IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 15, 2021, accepted November 29, 2021, date of publication December 8, 2021,
date of current version December 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3134138

Impact of Hyperparameter Tuning on Machine Learning Models in Stock Price Forecasting

KAZI EKRAMUL HOQUE AND HAMOUD ALJAMAAN
Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
Corresponding author: Kazi Ekramul Hoque (g201803260@kfupm.edu.sa)

This work was supported by the King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia.

ABSTRACT Stock price forecasting has been reported as a challenging task in the scientific and financial communities due to stock prices' nonlinear and dynamic nature. Machine learning models exhibit capabilities that allow them to handle nonlinear data and be candidate tools for stock price forecasting. In this study, an empirical evaluation of eight conventional machine learning models' is conducted to forecast the stock price of eleven companies belonging to the Saudi Stock Exchange. Moreover, the optimal configuration of hyperparameters in each machine learning model is identified. Forecasting performance is evaluated by two well-known error metrics: Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). Wilcoxson effect size is utilized to determine the impact of hyperparameter tuning by comparing tuned and un-tuned machine learning models' forecasting performance. Empirical results indicate there are varying impacts of hyperparameter tuning of machine learning models in forecasting stock price. After tuning the hyperparameters, Support Vector Regression outperforms other forecasting models with a significant statistical difference. In contrast, Kernel Ridge Regression shows noteworthy forecasting performance without hyperparameter tuning with respect to other un-tuned forecasting models. However, Decision Tree and K-Nearest Neighbour are the poor-performing models which demonstrate inadequate forecasting performance even after hyperparameter tuning.

INDEX TERMS Gaussian process regression, hyperparameter tuning, kernel ridge regression, LASSO, machine learning, stock price forecasting, support vector regression, time series analysis.

I. INTRODUCTION

Stock markets are the most attractive place to invest in the financial markets due to its outstanding revenue opportunities, which also encompasses a tremendous risk factor of losing colossal capital. This phenomenon is prompted by the volatile nature of daily stocks prices, which depends on abundant grounds such as: a company's reputation and financial performance, global economic condition, political stability, foreign policy, etc. [1]. As for investors in stock markets, forecasting stock price is a desirable and crucial task, yet it is the most challenging task due to the frequently changing stocks prices nature. As a result, forecasting stocks prices with reasonable accuracy can enable investors to boost their rewards and minimize their losses.

In Saudi Arabia, the Capital Market Authority (CMA) regulates the Saudi Arabian Capital Market by enforcing Capital Market Laws from its establishment [9]. CMA was formed to safeguard investors and residence from fraudulent and illegal stock trading. Currently, the Saudi Stock Exchange is commonly known as Tadawul in the Kingdom, regulated by CMA [10]. Tadawul comprises eleven different sectors such as: Energy, Materials, Information Technology, Financial organization, etc. [11].

Time series data is formed by a set of observations with a timestamp which is the best form to visualize and analyze stock markets. Time series analysis involves the utilization of historical data to forecast the future. System analysts often use these forecasting models to manage and plan future events to minimize risks, maximize resource utilization, and increase profit. These forecasting techniques are commonly used to approximate stock prices and future trend direction. Nowadays, machine learning models are becoming more popular in stock price forecasting due to their time-series forecasting capabilities [12].

In general, machine learning models consist of two parameters: model parameters and hyperparameters. The model parameters are learned by the model during the training

The associate editor coordinating the review of this manuscript and approving it for publication was Shunfeng Cheng.

phase. In contrast, machine learning model hyperparameters must be configured at the beginning of training, and during training, it may change (early stopping and learning rate decay) or remain constant. In this experiment, after configuring, hyperparameters remain constant throughout the training phase. As a result, to develop a robust machine learning model, searching for the best hyperparameter setting may turn out to be crucial [13], [14]. Moreover, the default hyperparameter configuration of a machine learning model does not guarantee the best performance. Additionally, the hyperparameter's optimal values in a machine learning model sometimes depend on the dataset and problem domain [15]. Thus, a range of hyperparameter values is explored in developing an ideal machine learning model. This procedure of finding the best hyperparameter setting for a machine learning model is often known as hyperparameter tuning [16]. Manually searching for the best hyperparameter is still widespread in research; however, it requires an adequate understanding of the hyperparameter setting of corresponding machine learning models [17]. It is sometimes impossible to use manual tuning due to numerous hyperparameters, time inefficient model evaluation, and the complexity of specific problem domains. Accordingly, researchers proposed several hyperparameter optimization techniques to automate or partially automate the hyperparameter tuning process [18].

In this study, we aim to illustrate the forecasting capabilities of eight machine learning models (Decision Tree (DT), Support Vector Regression (SVR), K Nearest Neighbour (KNN), Gaussian Process Regression (GPR), Stochastic Gradient Descent (SGD), Partial Least Squares Regression (PLS), Kernel Ridge Regression (KRR), Least Absolute Shrinkage And Selection Operator (LASSO)). In addition, the grid search hyperparameter tuning method is implemented to identify the best hyperparameter configuration for each machine learning model. Finally, the impact of tuning hyperparameters of these models is investigated in the context of the Saudi Stock Exchange.

The rest of this paper is organized as follows: Section II reviews the existing literature in stock price forecasting. Section III briefly discusses the principle ideas of machine learning algorithms that are utilized in our empirical investigation. Section IV reports the overall configuration of our empirical study. Section V illustrates the results of this experiment with suitable analysis. Section VI points out the possible threats to validity in our experiment. Finally, in section VII, we conclude our paper with a direction to future research in this domain.

II. LITERATURE REVIEW

In this section we review the related work conducted in stock prices forecasting using machine learning models.

Forecasting models using time series analysis came into light with the widely used traditional statistical models named Autoregressive Integrated Moving Average (ARIMA) [19]. Moreover, ARIMA and its variants gained considerable interest due to its extensive usage in various forecasting activities, especially in stock price forecasting [20]. For forecasting stocks price, previous studies utilized traditional models such as: Linear Regression [21], Autoregressive Integrated Moving Average (ARIMA) [22], Moving Average Convergence / Divergence (MACD), and Relative Strength Index (RSI) [23]. These statistical models are usually developed with linear function structure, making them suffer from performance issues in real data with noise and non-linearity characteristics [24]. On the other hand, machine learning models showed promising performance in detecting underlying nonlinear relationships and random assumptions. In addition, machine learning models illustrated an extraordinary capability in handling noisy real-world data [25].

