA(p_t) - 2 * EMA(EMA_t) + EMA(EMA_t)$
DEMA	Double exponential moving average	$DEMA_t = 2 * EMA(p_t) - EMA(EMA_t)$
TRIMA	Triangular moving average	$TRIMA_t = SMA_1 + SMA_2 + \dots + SMA_{t-1} + SMA_t / (n + 1) / 2$
MACD	Moving average convergence divergence	$MACD_t = EMA_{12 \text{ periods}} - EMA_{26 \text{ periods}}$
TP	Typical price indicator	$TP_t = \sum_{i=1}^n p_{t,High} - p_{t,Low} - p_{t,Close} / 3$
WC	Weighted close price	$WC_t = (p_{t,Close} * 2) + p_{t,High} + p_{t,Low} / 4$
WR	Williams' %R	$WR_t = (maxPrice - p_t / maxPrice - minPrice) \times -100$
SO	Stochastic oscillator	$SO_t = (p_t - minPrice / maxPrice - minPrice) \times 100$
OBV	On balance volume	$OBV_t = OBV_{t-1} (+ -)Vol_t$
ROC	Rate of change indicator	$ROC_t = (p_t - p_{t-n} / p_{t-n}) \times 100$
CCI	Commodity channel index	$CCI_t = TP_t - SMA(TP_t) / 0.015 \times Mean \ Deviation$
MFI	Money flow index	$MFI_t = 100 - 100 / (1 + Money \ Flow \ Ratio)$
RSI	Relative strength index	$RSI_t = 100 - (100 / 1 + AvgGain / AvgLoss)$
AD	Accumulation distribution	$AD_t = AD_{prev} + ((p_{Close} - p_{Low}) - (p_{High} - p_{Close})) / (p_{High} - p_{Low}) \times Vol$
CMO	Chande momentum oscillator	$CMO_t = SumHighestClose - SumLowestClose / SumHighestClose + SumHighestClose \times 100$
PPO	Percentage price oscillator	$PPO_t = MACD_t / EMA_{26 \text{ periods}} \times 100$
TR	True range	$TR_t = \text{Max}(p_{t,High} - p_{t,Low} , p_{t,High} - p_{t,Close} , p_{t,Low} - p_{t,Close})$
ATR	Average true range	$ATR_t = TR_1 + TR_2 + \dots + TR_t / n$
AU	Aroon up	$AU_t = 25 - H/n \times 100$, H is the number of days since the highest price value
AD	Aroon down	$AD_t = 25 - L/n \times 100$, L is the number of days since the lowest price value
BOP	Balance of power	$BOP_t = (p_{t,Close} - p_{t,Open}) / (p_{t,High} - p_{t,Low})$
BOLU	Upper Bollinger band	$BOLU_t = SMA(TP_t) + 2 * \sigma_{TP}$
BOLD	Lower Bollinger band	$BOLD_t = SMA(TP_t) - 2 * \sigma_{TP}$

3.1.3. *Data Scaling and Normalization.* The last step in preparing the data for our learning task involves normalizing and re-scaling the data instances. This is required to ensure the generalization of the learned data and increase its prediction effectiveness for new unseen data instances. The normalization we apply is that instead of directly using the actual feature values of an instance, we use the daily change in a feature value (expressed in %); e.g., we use the change in the stock price between two consecutive days as a feature for the later day. This is applied to all features to ensure that a learning model is not biased toward actual feature values (e.g., actual stock prices) but rather learns from the change in feature values from two consecutive days. Furthermore, we standardize the dataset by applying min-max normalization

[22] to re-scale the different features and put them in the same value range (all values will be in the range between 0 and 1).

3.2. *Supervised Learning for Stock Price Movement Prediction.* The next step is to leverage the preprocessed data described in Section 3.1 in learning a model that can be used to predict stock price movements. Such a model can be used to predict the future direction of a stock price (for instance, to predict whether a stock price is moving up or down the following day). The supervised learning task requires training a model using a set of data instances represented by feature values and annotated with a predefined set of labels (in our case,

two labels: “UP” and “DOWN”). Therefore, the data were represented as follows:

$$F_i = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,m} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,m} \\ f_{3,1} & f_{3,2} & f_{3,3} & \cdots & f_{3,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{n-1,1} & f_{n-1,2} & f_{n-1,3} & \cdots & f_{n-1,m} \\ f_{n,1} & f_{n,2} & f_{n,3} & \cdots & f_{n,m} \end{bmatrix}, \quad (1)$$

$$Y_i = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix},$$

where for stock i , F_i contains the feature values of its data instances represented as a consecutive series of n days, and each row $[f_{j,1} \ f_{j,2} \ f_{j,3} \ \cdots \ f_{j,m}]$ in F_i represents a day. Y_i comprises a set of annotated ground-truth labels such that each row in Y_i corresponds to a data instance in F_i (i.e., each value j in Y_i represents a label for day j in F_i).

Using this representation, we can apply supervised machine learning models to learn the prediction models for our task. We considered four commonly used machine learning models that are very effective for a variety of tasks. These are the support vector machine (SVM), random forest (RF), artificial neural network (ANN), and long short-term memory (LSTM). SVM learns by minimizing classification errors through constructing a hyperplane or a set of hyperplanes that maximizes the geometrical separation of data into different classes [23, 24]. On the other hand, RF learns by utilizing many randomly generated trees, such that the popular voting of these trees is used to classify instances [25]. Both the ANN and LSTM learn by building a neural network consisting of an input layer, one or more hidden layers, and an output layer. The ANN uses a simple neural network architecture with a feed-forward mechanism, whereas the LSTM utilizes a recurrent neural network and memory cells in the hidden layer to store previous information [26, 27]. The descriptions of these learners can be found in the studies presented in [23–27].

Once a model is learned, it can be used to predict the stock price movement (either “UP” or “DOWN”) for a new unseen data instance (i.e., predict a label y_{n+1} for a vector $[f_{n+1,1} \ f_{n+1,2} \ f_{n+1,3} \ \cdots \ f_{n+1,m}]$ that is in F_i), which is typically the next day in a series of consecutive trading days. The predicted labels (i.e., label per day) produced by the model can be seen as signals that are later utilized by the trading tool we implemented (described in the next section) to guide our trading strategy, i.e., when to monitor, buy, or sell stock shares.