Mehtab *et al.* [2] used various deep learning techniques to evaluate stock prices forecasting performance based on execution time and RMSE value with two different sliding window sizes. Their study suggests that CNN-based models performed better than LSTM based models in forecasting stock price in the context of India's National Stock Exchange (NSE). Azlan *et al.* [3] conducted a time series analysis using the clonal selection algorithm and found almost similar forecasting performance as ARIMA models on yahoo stock price. In time series forecasting, LSTM models illustrated superior forecasting performance with a long-term confident band [4]. Li and Bastos [26] conducted a systematic literature review that reported LSTM is the most widely used deep learning technique on stock price forecasting. In addition, the daily stock price timeframe was used in most of their reviewed studies. Henrique *et al.* [7] conducted an empirical study using support vector regression (SVR) to forecast the daily stock prices of selected companies from China, Brazil, and the USA. The study reports that using the linear kernel in SVR helps to minimize the forecasting error. Moreover, according to their research, SVR showed superior forecasting accuracy compared with the random walk-based model. Chou and Nguyen [6] designed a time series forecasting system using Least squares support vector regression (LSSVR) with sliding window meta-heuristic optimization to forecast the stock price. Moreover, they developed an application with a simple graphical user interface from their proposed method to make stock price forecasting easy. Olatunji *et al.* [8] proposed an artificial neural network model and evaluated the models using three Saudi Stock Exchange companies such as STC, SABIC, and Al Rajhi Bank. They used the previous five days' stock price to forecast the next day's stock price. They reported that their proposed model achieved significantly low root mean square error. Shrivastav and Kumar [27] conducted an empirical study on stock price forecasting using SVR and ARIMA model. The results illustrate the superiority of stock price forecasting performance of the SVR model compared with the ARIMA model.

Table 1 presents a summary of the surveyed related studies that had conducted a univariate time series analysis using different machine learning techniques in forecasting stock prices. In most studies, researchers used deep neural network models like LSTM, ANN, etc., whereas only a few

TABLE 1. Comparison Among Literature.

Ref (Year)	Objective	ML models	Evaluation metric	Sliding window	Univariate Variable	Train-Test	Datasets
[2] 2020	Stock price forecast	LSTM, CNN	RMSE	5 and 10	Open price	50-50	NSE, india
[3] 2020	Stock price forecast	Clonal Selection	RMSE, MAPE	2	Close price	70-30 80-20 90-10	SP500
[4] 2020	Stock price forecast	LSTM	RMSE	5	Close price	64-36	S&P 500, Google stock price
[5] 2019	Stock price forecast	LSTM	MAE	20	Close price	80-20	Apple's stock
[6] 2018	Stock price forecast	LSSVR	RMSE, MAPE, MAE	2	Close price	N/A	Taiwan
[7] 2018	Stock price forecast	SVR	RMSE, MAPE	7	Close price	70-30	Brazilian, American, and Chinese stocks
[8] 2013	Stock price forecast	ANN	RMSE, MAPE, R	5	Close price	70-30	SABIC, STC, Al Rajhi Bank

conventional models were investigated [26]. In this paper, we aim to fill this gap by empirically investigating the forecasting capabilities of several conventional machine learning models and performing a comprehensive empirical study within the context of the Saudi stock market (Tadawul).

III. BACKGROUND

A range of machine learning models is used in the literature to develop forecasting models nowadays. Moreover, we can conclude from the literature that the two most dominant core machine learning models are conventional models and neural networks. Conventional machine learning encompasses any core algorithmic framework used to find a solution by the use of data. These systems can learn automatically from data, with subject matter experts conducting preprocessing and selecting features to feed the algorithm. Usually, these trained machine learning models need all inputs to be structured data, such as numbers. Neural network algorithms are capable of learning key aspects from underlying data and determining which features to focus on without explicit expert identification. However, we observe that only a few conventional machine models are used to forecast stock prices from the literature. Moreover, these models do not assure a higher forecasting accuracy. However, we still investigate these models due to their straightforward implementation and ease of explaining non-technical individuals. In our study, we focus on evaluating various conventional machine models rather than neural network models. Here we briefly discuss the machine learning models used in this study:

A. DECISION TREE (DT)

In machine learning, the concept of employing a decision tree was first introduced by Quinlan, which was referred to as the Induction of Decision tree, commonly known as ID3 [28]. In ID3, it searches for the decision tree that can correctly classify by generating all possible combinations of decision stumps. In classification problems, the decision tree aims to maximize the Information gain, whereas, in regression problems, it minimizes standard deviation or mean square error. In this study, we have used decision tree regression to develop one of the proposed forecasting models.

B. SUPPORT VECTOR REGRESSION (SVR)

To solve binary classification problems, Vladimir of the AT&T Bell Laboratories first introduced Support Vector Machine (SVM) that formulated it as a convex optimization problem [29]. For solving regression problems, Support Vector Regression (SVR) was remodeled by specifying the ε intensive region around the function, which can approximate the continue values with reasonable model complexity. Moreover, SVR formulates regression problems as an optimization problem that seeks the tightest path possible around the surface by reducing the prediction error [30]. This condition can be represented as an equation as follows [31]:

$$minimize\ \frac{1}{2}\|w\|^2 \tag{1}$$

where the magnitude of the vector containing real continous values is denoted as $\|w\|$, which is being predicted.

C. K NEAREST NEIGHBOUR (KNN)

The K-nearest neighbour algorithm [32] is one of the initial machine learning algorithms. However, It is frequently used for classification and regression due to its clarity and configurability. KNN [33] is usually referred to as a lazy learning model since it does not develop any model or function using the training set. Instead, for each test set element, it finds similar k nearest records from the training set. Then, the prediction is performed by majority voting among that k nearest records. In KNN regression analysis [33], to forecast the values in the test set, the k most past patterns were identified and combined from the training set.

D. GAUSSIAN PROCESS REGRESSION (GPR)

To solve nonlinear regression problems, Gaussian Process Regression (GPR) follows a non-parametric and probabilistic approach. According to GPR, The measurements of the output variable y are produced in the following way [34]:

$$y = f(x) + \varepsilon \tag{2}$$

$$f(x) \sim GP\left(m(x), \mathrm{k}\left(x, x'\right)\right) \tag{3}$$

$$\varepsilon \sim \left(0, \sigma^2\right) \tag{4}$$

where x is the input variables, f is the function of Gaussian Process Prior, m is the mean function, k is the covariance function, and ε is Gaussian noise with variance σ^2. Gaussian processes are entirely characterized by their mean and covariance functions, which encode our conclusions about the process. After selecting the mean and covariance functions, the output variable is predicted in the form of predictive Gaussian distribution.