3.3. Automated Decision-Making for Stock Trading. Finally, we automate the task of trading in the stock market by developing a decision-making tool (i.e., a bot that monitors the market) and use the predictions produced by the underlying learned model (described in Section 3.2) to make trading decisions (i.e., whether to purchase, sell, or keep monitoring the stock), making potentially positive returns while executing these trades. In our tool, we incorporate trading risk management principles that are widely adopted by stock traders and shown to help reduce potential losses and generate substantial profits [29, 30]. We mainly consider three types of principles: (1) setting a limit for the maximum capital to be committed per trade (to control the maximum risk to be taken per trade), (2) setting stop-loss points to limit losses resulting from unprofitable trades, and (3) setting take-profit points to ensure profits before a stock price reverts [28–30]. For a given stock symbol, our trading strategy consists of the following steps:

Step 1. We start with the initial capital investment balance B' in Saudi Riyal (SAR) and select a symbol for trade.

Step 2. Setting of 1 maximum capital risk CR' (in %), (2) stop-loss point SL' (in %), and (3) take-profit point TP' (in %).

Step 3. Assign day to initial date $D' = D$.

Step 4. We learn a model M' using the training data up to day D' , and the model is used to predict the price direction label L' for the next day (i.e., $D' + 1$).

Step 5. Move to the next day, assign day to $D' = D' + 1$, and monitor the symbol price for that day.

Step 6. If there is open trade and either stop-loss point SL' or take-profit point TP' is satisfied, then close trade (sell), update balance B' , and proceed to *Step 4*.

Step 7. If label $L' = UP$ and balance $B' > 0$, then open trade with risk CR' (buy) and update B' ; otherwise, if label $L' = DOWN$ is received k consecutive times, then close trade (sell) and update B' .

Step 8. Go to *Step 4*.

Figure 1 illustrates the procedure for this strategy. It should be noted that the execution of the strategy continues until it is interrupted (e.g., an investor is no longer interested in proceeding with stock trading). In Section 4, we present a case study using this strategy in a more realistic setting to build a portfolio combining several market symbols. Furthermore, we discuss how to set the risk management parameters, CR' , SL' , and TP' .

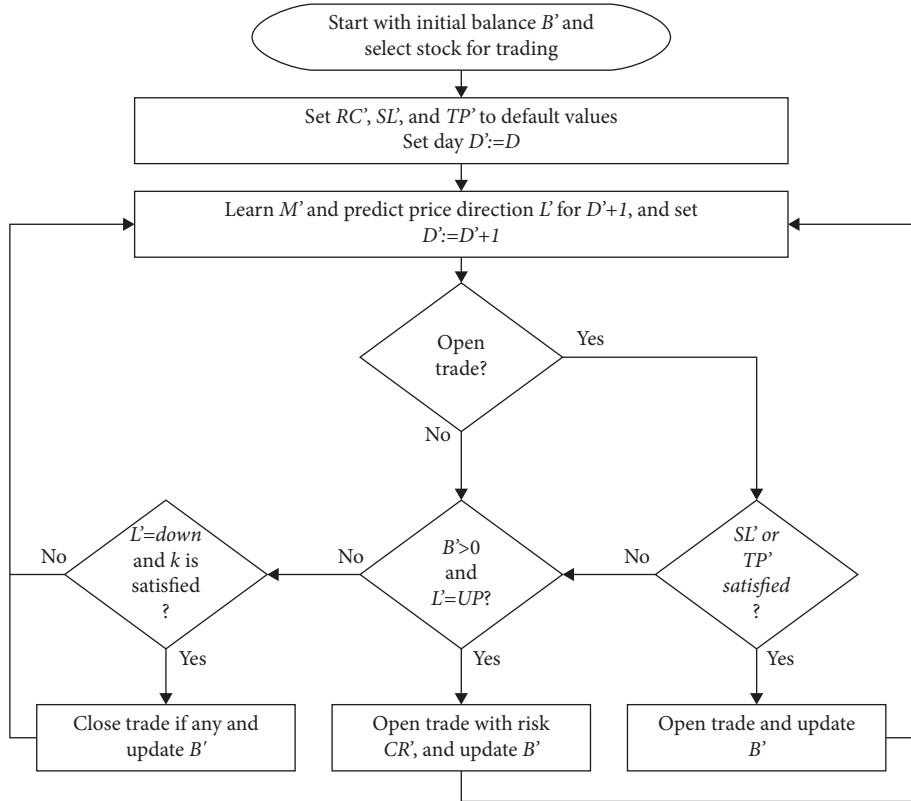


FIGURE 1: Automated trading procedure that relies on predictions by model learning.

4. Results and Discussion

Next, we present an evaluation of the proposed framework. First, the experimental settings are described. We then present and discuss the results of applying our model to predict stock price movements, and later we analyze the effect of features on the prediction quality. Subsequently, we evaluate and analyze our automated trading framework by using it in a more realistic stock trading scenario.

4.1. Experimental Setup. The dataset used in our experiments is historical data for the Saudi Stock Exchange (Tadawul) (described in Section 3.1) for the period from January 2010 to the end of August 2021. This covers 2910 daily trading instances per stock. Two types of experiments were conducted. First, we evaluated the effectiveness of the learned models in predicting stock price directions. For demonstration purposes, we select a sample of four companies: Sabic (materials industry), Jarir (retailing), Alrajhi (banking), and STC (telecommunications). For each of them, we generated a feature set, as described in Section 3.1.2. More specifically, for SMA, EMA, TEMA, DEMA, TRIMA, TP, WR, ROC, CCI, CMO, and ATR, we generated several features for each indicator, capturing different cutoff times (e.g., SMA was calculated for 5, 10, 15, 20, 25, 50, and 100 days). This resulted in a total of 80 features.

We used four models (i.e., SVM, RF, ANN, and LSTM) that were implemented using the WEKA toolkit [31] (we

relied on the WekaDeeplearning4j library [32] to provide support for LSTM in WEKA). We fine-tuned the hyper-parameters of each model to maximize its performance. This is performed by applying 10-fold cross-validation (CV) while exploring a defined range of values for each parameter and then selecting the parameter's value, which maximizes the performance. Particularly, for SVM, we set $C=0.9$ and use a second-degree polynomial kernel. For RF, we set the number of trees (iterations) to 350 while keeping the default values for the other parameters. For the ANN, we set the learning rate to 0.02, the number of epochs to 1500, and used a single hidden layer with the number of nodes = $(\#features + \#classes)/2$, which is the default for WEKA. Finally, for LSTM, we set the number of epochs to 10, the optimization algorithm to Adam, and used a single LSTM layer with the number of nodes = $\#features$, while the activation function for the output layer was set to SoftMax. We do not apply any feature selection, as our initial preliminary experimentation shows that these learners favor more features when considering our task (this will be later explored in Section 4.3).

Further, the second experiment evaluates the effectiveness of our automated trading strategy (as discussed in Section 3.3) by simulating trading actions as an investment portfolio. We set the capital investment balance B' to 100,000 Saudi Riyal (SAR), maximum capital risk CR' to 2%, and k to 5. For stop-loss points, we experimented with several values ranging between 5% and 10%, whereas for take-profit points, the explored values were between 5% and

15%. We set the brokerage fees, deducted by a broker platform, to 0.15% of each trade value (as approved by the Tadawul exchange authority). It should also be noted that we simulate our automated trading (i.e., executing the buying and selling of stock shares) by relying on stock open prices.

4.2. Results and Analysis of Model Learning. We evaluate the performance of the different learning models, SVM, RF, ANN, and LSTM, to predict stock price directions. We reported the results using three metrics: accuracy, precision, and recall. Accuracy measures the number of correctly predicted instances to the total number of all instances, and precision measures the number of correctly predicted instances for one class to the number of predicted instances for that class; recall measures the number of correctly predicted instances for one class to all instances in the dataset with that class. We applied 10-fold cross-validation (CV) such that, in each fold, we trained a model with 90% of the data and tested it on the remaining 10%. Table 4 summarizes the experimental results.

As shown in Table 4, in terms of effectiveness (i.e., the performance of prediction), the examined models achieved surprisingly high performance when predicting stock price movements, reaching up to a 94.1% accuracy, a 94.6% precision, and an approximately 95% recall. Additionally, we see that the RF and SVM achieved an almost equivalent performance (RF and SVM differ slightly for a few instances), and a paired one-sided t -test suggests no significant difference between them. In contrast, we see that both ANN and LSTM result in a noticeably lower prediction performance than the other two for all metrics; also, statistical analysis confirms that the difference in the performance of these two compared to SVM and RF is significant. In addition, in terms of model efficiency (i.e., the time elapsed during training), we see a substantial difference among these learners; i.e., SVM and RF spend less than 20 sec to train, whereas it takes approximately 2 min for LSTM and approximately 5 min for ANN (this is expected because neural network-based models are known to be slower owing to their learning procedure). The results from these experiments suggest that SVM and RF are more suitable, considering our task, as they are shown to be both efficient and effective.

However, a potential limitation of this analysis is that it relies on CV data partitioning to evaluate the generalization performance of these models, knowing that the problem addressed in this study is a time series-based problem (we observe that prior work as well used CV for model evaluation). This is because the CV partitioning mechanism splits the data into training and testing independently of the timely dependency among the data instances; this leads to models that are unable to capture temporal uncertainty when using past data to make future predictions [33]. This can also introduce some learning bias due to learning from data instances that temporally succeed the instances that are used for testing (e.g., learning from the instances of 2012 to 2021 and predicting on 2010 and 2011); this may result in an overestimation of the generalization performance of the learned models.

TABLE 4: Performance results of different learning models for the companies Jarir, Sabic, Alrajhi, and STC applying 10-fold cross-validation. Underlined values represent the model with the highest performance.

Symbol	Ticker ID	Model	Accuracy (%)	Precision (%)	Recall (%)
Jarir	4190	SVM	<u>93.9</u>	92.9	<u>92.9</u>
		RF	<u>93.9</u>	<u>93.0</u>	<u>92.8</u>
		ANN	91.3	90.1	89.8
		LSTM	92.3	90.2	92.3
Sabic	2010	SVM	94.0	<u>94.6</u>	94.7
		RF	<u>94.1</u>	<u>94.6</u>	<u>94.9</u>
		ANN	91.3	92.3	92.3
		LSTM	93.0	93.7	93.9
Alrajhi	1120	SVM	92.3	90.9	<u>90.3</u>
		RF	<u>92.6</u>	<u>91.6</u>	90.2
		ANN	89.8	88.4	86.7
		LSTM	90.6	84.6	91.9
STC	7010	SVM	<u>93.8</u>	<u>93.9</u>	94.2
		RF	93.7	93.6	<u>94.5</u>
		ANN	92.1	92.1	<u>92.2</u>
		LSTM	92.9	93.1	93.4

TABLE 5: Performance results of different learning models for two companies Jarir and Sabic applying temporal partitioning. Underlined values represent the model with the highest performance.

Symbol	Ticker ID	Model	Accuracy (%)	Precision (%)	Recall (%)
Jarir	4190	SVM	73.3	75.8	80.8
		RF	<u>73.5</u>	<u>76.0</u>	<u>81.2</u>
		ANN	71.7	74.6	79.5
		LSTM	72.6	75.8	79.2
Sabic	2010	SVM	69.5	70.4	76.7
		RF	<u>70.1</u>	<u>70.8</u>	<u>77.3</u>
		ANN	67.8	68.8	76.5
		LSTM	69.1	70.5	76.3

To address this limitation, we conducted further experiments considering the temporal constraints of our task. We partitioned the data into two sets: one for training, which covers the period from 2010 to the end of 2017 (approximately 70% of the data), and the other for testing, spanning the period from 2018 to August 2021 (30% of the data). We apply learning in a progressive manner such that, for each day in the testing set, its price direction is predicted by training a model incorporating all the data for the days preceding that day (e.g., predicting the direction of the stock price for August 31, 2021, involves training with all the instances up to August 30th). This resulted in the incremental learning of 914 models representing the learning of a model for each testing day from 2018 to August 2021. The results from these experiments are presented in Table 5 (note that only the results of two symbols are demonstrated as training this large number of models for a given symbol takes several days, particularly for ANN and LSTM).