E. LINEAR REGRESSION WITH STOCHASTIC GRADIENT DESCENT (SGD)

At Present, the SGD plays an essential role in different activities of machine learning [35]. It is an optimization technique that uses a noisy gradient combined with a reduced step size. Usually, The slope of a function is referred to as a gradient. It quantifies the magnitude of change in an input variable, resulting in the change of the output variable. Gradient descent is a convex function that produces the partial derivative of a series of input parameters. Finally, SGD is the technique that extends gradient descent by optimizing the process in a stochastic manner [36]. In our experiment, we utilized SGD as a basic stochastic gradient descent learning method for fitting linear regression models. It supports a range of learning rates and penalty which are presented in table 4. SGD has previously been used to solve classification problems such as detecting code smells [37] and categorizing text documents [38], as well as regression tasks such as biomass prediction [39] and healthcare analysis [40].

F. PARTIAL LEAST SQUARES REGRESSION (PLS)

Partial Least Squares regression encompasses the renowned principle of partial correlation [41]; it aims to forecast y using a number of input variables x. The predictors of a PLS regression model are developed by the linear combination of input variables x. These predictors are generally referred to as PLS components or latent vectors, which maximize the correlation between x and y. Finally, PLS regression analysis can be utilized for both linear and nonlinear data forecasting [42].

G. KERNEL RIDGE REGRESSION (KRR)

In 2000, Cristianini and Shawe-Taylor [43] introduced the expression ''Kernel Ridge Regression'' to refer to a specialized form of Support Vector Regression which was a variant of the previous ''ridge regression in dual variables'' [44]. The main difference between SVR and KRR is in the selection of loss function. Additionally, KRR enables model fitting to be performed more quickly than SVR. On the other hand, SVR performs forecasting more rapidly than KRR.

H. LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR (LASSO)

For parameter estimation in regression problems, the lasso introduced by Tibshirani [45] has become a widespread substitution to the simple least-squares method. Its success is attributed to the main function of this method: Least Absolute Shrinkage, which shrunk the vector of regression coefficients, with the probability of setting certain coefficients to zero. Thus, it results in a continuous estimation and variable selection.

IV. EMPIRICAL STUDY

This section outlines our empirical research objective and formulates research questions to achieve our research objective. Then, we explain the dataset used with a descriptive analysis of the underlying dataset distribution. In addition, we briefly describe different forecasting error measures used in this empirical study to measure forecasting errors of the machine learning models. Finally, we explain the performed statistical analysis to examine to which extent the performance differences between machine learning models are significant or not. This empirical study is entirely implemented and conducted using Python.

A. GOAL

Using the GQM template [46], we lay the goal of our empirical study as follows:

- **Evaluate**: The forecasting capabilities and the impact of hyperparameter tuning of conventional machine learning models
- **Purpose**: Forecasting the stocks prices
- **Respect**: Magnitude of relative error, mean absolute percentage error, and root mean square error
- **Perspective**: Researcher, Stock Investor, Financial Data Analyst
- **Context**: 11 stock prices in the Saudi Stock Market (Tadawul)

To achieve our goal, we have formulated the following research questions to guide our empirical investigation:

RQ 1. What are the capabilities of conventional machine learning models in forecasting stocks prices in the Saudi Stock Market?

RQ 2. What are hyperparameter tuned conventional machine learning models' capabilities in forecasting stock price in the Saudi Stock Market?

RQ 3. To which extent does hyperparameter tuning impact the performance of conventional machine learning models in forecasting stock price in the Saudi Stock Market?

B. SAUDI STOCK MARKET DATASETS

In this research, we utilize the Saudi Stock Market (Tadawul) data to conduct our empirical study. There are eleven sectors in Tadawul, where each sector consists of a large number of companies.

1) DATASET DESCRIPTION

In this study, we have selected one stock company from each sector to interpret our findings well. The list of the selected companies with relevant details are given in Table 2.

We collected the data from www.investing.com in January 2021, which provides the Saudi Stock Exchange's daily stock price. Moreover, each dataset contains each company's daily stock data from the listing date to 31st December 2020. Each

TABLE 2. Datasets Description.

Company Name	Trading Name	Sector	Listing Date to 31/12/2020	Data Points
Saudi Arabian Oil Co.	SAUDI ARAMCO	Energy	12/12/2019	265
Saudi Arabian Mining Co	MAADEN	Materials	29/07/2008	3097
Saudi Public Transport Co.	SAPTCO	Industrials	01/01/2002	4950
Saudi Industrial Development Co.	SIDC	Consumer Discretionary	01/01/2002	4950
Almarai Co.	ALMARAI	Consumer Staples	18/08/2005	3869
Saudi Chemical Co.	CHEMICAL	Health Care	01/01/2002	4950
Alinma Bank	ALINMA	Financials	04/06/2008	3136
Arab Sea Information System Co.	ARAB SEA	Information Technology	27/02/2017	791
Saudi Telecom Co.	STC	Communication Services	26/01/2003	4632
National Gas and Industrialization Co.	GASCO	Utilities	01/01/2002	4950
Taiba Investments Co.	TAIBA	Real Estate	01/01/2002	4950

TABLE 3. Dataset Characteristics.

Company	Price (SAR)			
	Min	Max	Mean	Std. Dev.
SAUDI ARAMCO	27.8	38	33.69	2.04
MAADEN	9.42	59.5	33.64	12.09
SAPTCO	5.44	70.4	17.37	9.93
SIDC	3.35	135	17.09	14.77
ALMARAI	9.74	76	37.60	16.15
CHEMICAL	5.51	75.87	29.42	11.08
ALINMA	6.64	21	12.24	3.55
ARAB SEA	9.00	101.6	24.65	22.46
STC	28.88	186	70.93	28.47
GASCO	15	171	33.26	21.06
TAIBA	6.63	85.71	27.40	13.16

of these stock data consists daily closing price of that stock. In this study, We have only used the closing stock price as a univariate time series to forecast the stock price because most of the previous literature used close price as a univariate variable [3]–[6]. Additionally, the closing price is the stock's final price on a particular day, making it more valuable to predict.

2) DATASET CHARACTERISTICS

Table 3 summarizes the statistical characteristics of each stock dataset used in this analysis.

SAUDI ARAMCO and ALINMA have the lowest standard deviation, indicating that their stock prices are very similar to mean value and that their stock prices fluctuate very little. In contrast, the maximum difference between the minimum and maximum stock price, as well as the maximum standard deviation, is seen in ARAB SEA, STC, and GASCO, implying a higher degree of fluctuation. Different stocks distributions within the chosen stock dataset will assist us in validating the outcome of our empirical study.

C. DATA PREPOSSESSING

Each dataset contains seven features (open, close, high, low, volume, change), including the time stamp for each sample. We have utilized only the closing stock price as the independent variable, removing all other features except the time stamp and closing stock price. There were no missing values in the closing stock price from the listing date to 31st December 2020 in any stock datasets we selected. Then, the data is converted into time-indexed closing stock prices. Later, this data is transformed into an appropriate format for supervised learning using the sliding window technique, which is often known as a lagged variable. In this study, the sliding window of size five is implemented.

D. EXPERIMENT DESIGN

Figure 1 illustrates the overall structure of our experiment design based on the Alinma dataset, and then we replicate the same procedure for the other selected datasets.

A sliding window of size five is commonly used in forecasting stock price, mentioned in the surveyed literature [3], [4], [8]. As a result, the feature list contains stock prices of t-4, t-3, t-2, t-1, t days where t corresponds to today. Furthermore, the output list includes the stock price of t + 1 day that we will forecast using the machine learning models. After that, we split the feature and output list in 80%-20% into training and testing sets without any randomization. As a result, in this experiment, training data (the first 80% of the dataset) is historical stock price and testing data is the future value of the stock of that company. Suppose for STC dataset training set is from 26-01-2003 to 11-04-2017, and the test set is from 12-04-2017 to 31-12-2020.

In the training set, at first, we develop machine learning models with the default hyperparameter configurations by sci-kit learn [47] and evaluate our models in the test set.

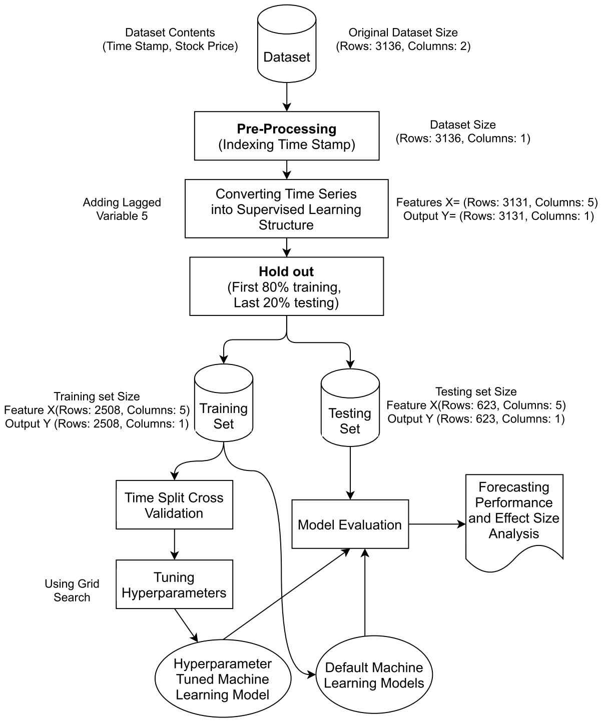


FIGURE 1. Experiment Design for Each Dataset (Alinma Dataset).

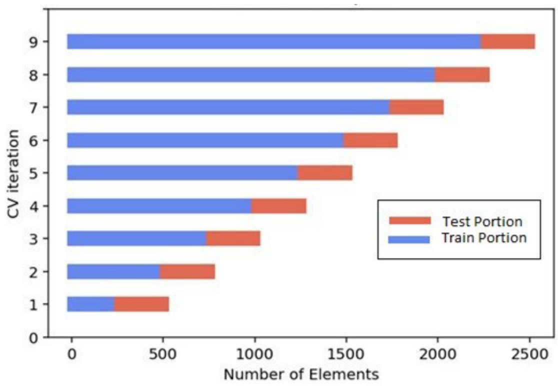


FIGURE 2. Time Split Cross-Validation fold size in Alinma Dataset.

In each dataset, we fit the default model 30 times and calculate the average error to produce more reliable and robust estimates of the machine learning models. We label these models as un-tuned machine learning models. Additionally, we create a hyperparameter space for each machine learning model. We use time split cross-validation to split the training set into nine folds which is illustrated in Figure 2.

Then, we find the best hyperparameter from the hyperparameter space over the nine folds. Moreover, we use the grid search technique, which rigorously develops machine learning models for all hyperparameters' possible combinations in the hyperparameter space and finds the best combination with the average lowest error in all nine splits.

After finding the best hyperparameter configuration, we build each machine learning model according to that arrangement. In each dataset, we also fit the tuned model 30 times and calculate the average error to gain more confidence in our machine learning models forecasting estimates. Then, we evaluate our tuned machine learning models in the test set. Finally, we analyze and compare the tuned and un-tuned models.

E. HYPERPARAMETER TUNING TECHNIQUES

The classical method for hyperparameter tuning is grid search, which is basically an exhaustive search of a given subset of possible values in hyperparameter space. A grid search algorithm is driven by some type of performance metric, which is commonly determined by cross-validation on the training set. Additionally, other tuning techniques such as random search [48], gradient search [49], and Bayesian optimization [50] have been proposed. Random Search chooses the hyperparameter combinations at random, which substitutes exhaustive search of all possible combinations in a grid search. This is applicable for both discrete and continuous/mixed value hyperparameters. For some specific learning algorithms, gradient-based optimization may be used to calculate the gradient considering hyperparameters and subsequently optimize the hyperparameters via gradient descent. In the Bayesian hyperparameter tuning method, a probabilistic model is developed from a function by translating the hyperparameter values. Bayesian optimization seeks to acquire data about this function and the position of the optimum by repeatedly assessing a potential hyperparameter configuration based on the existing model and then updating it. The optimum configuration is generated by evaluating the target on the validation set. Although there are various suggested hyperparameter tuning techniques, grid search remains state of the art for several reasons: Firstly, the implementation of grid search is straightforward and it supports parallelization. In general, it finds a better configuration of hyperparameters than manual inspection with an equal amount of time. Finally, if the dataset and hyperparameter search space is not excessively large, then grid search illustrates noteworthy performance in choosing the best hyperparameter configuration [51].

F. HYPERPARAMETER SPACE

Table 4 outlines the hyperparameter space for each machine learning model. The bolded value indicates the hyperparameter default value. Our study considers three types of hyperparameter space: integer-valued space, continuous-valued space, and string-valued space. To define the continuous-valued space, we raise and reduce the default values by the multiplication of 10. We attempt to consider all possible string values available in the string-valued space. We select accepted values around the default value with a constant interval for integer-valued space, or sometimes we set some random values to justify the range.

Each hyperparameter space is structured according to the type of that hyperparameter space. For instance, for the KNN 'algorithm' hyperparameter, we consider all the possible

TABLE 4. Hyperparameter Space.