From Table 5, it is clear that there is a major reduction in performance for all models, suggesting that the performance

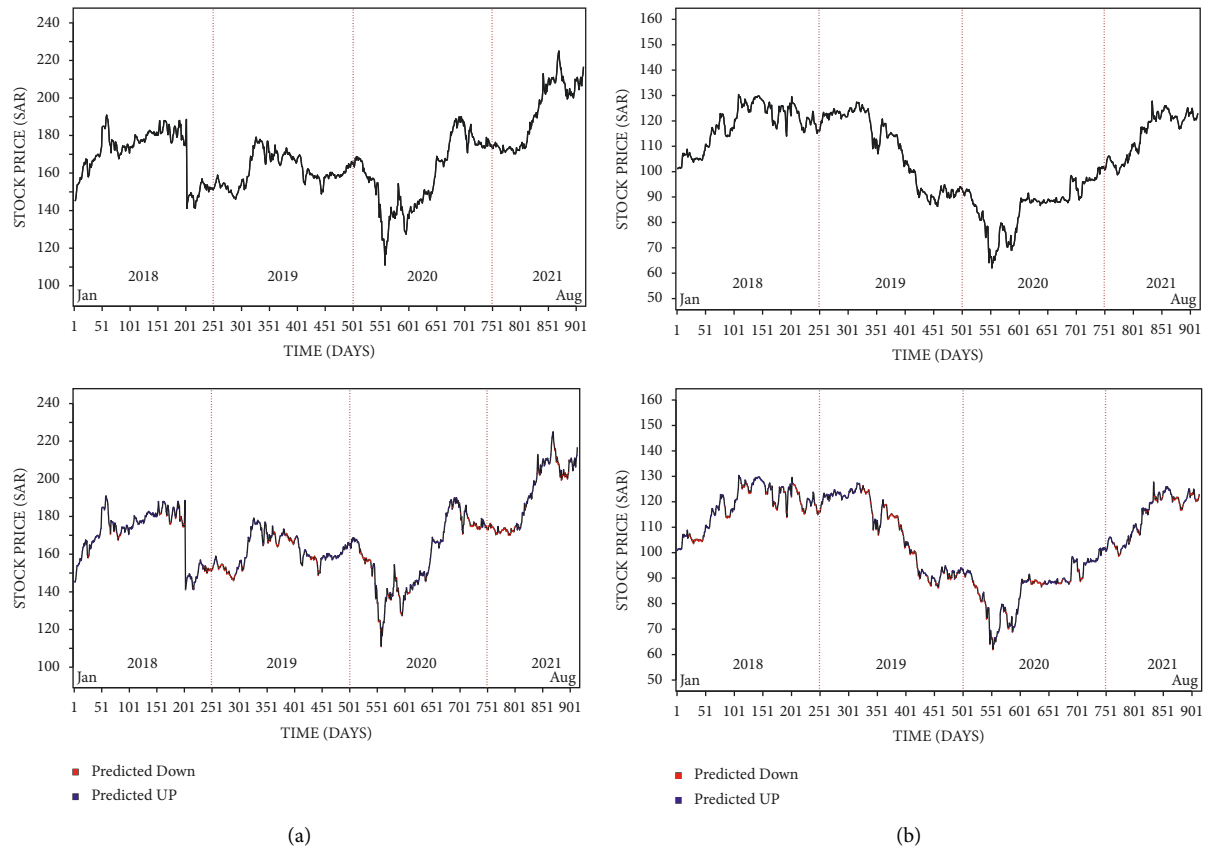


FIGURE 2: Predicted stock price movements (produced by the RF model) are compared to the actual stock closing prices for two stock companies, Jarir (a) and Sabic (b). Red markers correspond to stock price predicted to be down, whereas blue markers correspond to price predicted to move up.

was overestimated previously owing to the adaptation of CV. Nevertheless, the performance achieved in this study is still high considering the nature of the task; in fact, it outperforms the results reported in previous studies [11, 15, 18]. This is also supported by Figures 2(a) and 2(b) as they show that generally, our best performing model is successfully able to predict stock price movements early and just before the actual stock price moves up or down.

Lastly, in terms of which model obtained the highest effectiveness, we noticed a strong correlation of these results with the results of the CV case, as RF and SVM were the highest among the four learners. Later, in Section 4.4, we use the models produced from this experimental setting to build an investment portfolio and make decisions for trading actions.

4.3. The Effect of Features on Model Performance. Having presented that our models resulted in a decent performance for our prediction task, we now explore the effect of different features (i.e., the technical indicators used in our study) on the performance of these models. This is to examine whether the learned models can benefit from a large number of features available during training and achieve high prediction effectiveness. We achieve this by conducting the following experiment.

We start by learning a model, in a progressive manner, for each of the 80 features in our set (i.e., a separate model is learned for each feature), and then we test on the data instances spanning the period from 2018 to August 2021. We average the performance of the resulting models to obtain the mean prediction performance given a single feature. We then examine the performance of model learning while increasing the number of features. This is done by iteratively adding more features to models, re-train them, and re-test them, which continues until all the 80 features are included in a single model. Applying our experiment this way allows us to find the mean prediction performance as the number of features is increased from 1 to 80. Because the number of possible models to be selected during each iteration is enormous (e.g., to train a model with 20 features, there are $\binom{80}{20}$ possible feature subsets, which is over a billion), we randomly select 10 feature subsets and average the prediction performance over the 10 resulting models.