ML Models	Hyperparameters	Hyperparameter Space
Decision Tree (DT)	Max feature	**'auto'**,' sqrt',' log2'
	Min Leaf	**1**, 2, 3, 5
	Min Split	**2**, 4, 6, 8
	Spitter	**'best'**, 'random'
Support Vector Regression (SVR)	Kernel	'linear', **'rfb'**, 'sigmoid'
	C	0.01, 0.1, **1.0**, 10, 100
	epsilon	0.001, 0.01,**0.1**, 1.0, 10
K Nearest Neighbors (KNN)	algorithm	**'auto'**, 'ball_tree', 'kd_tree', 'brute'
	n_neighbors	2, **5**, 10, 15
Gaussian Process Regression (GPR)	kernel	DotProduct, WhiteKernel, DotProduct+ WhiteKernel, **None**
Stochastic Gradient Descent (SGD)	learning_rate	'constant', 'optimal', **'invscaling'**, 'adaptive'
	penalty	**'L2'**, 'L1', 'elasticnet'
	max_iter	**1000**, 3000, 5000, 7000
Least Absolute Shrinkage and Selection Operator (LASSO)	selection	**'cyclic'**, 'random'
	alpha	0.001, 0.1, **1.0**, 10 , 100, 1000
Partial Least Squares Regression (PLS)	n_components	**2**, 3, 4, 5, 10, 50, 100
	scale	**True**, False
Kernel Ridge Regression (KRR)	alpha	0.001, 0.01, 0.1, **1**, 10, 100, 1000
	kernel	**'linear'**, 'poly', 'rbf', 'sigmoid'

algorithms KNN supports, which is a string-valued space. Then for the C hyperparameter value in SVR, a value from 0.01 to 100 is selected with ten multiplicative intervals.

G. FORECASTING EVALUATION MEASURES

In time series analysis, we measure the forecasting performance by measuring the forecasting error that represents the gap between the real value and the forecasted value. In our empirical study, the machine learning models are trained with hyperparameters tuning on the 80% training dataset. After we build the fine-tuned models, we report their forecasting error by testing the models forecasting performance on the unseen 20% test dataset, which is a common practice in the existing literature [52], [53].

This study employs two primary statistical loss functions: Root Mean Square Error (RMSE) and Mean Magnitude of Relative Error (MMRE) to evaluate our investigated models' forecasting performance. These loss functions are frequently used in recent time series forecasting studies [54], [55]. The error functions are discussed below:

1) ROOT MEAN SQUARE ERROR (RMSE)

The square of the difference between actual and forecasted values is calculated. Then, the mean value is taken from all the instances. Finally, RMSE is calculated by taking the square root of the resulting mean value. RMSE measure is one of the most preferred scale-dependent measures due to its suitability for evaluating various models built using the same dataset. Moreover, RMSE's theoretical resemblance with the statistical models made it ubiquitous in assessing forecasting model performance [56].

$$RMSE = \sqrt{\frac{1}{m}\sum_{t=1}^{m}\left(\hat{x}_t - x_t\right)^2} \tag{5}$$

2) MEAN ABSOLUTE PERCENTAGE ERROR (MAPE)

The difference between the actual and forecasted value is divided by the real value. Then, MAPE is calculated by taking the mean value of all instances and taking percentages. This measure is also commonly known as the Mean Magnitude of Relative Error (MMRE) in some literature [57]. Additionally, MAPE is the most widely used measure to evaluate forecasting models [58].

$$MAPE = \frac{1}{m}\sum_{t=1}^{m}|\frac{x_t - \hat{x}_t}{x_t}| \times 100\% \tag{6}$$

3) MAGNITUDE OF RELATIVE ERROR (MRE)

The magnitude of Relative Error (MRE) is a simple form of MMRE mentioned above without taking the mean from all the instances. We use MRE in our statistical analysis to construct and evaluate our null and alternate hypotheses. The formula of MRE is presented below:

$$MRE = |\frac{x_t - \hat{x}_t}{x_t}| \tag{7}$$

Here, m is the number of samples in our test set, x_t is the actual value from the test set and $\hat{x}_t$ is the corresponding forecasted value by the proposed machine learning models.

H. STATISTICAL ANALYSIS

In this study, we perform two different statistical tests. We first examine the significant difference in forecasting performance between conventional machine learning models using a non-parametric Wilcoxon signed-rank statistical test [59] in terms of the magnitude of relative error (MRE) at a significance level of 0.05 with Bonferroni correction [60]. Wilcoxon's main advantage is that it does not require any form of distribution in the data as it is non-parametric testing. Additionally, it is unbiased by outlier data since it does not consider the magnitude of the value. Instead, it applies signs and ranks of the value. Lastly, this test is widely used in the literature to compare forecasting models [52]. Then, the impact of hyperparameter tuning between un-tuned and tuned models is measured using the Wilcoxson effect size. It gives a clear illustration of how much forecasting performance is improved after tuning hyperparameters of each machine learning model.

1) WILCOXON SIGNED-RANK STATISTICAL TEST WITH BONFERRONI CORRECTION

In the Wilcoxon test, at first, the absolute difference between each pair of observations is measured, and then it ranks the absolute difference in ascending order. After that, every absolute difference to the corresponding rank is enclosed with a sign [61]. T_+ is represented as the summation of the rank with a plus sign, and T_- is described as the summation of the rank with a minus sign. The distribution of T can be approximated by a normal distribution if the size of the sample (n) is more than 25 where the mean,

$$\mu_T = \frac{n\left(n+1\right)}{4} \tag{8}$$

and standard deviation,

$$\sigma_T = \sqrt{\frac{n\left(n+1\right)\left(2n+1\right)}{4}} \tag{9}$$

Finally, the test statistics Z is formulated as follows:

$$Z = \frac{|T - \mu_T| - 0.5}{\sigma_T} \tag{10}$$

To reject the null hypothesis and accept the alternative hypothesis, the calculated value of Z must be greater than or equal to the critical value Z_α, where α is the level of statistical significance. The appropriate value of α depends on the domain of study.

A particular alpha value might be appropriate for a single hypothesis test, but it is not suitable for all concurrently conducted hypothesis tests on the same data. The Bonferroni correction method is used to correct alpha value when performing several dependent or independent statistical experiments. To prevent a large number of misleading type 1 error, the alpha value must be decreased to reflect the number of comparisons conducted. In Bonferroni correction, the alpha value is divided by the number of comparisons n, resulting in the corrected alpha value equal to α/n. In forecasting price literature, the tropical value of α is 0.05. But we have conducted a pairwise comparison among the forecasting performance of the machine learning models. Each dataset is subjected to a total of 28 comparisons. After the Bonferroni correction, the adjusted value of α is 0.05/28, or 0.0017857. The null and alternate hypotheses of this empirical analysis are as follows:

H_0: $MRE_X = MRE_Y$ (There is no difference in forecasting performance between the two machine learning models in terms of MRE).

H_a: $MRE_X \neq MRE_Y$ (There is a difference in forecasting performance between the two machine learning models in terms of MRE).