We applied the aforementioned procedure to the Jarir stock symbol using RF as our main learner and we model the performance of learned models using accuracy, precision, and recall (it should be noted that running this procedure for a single stock symbol takes over 10 days on a single PC). The result is illustrated in Figures 3(a) and 3(b) above. It is worth

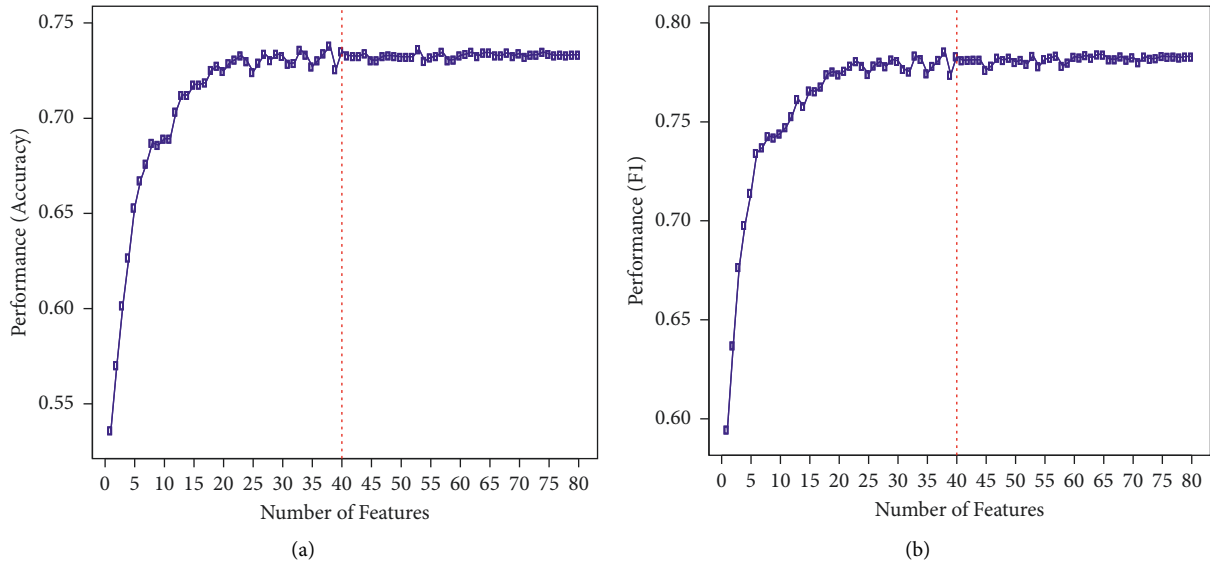


FIGURE 3: Modeling the performance of stock price direction prediction with the increasing number of features used in model learning. Two performance metrics are used: accuracy (a) and $F1$ (b).

noting that we combined both precision and recall using the $F1$ measure, which allows us to plot them as a single performance measure. From Figure 3, we observe that, on average, the prediction quality increases as more features are incorporated into the learned models, which is confirmed for both accuracy and $F1$. The prediction quality, however, is shown to stabilize as the number of features reaches a certain threshold (e.g., 40 features or more), suggesting that there is a minimum number of features that are sufficient to learn a model without degrading the performance. Another observation is that the number of learned features seems to affect the model performance significantly, but the actual features used do not seem to make a major impact on the prediction quality. This can be validated as the mean performance, computed at each iteration using 10 randomly sampled feature subsets, is shown to improve with adding more features, which suggests that the collective performance gain of different features is more noticeable than the gain by each feature individually. It is worth mentioning that the technical indicators, representing features in this study, are used to capture various aspects of stocks such as trend, volume, and momentum, so as a collective group, we assume that they provide a better indication of a stock price movement, which can be supported by this analysis.

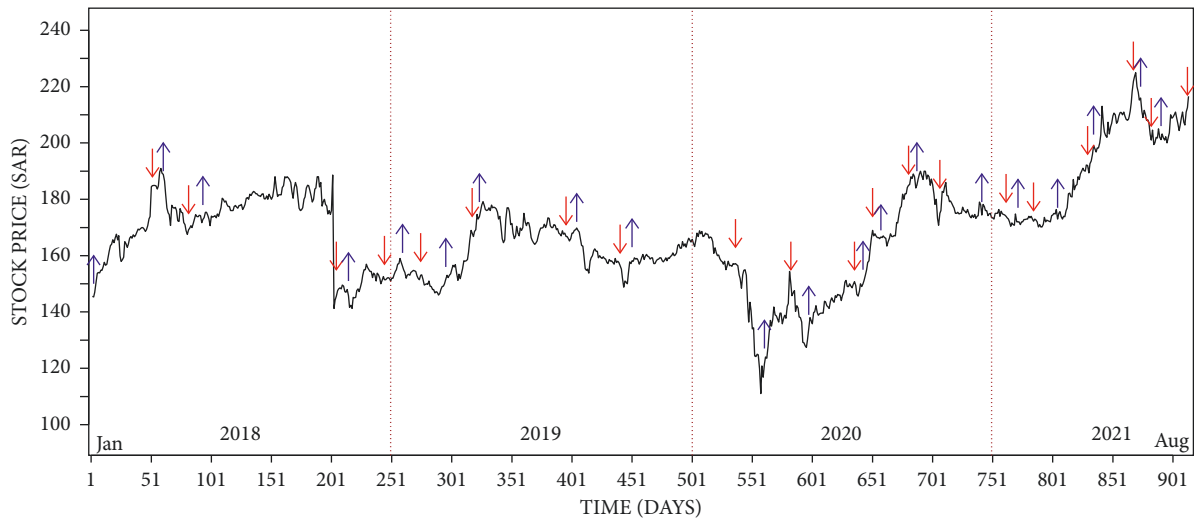
4.4. Results and Analysis of Simulating Automated Trading. To evaluate our framework realistically, we simulate the creation of a portfolio that invests in a diverse set of symbols consisting of 10 companies listed in Tadawul. These companies are shown in Table 6. We use a total capital of 100,000 Saudi Riyals (SAR) and divide it equally among these symbols (i.e., each company has a dedicated 10,000 SAR for investment). It should be noted that our selection of symbols is somehow arbitrary because we focused on diversifying our portfolio and selecting companies that were generating positive revenues for the period preceding our testing

TABLE 6: List of the stock market symbols that are considered by our investment portfolio.

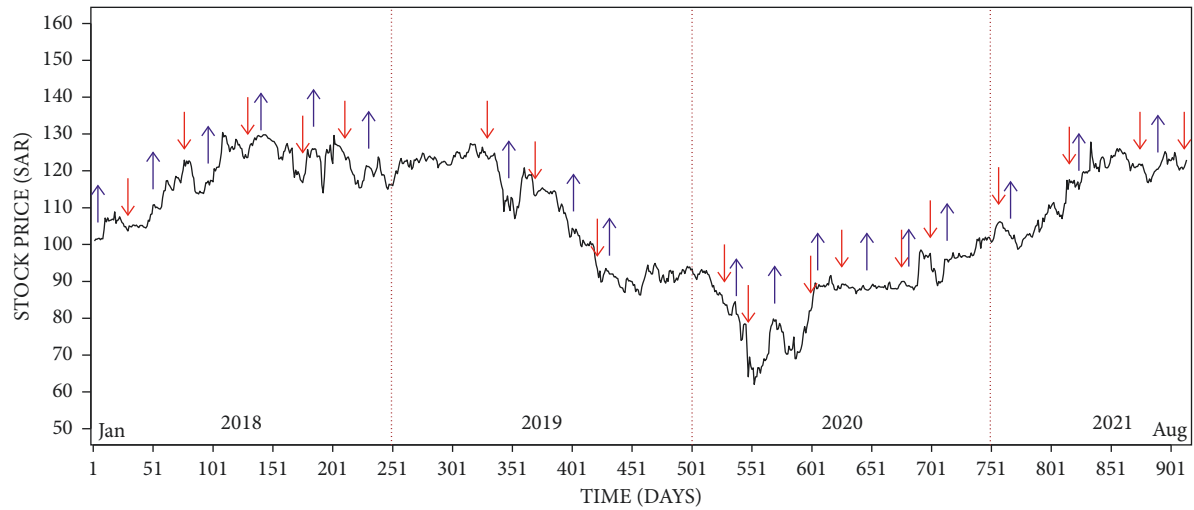
Symbol	Ticker ID	Category
Jarir	4190	Retailing
Sabic	2010	Materials industry
Alrajhi	1120	Banking
STC	7010	Telecommunication
Albilad	1140	Banking
Alinma	1150	Banking
Aldrees	4200	Energy
Alothaim	4001	Food and staples
Mobily	7020	Telecommunication
Tkaful	8230	Insurance

period. Nevertheless, to avoid any bias toward a certain industry or a category of stocks, we selected our companies from the major sectors of the market like the material industry, energy, banking, telecommunication, food, retailing, and insurance. Also, we considered companies representing three different market capitalizations: large-cap (10 billion SAR or more) such as Alrajhi, Sabic, STC, and Alinma; mid-cap (1 billion SAR to 10 billion SAR) like Albilad, Jarir, and Mobily; small-capital (less than 1 billion SAR) like Alothaim, Aldrees, and Takaful. Our future work, however, will consider more comprehensive ways to develop better techniques (potentially automated and data-driven) for selecting the most suitable stock symbols for investment.

We use a model learned with RF (owing to its efficiency and effectiveness, as discussed previously) to predict the stock price movements (up and down) for every day while using the proposed automated decision-making strategy (described in Section 3.3) to conduct various trading actions (Section 4.1 provides more insights into how we set our trading tool). This process was applied to all 10 symbols selected for our portfolio (Figure 4 illustrates a step-by-step



(a)



(b)

FIGURE 4: A step-by-step illustration of applying our automated trading process to two symbols, Jarir (a) and Sabic (b), for the period from January 2018 to August 2021. A blue arrow indicates a decision to purchase a stock at a given time (performed once or followed by several purchases as the stock price is moving up). Red arrows indicate decisions to exit trades by selling all purchased stock shares.

TABLE 7: The performance of our method (rounded) for the period from January 2018 to August 2021 using seven different portfolio settings. The performance is compared to the buy-hold strategy and the returns by the market index (TASI). S/L and T/P represent the stop-loss and take-profit points, respectively.

Method	Portfolio settings		Returns				Total (%)	#Trades	Win rate (%)
	S/L (%)	T/P (%)	2018 (%)	2019 (%)	2020 (%)	2021 (%)			
1	5	10	10	12	20	32	74	289	51
2	5	15	11	12	15	30	65	228	46
3	7.5	10	16	16	17	37	86	221	63
4	7.5	15	15	14	15	31	75	173	59
5	10	10	16	11	16	38	81	192	69
6	10	15	20	8	15	27	70	155	65
7	Optimal		25	23	26	45	119	265	66
	Buy-hold		24	14	12	26	76		N/A
	Market index (TASI)		8	7	4	30	49		N/A

TABLE 8: The returns per symbol in our portfolio are reported for the entire trading period. Aldrees resulted in the highest returns (246%), whereas Alothaim has the lowest (34%).

Stock symbol	2018 (%)	2019 (%)	2020 (%)	2021 (%)	Total (%)
Jarir	24	7	14	25	70
Sabic	8	0	6	21	35
Alrajhi	25	11	19	77	132
STC	17	-1	1	27	45
Albilad	22	13	4	62	101
Alinma	7	9	0	43	59
Aldrees	31	89	54	72	246
Alothaim	19	5	10	0	34
Mobily	-13	21	35	13	56
Tkaful	21	1	30	29	81
Portfolio total	16	16	17	37	86

TABLE 9: A side-by-side comparison, reported by total returns, between our method (best and optimal settings of S/L and T/P) and the buy-hold strategy for each stock symbol.

Stock symbol	Our method (best) (%)	Our method (optimal) (%)	Buy-Hold (%)
Jarir	70	107	73
Sabic	35	50	23
Alrajhi	132	161	125
STC	45	84	72
Albilad	101	134	84
Alinma	59	75	45
Aldrees	246	246	117
Alothaim	34	90	76
Mobily	56	145	87
Tkaful	81	106	58
Total	86	119	76

application of this process on two symbols Jarir and Sabic). The results of our experiments are presented in Table 7, showing the portfolio's performance after deducting the brokerage fees, as explained in Section 4.1, for the period starting in 2018 and ending in August 2021. The results are reported for different settings of stop-loss (S/L) and take-profit (T/P) points. In addition, a case of applying optimal values for these two parameters for each symbol is included (i.e., by sweeping S/L and T/P's values, explained in Section 4.1, for a given company's symbol) to show the upper bound of the potential returns when S/L and T/P are learned from previous data and optimized for each symbol. The performance of our portfolio is also compared with a buy-hold strategy as well as the returns by the main index of the Saudi stock market (TASI), as shown in Table 7.

Additionally, we report, in Table 8 the performance of our portfolio in terms of investment returns per company for the entire evaluation period by considering the results of one portfolio setting, no 3, which resulted in the best returns (excluding the optimal settings). Moreover, in Table 9, we show a side-by-side comparison between our method (both best and optimal settings) and the buy-hold strategy for each symbol using the total returns as a performance metric.