2) WILCOXON EFFECT SIZE TEST

The effect size test measures the magnitude of a treatment outcome. The Wilcoxon effect size [62] is measured as the simple difference between the proportion of favorable and unfavorable data of rank sums in the Wilcoxon signed-rank test.

$$r = f - u \tag{11}$$

Here, r is the effect size, f is the favorable portion, and u is the unfavorable portion.

In the statistical significance test, the p-value shows the significance of the difference between groups is sufficient or not. The calculated p-value is often dependent on the standard error (SE). Moreover, the sample size has an impact on standard error and, accordingly, on the p-value. If the sample size rises, the standard error drops, and the p-value drop [63]. This sample size dependency makes p-values as confounded. Often a statistically significant finding means that a huge sample size is used [64], [65]. As a result, the p-value does not indicate the magnitude of the difference in the mean scores of the groups or the frequency of the relationship between the examined variables. Therefore, we conduct Wilcoxon Effect Size to demonstrate the magnitude of difference between forecasting performance of machine learning models.

V. EMPIRICAL RESULTS

In this section, we examine the forecasting abilities of conventional machine learning models with default hyperparameter settings. Then, we illustrate the best hyperparameter configuration for each machine learning model across all the datasets. After that, we review conventional machine learning models forecasting performance with the best hyperparameter settings. Finally, we analyze the effect of hyperparameter tuning in the selected machine learning models according to Saudi Stock Exchange.

A. FORECASTING PERFORMANCE OF UN-TUNED MACHINE LEARNING MODELS

Table 5 demonstrates the investigated forecasting performance in terms of RMSE and MAPE values over all the stock companies datasets. Maximum forecasting performance refers to minimum error is bolded and underlined for each dataset. Un-tuned Stochastic Gradient Descent (SGD) model consistently performed poorly with a very high error in all the datasets. One noticeable phenomenon is most of the models performed poorly in the Arab Sea dataset. On the other hand, the un-tuned kernel ridge regression (KRR) model performs considerably well by scoring the lowest RMSE and MAPE in nine datasets. Additionally, on the other datasets, it performs very close to the best-performing model.

To answer the research questions in our study, we reveal each model's performance in a win/loss scenario according to the statistical testing and forecasting performance error [66]. In each paired model comparison, if the null hypnosis is rejected using Wilcoxson statistical test with Bonferroni p-value correction, we look into the corresponding machine learning models performance score. Then we assign a win to the machine learning model with the least MAPE and assign a loss to the machine model with the high MAPE value. If the null hypnosis is accepted, we do not give win/loss to any machine learning model as there is no statistical difference between the compared models.

According to the win-loss scenario, Table 6 represents the outcome of Wilcoxson statistical testing to illustrate the significant difference in forecasting performance between each pair of un-tuned machine learning models. For illustration, let us consider Kernel Ridge regression's (KRR) forecasting performance in the context of all datasets except GASCO and TAIBA. KRR achieved the highest wins without any loss against any other models. But in GASCO and TAIBA, LASSO earned the highest wins without any loss. Overall, un-tuned KRR and LASSO can be labeled as the best two models in this scenario, with 62 and 55 wins, respectively. On the contrary, un-tuned SGD and GPR are the least performing forecasting models with 77 and 64 losses. Furthermore, we extend the analysis of the statistical test results in Table 7 by illustrating the paired comparison between each model. Each row reflects the number of wins for the row model against the column model, and likewise, the column model losses against the row model. For example, SVR has won nine times against KNN, and KNN has lost nine times against SVR. The final column computes the percentage of each model overall wins out of all possible paired comparisons, with each model having 77 paired comparisons (7 paired comparisons of each model multiplied by 11 stock datasets). Likewise, the loss percentage is computed in the final row. It is significant to mention that a paired comparison might not always result in a win or a loss, if there is no significant difference in performance between the two models.

TABLE 5. Un-Tuned Machine Learning Model Performance.

	DT		SVR		KNN		GPR		SGD		PLS		Kernel Ridge		LASSO	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
SAUDI ARAMCO	0.41	0.93	0.28	0.59	0.34	0.74	2.03	2.89	2e12	6e12	0.26	0.55	**0.25**	**0.52**	0.58	1.53
MAADEN	1.31	2.23	1.04	1.66	1.13	1.86	16.11	22.73	4e11	9e11	0.95	1.56	0.90	1.46	**0.89**	**1.45**
SAPTCO	0.47	2.29	0.29	1.32	0.35	1.57	43.35	73.04	2e9	1e10	0.30	1.35	**0.27**	**1.21**	0.28	1.23
SIDC	0.35	2.56	0.25	1.77	0.28	2.00	29.73	86.25	3e10	3e11	0.23	1.54	**0.21**	**1.29**	0.21	1.42
ALMARAI	1.15	1.63	0.94	1.26	1.01	1.43	15.60	17.84	5e11	1e12	0.94	1.24	**0.85**	**1.12**	0.85	1.13
CHEMICAL	0.77	2.10	0.44	1.09	0.53	1.45	15.70	17.24	2e11	9e11	0.47	1.18	**0.44**	**1.08**	0.44	1.09
ALINMA	0.46	1.80	0.52	1.46	0.44	1.60	21.37	44.29	1349	5757	0.27	1.04	**0.25**	**0.96**	0.74	3.99
ARAB SEA	38.45	42.40	47.68	55.36	37.12	40.20	66.41	89.80	7e8	1e9	2.68	2.77	**2.62**	**2.66**	3.33	3.66
STC	2.27	1.82	1.55	1.14	1.75	1.40	59.68	49.47	4e12	5e12	1.37	1.02	**1.23**	**0.93**	**1.24**	**0.93**
GASCO	0.63	1.37	0.43	0.90	0.43	0.98	17.10	16.93	6e11	2e12	0.42	0.84	**0.41**	**0.82**	**0.41**	**0.82**
TAIBA	1.06	2.32	0.42	0.80	0.52	1.16	5.10	5.79	3e11	1e12	0.42	0.80	0.40	0.76	**0.40**	**0.75**

TABLE 6. Win-Loss Statistical Performance of Un-Tuned Machine Learning models for Each Stock.