The results in Tables 7 and 8 show that our trading framework is very promising, as it can lead to high investment returns ranging from 65% to 86% for a period of

approximately 3.7 years and has the potential of reaching up to 119% for the same period. Also, the framework is shown to result in positive investment returns for all symbols in our portfolio, as Table 8 indicates. In comparison to the main Saudi market index (TASI), the performance of our framework is shown to exceed the performance of the market, i.e., it can achieve returns twice or more than the market returns for the same period. The results in Table 7 also show that most of the trades executed by the framework are profitable, leading to winning ratios that are higher than the loss ratios for almost all portfolio settings (the maximum is 69% for setting 5). Even when having loss ratios that overtake win ratios (as in setting 2 in Table 7), we see that our framework resulted in positive returns, suggesting that the resulting wins are large, while losses are relatively small.

By comparing our framework with the buy-hold strategy, reported in Tables 7 and 9, we can indicate the superiority of our method over the buy-hold strategy, especially when the stop-loss and take-profit points used by our method are fine-tuned and optimized using previous data. In fact, our results suggest that there is a significant impact of the stop-loss (S/L) and take-profit (T/P) points, which is similar to what is discussed in prior work [28–30]. Selecting suitable values, by analyzing market volatility, exploring technical indicators, and optimizing the parameters on

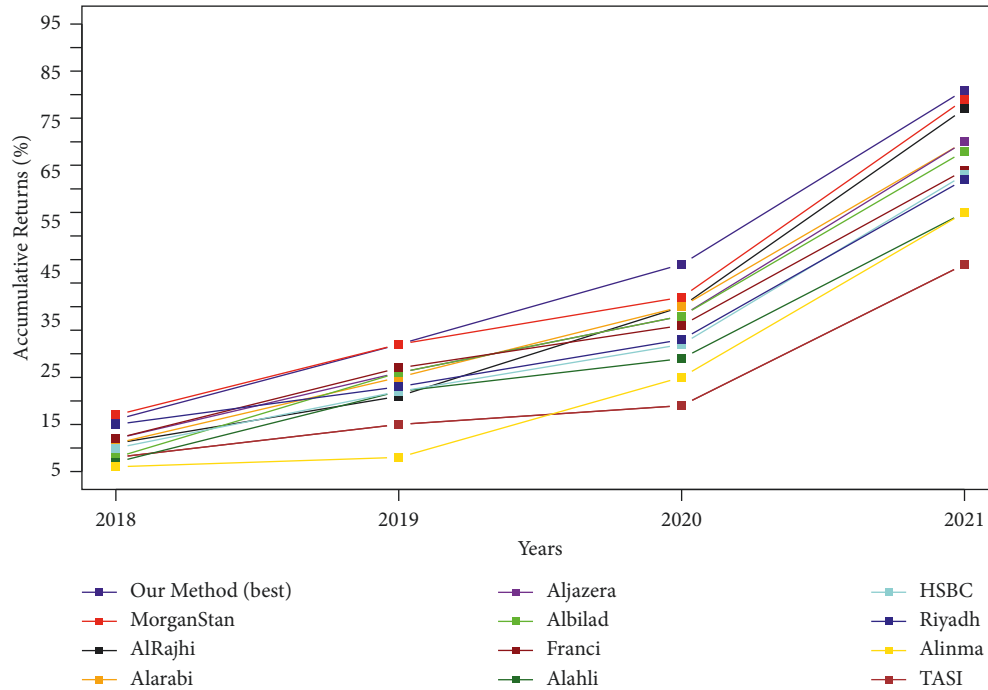


FIGURE 5: Performance comparison (accumulative returns) of our best portfolio with the top hedge funds managed by investment banks in Saudi Arabia in addition to the stock market index (TASI) for the period from January 2018 to August 2021.

TABLE 10: The performance of hedge funds (rounded) managed by the top investment banks in Saudi Arabia for the period from January 2018 to August 2021.

Hedge fund	2018 (%)	2019 (%)	2020 (%)	2021 (%)	Total (%)
Morgan Stanley SA	17	15	10	42	84
Alrajhi capital	11	10	19	42	82
Alarabi national	11	14	15	35	75
Aljazera capital	12	14	12	37	75
Albilad capital	8	18	12	34	72
Saudi franci	12	15	9	33	69
HSBC SA	10	12	10	36	68
Riyadh capital	15	8	10	34	67
Alinma investment	6	2	17	35	60
Alahli capital	7	15	7	31	60
Median	11	14	10	35	70
Our method (best)	16	16	17	37	86

training data, is expected to have a major impact on the framework performance, which is supported by our study.

We further compare our trading framework to a sample of hedge funds investing mainly in the Saudi stock market and managed by top investment banks in Saudi Arabia (as shown in Figure 5 and Table 10). It should be noted that these funds rely on the expertise of financial advisors and professional traders in those banks to make trading decisions and do not apply any automated trading. The results show that in terms of total return, our framework (especially with portfolio setting no 3) outperformed all investment funds for the same period; also, the improvement over these funds is statistically significant, except for the top two hedge funds. Moreover, as Table 10 and Figure 5 show, the performance of the framework is consistent over

the four years, generating returns higher than the median of all funds for each year.

Overall, the analysis performed in this section shows the high potential of automated trading (known as robot expert) for automating the task of investing in the financial market and suggests that it can produce investment returns that are as good as those of human professional traders. It also shows the major impact of incorporating risk management techniques to leverage the performance of automated trading, especially when relying on prediction models that are shown to be imperfect (i.e., as in our case, the accuracy of RF reaches 73.5%). One limitation of our analysis, however, is that it considered a period exemplifying a growth in Saudi Arabia's economy coupled with a rise in the stock market (i.e., the market was trending up for most of the examined

period). This potentially can lead to overestimating the performance of our method as no economic recession or market decline was observed during that period. Therefore, our future work will attempt to address this limitation by exploring several directions; for instance, examining a set of stock market symbols that are trending down and measuring the effectiveness of our method while being used for automated trading in the stock market.