	DT		SVR		KNN		GPR		SGD		PLS		KRR		LASSO	
	W	L	W	L	W	L	W	L	W	L	W	L	W	L	W	L
SAUDI ARAMCO	2	2	**4**	**0**	3	1	1	4	0	7	**4**	**0**	**4**	**0**	1	5
MAADEN	2	5	4	2	3	4	1	6	0	7	4	2	**6**	**0**	**6**	**0**
SAPTCO	2	5	4	2	3	4	1	6	0	7	4	2	**6**	**0**	**6**	**0**
SIDC	2	5	4	3	3	4	1	6	0	7	5	1	**7**	**0**	5	1
ALMARAI	2	5	4	2	3	4	1	6	0	7	4	2	**6**	**0**	**6**	**0**
CHEMICAL	2	5	**5**	**0**	3	4	1	6	0	7	4	3	**5**	**0**	**5**	**0**
ALINMA	3	3	5	1	3	3	1	6	0	7	5	1	**7**	**0**	2	5
ARAB SEA	3	4	2	5	4	3	1	6	0	7	**6**	**0**	**6**	**0**	5	2
STC	2	5	4	3	3	4	1	6	0	7	5	2	**6**	**0**	**6**	**0**
GASCO	2	5	4	3	3	4	1	6	0	7	5	1	5	0	**6**	**0**
TAIBA	2	5	4	1	3	4	1	6	0	7	4	1	4	1	**7**	**0**
Total	24	49	44	22	34	39	11	64	0	77	50	15	**62**	**1**	55	13

TABLE 7. Un-Tuned ML Models Paired Comparison Results.

ML Model	DT	SVR	KNN	GPR	SGD	PLS	KRR	LASSO	Win	Win %
DT	-	1		10	11			2	24	31.17%
SVR	10	-	9	11	11	1		2	44	57.14%
KNN	9	1	-	11	11			2	34	44.16%
GPR				-	11				11	14.29%
SGD					-				0	0%
PLS	10	4	11	11	11	-		3	50	64.94%
KRR	11	8	10	11	11	7	-	4	**62**	**80.51%**
LASSO	9	8	9	10	11	7	1	-	55	71.42%
Loss	49	22	39	64	77	15	**1**	13		
Loss %	63.63%	31.17%	50.64%	83.12%	100%	19.48%	**1.29%**	16.88%		

In Table 7, the paired comparison exhibits the dominance of un-tuned KRR models in forecasting stock prices over other models with the highest win percentage. Additionally, it losses only once against the un-tuned LASSO, which can be labeled as the second-best model because it achieved the second-highest win ratio. In contrast, SGD and GPR have the highest loss ratio (i.e., 100% and 83.12%), indicating all other models have significantly outperformed these two models in almost all the stock datasets. Finally, figures of the test set forecasting performance for all the un-tuned models (total 88) are available for download as supplementary documents.

RQ1 Answer: Un-tuned KRR and LASSO models illustrate significantly superior or at least similar forecasting performance compared to other un-tuned models in forecasting the selected 11 Saudi companies' stock prices. On the contrary, un-tuned SGD and GPR models are the least performing models in stock price forecasting.

B. BEST HYPERPARAMETERS CONFIGURATION

In this study, we use grid search to find the best hyperparameters for each investigated machine learning model. Grid Search is applied in the training set using time split cross-validation with MAPE score as the optimizing scorer function. The grid search technique attempt to find the best value of hyperparameters using all possible combinations in the hyperparameter space. Though it is computationally heavy, it effectively finds the best hyperparameters of a machine learning model [55]. In our empirical study, the best hyperparameters combination is selected based on the minimum average MAPE score in nine folds of the training set. The total number of model fits is the multiplication of the nine folds with the combination of values in each hyperparameter space mentioned in Table 4.

Finally, Table V-A gives a comprehensive idea about the best value of hyperparameter of the proposed machine learning models in forecasting stock prices in the context of the selected stock datasets.

C. FORECASTING PERFORMANCE OF TUNED MACHINE LEARNING MODELS

Table 9 presents the explored forecasting performance of the hyperparameter optimized machine learning models across all datasets. After hyperparameter tuning, we observe a

TABLE 8. Best Hyperparameter Configuration.

	DT				SVR			KNN		GPR	SGD			PLS		KRR		LASSO	
	Max feature	Min Leaf	Min Split	Spitter	Kernel	C	Epsilon	algorithm	neigh-bors	Kernel	Learning Rate	Penalty	Max Iteration	n_com-ponents	Scale	Alpha	Kernel	selection	Alpha
SAUDI ARAMCO	Auto	2	6	Random	Linear	10	0.1	Auto	10	DotProduct+ WhiteKernel	Adaptive	L1	1000	2	FALSE	1	Linear	Cyclic	0.001
MAADEN	Auto	1	8	Random	Linear	0.1	0.1	Brute	10	DotProduct+ WhiteKernel	Adaptive	Elasticnet	3000	3	FALSE	0.001	Linear	Cyclic	0.001
SAPTCO	Auto	1	8	Random	Linear	1	0.001	Ball Tree	10	DotProduct+ WhiteKernel	Adaptive	L2	1000	3	FALSE	1	Linear	Cyclic	0.001
SIDC	Auto	1	8	Random	Linear	10	0.001	Auto	5	DotProduct+ WhiteKernel	Adaptive	L1	3000	2	FALSE	0.01	Linear	Cyclic	0.001
ALMARAI	Auto	1	4	Random	Linear	0.1	0.01	Brute	10	DotProduct+ WhiteKernel	Adaptive	L1	3000	3	FALSE	100	Linear	Cyclic	0.1
CHEMICAL	Auto	3	8	Random	Linear	0.1	0.001	Brute	10	DotProduct+ WhiteKernel	Adaptive	L2	1000	2	FALSE	100	Linear	Cyclic	0.1
ALINMA	Auto	5	2	Random	Linear	1	0.001	Brute	10	DotProduct+ WhiteKernel	invscaling	L2	3000	4	FALSE	0.01	Linear	Random	0.001
ARAB SEA	Auto	3	8	Random	Linear	100	0.001	Brute	10	DotProduct+ WhiteKernel	invscaling	L2	1000	3	FALSE	1	Linear	Random	0.001
STC	Auto	3	2	Best	Linear	0.1	0.01	Auto	10	DotProduct+ WhiteKernel	Adaptive	Elasticnet	3000	3	FALSE	0.001	Linear	Cyclic	0.1
GASCO	Auto	5	2	Best	Linear	10	0.001	Auto	10	DotProduct+ WhiteKernel	Adaptive	Elasticnet	3000	2	FALSE	0.001	Linear	Cyclic	0.001
TAIBA	Auto	2	6	Random	Linear	1	0.001	Auto	10	DotProduct+ WhiteKernel	Adaptive	L1	10000	2	FALSE	0.1	Linear	Cyclic	0.001

significant improvement in the forecasting performance of SGD and GPR models, which were the least performing models in their un-tuned state. However, the tuned SGD model performed poorly by scoring a massive error in forecasting SAUDI ARAMCO, STC, and ARAB SEA stock price. After hyperparameter tuning, SVR becomes one of the top forecasting models by scoring the minimum RMSE and MAPE values in all the selected datasets. On the contrary, little or no significant error reduction is seen in the KRR and LASSO models, which are the two best-performing models in their un-tuned state. However, KRR scored the minimum RMSE and MAPE in forecasting several stock datasets.