5. Conclusions

This study addresses the problem of automating the task of investing and trading in stock markets through developing a framework that acts as an advisory robot for making trading decisions (i.e., buying, holding, and selling companies' shares). The findings from our experiments suggest that incorporating machine learning models, as well as portfolio risk management principles, can be significantly effective in automating this task while generating high investment returns that are comparable to the top hedge funds managed by professional financial advisors. In the future, our work will focus on enhancing the performance of the proposed framework. We would like to expand on the features used by our framework in learning a prediction model for stock price movements by including a set of features that reflect on the market news context (e.g., companies' announcements, local and global market news). Moreover, another option is to explore applying more comprehensive and automated approaches to select and choose stock symbols for investments. All these enhancements could potentially increase the framework performance and lead to improvements in its trading strategy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the Researchers Supporting Project (No. RSP2022R449), King Saud University, Riyadh, Saudi Arabia, for supporting this work.

References

- [1] N. M. Masoud, "The impact of stock market performance upon economic growth," *International Journal of Economics and Financial Issues*, vol. 3, no. 4, pp. 546–565, 2013.
- [2] S. Bahadur and S. Neupane, "Stock market and economic development: a causality test," *The Journal of Nepalese Business Studies*, vol. 3, no. 1, p. 647, 2006.
- [3] R. Wolski, "Risk and return in the real estate, bond and stock markets," *Real Estate Management and Valuation*, vol. 25, pp. 15–22, 2017.
- [4] G. D. Jud, S. E. Roulac, and D. T. Winkler, "Evaluating the risk of housing investment," *The Appraisal Journal*, vol. 73, no. 4, p. 858, 2005.
- [5] J. R. Webb and J. H. Rubens, "The effect of unbundling asset returns on restricted mixed-asset portfolios," *alternative Ideas in real estate investment*, vol. 2, pp. 83–97, 1995.
- [6] G. Kumar, S. Jain, and U. P. Singh, "Stock market forecasting using computational intelligence: a survey," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1069–1101, 2021.
- [7] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: a review and taxonomy of prediction techniques," *International Journal of Financial Studies*, vol. 7, no. 2, p. 26, 2019.
- [8] B. Zaini, R. Mansor, and N. Yusof, "Classify stock market movement based on technical analysis indicators using logistic regression," *ARBMS*, vol. 14, no. 1, pp. 35–41, 2019.
- [9] E. Güreşen and G. Kayakutlu, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications*, vol. 38, pp. 10389–10397, 2011.
- [10] W. C. Chiang, D. Enke, and T. Wu, "An adaptive stock index trading decision support system," *Expert Systems with Applications*, vol. 59, pp. 195–207, 2016.
- [11] X. Pang, Y. Zhou, and P. Wang, "An innovative neural network approach for stock market prediction," *The Journal of Supercomputing*, vol. 76, pp. 2098–2118, 2020.
- [12] M. Usmani, S. H. Adil, and K. Raza, *Stock Market Prediction Using Machine Learning Techniques in Proc of ICCOINS*, Malaysia, Kuala Lumpur, 2016.
- [13] M. Usmani, V. K. Menon, and S. Ka, "NSE stock market prediction using deep-learning models," *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018.
- [14] M. Agrawal, A. Khan, and P. Shukla, "Stock price prediction using technical indicators: a predictive model using optimal deep learning," *International Journal of Recent Technology and Engineering*, vol. 2, no. 2, p. 2457, 2019.
- [15] F. A. Alturki and A. Aldughaiyem, "Trading Saudi stock market shares using multivariate recurrent neural network with a long short-term memory layer," *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 9, p. 34, 2020.
- [16] Y. Alsubaie, K. E. Hindi, and H. Alsalman, "Cost-sensitive prediction of stock price direction: selection of technical indicators," *IEEE Access*, vol. 7, pp. 146876–146892, 2019.
- [17] EReference data, *Saudi Stock Exchange Historical Data*, <https://www.saudiexchange.sa/wps/portal/tadawul/knowledg-e-center/about/ereference-data>, 2022.
- [18] O. B. Sezer and A. M. Ozbayoglu, "Algorithmic financial trading with deep convolutional neural networks: time series to image conversion approach," *Applied Soft Computing*, vol. 70, pp. 525–538, 2018.
- [19] M. Alsulmi, "Reducing manual effort to label stock market data by applying a metaheuristic search: a case study from the Saudi stock market," *IEEE Access*, vol. 9, pp. 110493–110504, 2021.
- [20] R. W. Colby, "The encyclopedia of technical market indicators," *McGraw-Hill*, vol. 674, p. 3234, 2003.
- [21] J. Chen, "Technical indicator," *Invest*, 2022, <https://www.investopedia.com/terms/t/technicalindicator.asp>.
- [22] S. Patro and K. K. Sahu, "Normalization: a preprocessing stage," *International Advanced Research Journal in Science*, vol. 2, no. 3, p. 7868, 2015.
- [23] B. E. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc of the fifth annual*

- workshop on Computational Learning Theory* vol. 213, p. 234, ACM, 1992.
- [24] V. N. Vapnik, "The nature of statistical learning theory," *SpringerPlus*, vol. 13, p. 565, 1995.
 - [25] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
 - [26] E. Grossi and M. Buscema, "Introduction to artificial neural networks," *European Journal of Gastroenterology and Hepatology*, vol. 19, pp. 1046–1054, 2007.
 - [27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.
 - [28] C. Păuna, "Capital and risk management for automated trading systems," *Proc of the 17th International Conference on Informatics in Economy*, pp. 183–188, 2018.
 - [29] K. M. Kaminski and A. W. Lo, "When do stop-loss rules stop losses?" *Journal of Financial Markets*, vol. 18, pp. 234–254, 2004.
 - [30] C. H. Chen, Y. H. Chen, and M. E. Wu, "A sophisticated optimization algorithm for obtaining a group trading strategy portfolio and its stop-loss and take-profit points," *Proc of IEEE SMC*, pp. 3417–3420, 2018.
 - [31] M. Hall, F. Eibe, and G. Holmes, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
 - [32] S. Lang, F. Bravo-Marquez, and C. Beckham, "Wekadeeplearning4j: a deep learning package for weka based on deeplearning4j," *Knowledge-Based Systems*, vol. 197, pp. 48–50, 2019.
 - [33] G. Shmueli and K. C. Lichtendahl, "Practical time series forecasting with r: a hands-on guide," *Axelrod schnell publishers*, vol. 34, p. 98, 2016.