Table 10 demonstrates the Wilcoxson statistical testing results in the win-loss scenario to show the substantial gap in forecasting efficiency between each pair of hyperparameter tuned machine learning models. For instance, let us consider SVR forecasting performance in the context of all the datasets. SVR has the highest number of wins without losing to another model, indicating the supremacy of the tuned SVR model forecasting performance.

On the other hand, tuned DT is ranked as the lowest-performing forecasting model with 70 losses. It also signifies that the hyperparameter tuning might impact other models more extensively than the DT model because DT won 24 instances in the un-tuned comparison. In contrast, it wins only six instances after hyperparameter tuning. Moreover, GPR is one of the worst-performing models in the un-tuned version, but after hyperparameter tuning, we notice the dramatic improvement of forecasting performance by achieving 37 wins and outperforming tuned KRR and LASSO models, which are the two top-performing models before hyperparameter tuning. Furthermore, we extend the investigation of the statistical test results in table 10 by showing the paired comparison between each hyperparameter tuned model.

In table 11, the paired comparison unveiled the supremacy of the tuned SVR model in forecasting stock prices over other models. Moreover, tuned SVR has achieved 0% loss, suggesting that not a single tuned machine learning outperformed SVR in the context of all exploited stock prices datasets. Furthermore, tuned GPR has achieved the second-highest win ratio, which has only lost twice against the best-performing tuned SVR model. In contrast, tuned DT and SGD have the most loss ratio (i.e., 90.9%, 81.8%), signifying all the other tuned models have significantly performed better than these two models in almost all the stock datasets. Finally, figures of the test set forecasting performance for all the tuned models (total 88) are available for download as supplementary documents.

RQ2 Answer: Tuned SVR and GPR models demonstrate significantly superior or at least similar forecasting performance compared to other tuned models in forecasting the selected 11 Saudi companies' stock prices. On the contrary, tuned DT and SGD models are the least performing models in stock price forecasting.

TABLE 9. Tuned Machine Model Performance.

	DT		SVR		KNN		GPR		SGD		PLS		KRR		LASSO	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
SAUDI ARAMCO	0.41	0.88	**0.25**	**0.51**	0.35	0.74	0.25	0.52	6e10	1e11	0.25	0.52	0.25	0.52	0.25	0.53
MAADEN	1.26	2.11	**0.89**	**1.45**	1.13	1.86	0.90	1.46	0.95	1.55	0.91	1.46	0.90	1.46	0.91	1.48
SAPTCO	0.37	1.77	**0.27**	**1.20**	0.33	1.47	0.27	1.21	0.28	1.25	0.28	1.22	0.27	1.21	0.27	1.21
SIDC	0.31	2.23	**0.21**	**1.26**	0.26	1.83	0.21	1.32	0.25	1.64	0.23	1.54	0.21	1.29	0.21	1.33
ALMARAI	1.09	1.51	**0.85**	**1.12**	1.04	1.45	**0.85**	**1.12**	0.90	1.22	0.86	1.14	**0.85**	**1.12**	0.85	1.13
CHEMICAL	0.63	1.77	**0.43**	**1.06**	0.49	1.32	0.44	1.08	0.46	1.20	0.47	1.18	0.44	1.07	0.43	1.07
ALINMA	0.45	1.75	**0.25**	**0.96**	0.44	1.57	**0.25**	**0.96**	0.47	2.04	**0.25**	**0.96**	**0.25**	**0.96**	**0.25**	**0.96**
ARAB SEA	37.36	40.73	**2.59**	**2.62**	37.51	41.00	2.62	2.66	7e8	1e8	2.64	2.67	2.62	2.66	2.64	2.67
STC	1.98	1.55	**1.23**	**0.93**	1.73	1.35	**1.23**	**0.93**	4e6	3e6	1.24	0.93	1.24	0.93	1.24	0.93
GASCO	0.49	1.10	**0.36**	**0.73**	0.42	0.93	0.41	0.82	1.05	3.25	0.42	0.84	0.41	0.82	0.41	0.82
TAIBA	0.75	1.63	**0.40**	**0.75**	0.49	1.03	0.40	0.76	0.41	0.80	0.42	0.80	0.40	0.76	0.40	0.77

TABLE 10. Win-Loss Statistical Performance of Hyperparameter Tuned ML models for Each Stock.

	DT		SVR		KNN		GPR		SGD		PLS		KRR		LASSO	
	W	L	W	L	W	L	W	L	W	L	W	L	W	L	W	L
SAUDI ARAMCO	1	5	2	0	1	1	2	0	0	7	2	0	**3**	**0**	2	0
MAADEN	0	7	**5**	**0**	1	6	3	0	2	5	3	1	3	0	3	1
SAPTCO	0	7	**3**	**0**	1	6	**3**	**0**	2	5	**3**	**0**	**3**	**0**	**3**	**0**
SIDC	0	7	**7**	**0**	1	5	5	1	1	5	3	4	5	1	4	3
ALMARAI	0	7	**3**	**0**	1	6	3	0	2	5	**3**	**0**	**3**	**0**	**3**	**0**
CHEMICAL	0	7	**4**	**0**	1	6	4	0	2	4	2	4	**4**	**0**	**4**	**0**
ALINMA	1	6	**3**	**0**	2	5	3	0	0	7	**3**	**0**	**3**	**0**	**3**	**0**
ARAB SEA	2	5	**3**	**0**	1	6	3	0	0	7	**3**	**0**	**3**	**0**	**3**	**0**
STC	1	6	**3**	**0**	2	5	3	0	0	7	**3**	**0**	**3**	**0**	**3**	**0**
GASCO	1	6	**7**	**0**	2	5	4	1	0	7	3	3	3	1	4	1
TAIBA	0	7	**5**	**0**	1	6	4	0	2	4	2	1	3	2	3	0
Total	6	70	**45**	**0**	14	57	37	2	11	63	30	13	36	4	35	5

TABLE 11. Hyperparameter Tuned ML Models Paired Comparison Results.

ML Model	DT	SVR	KNN	GPR	SGD	PLS	KRR	LASSO	Win	Win %
DT	-		1		5				6	7.79%
SVR	11	-	10	2	11	5	3	3	45	**58.4%**
KNN	9		-		5				14	18.18%
GPR	11		10	-	11	3	1	1	37	48.05%
SGD	6		5		-				11	14.29%
PLS	11		10		9	-			30	38.96%
KRR	11		11		11	2	-	1	36	46.75%
LASSO	11		10		11	3		-	35	45.45%
Loss	70	**0**	57	2	63	13	4	5		
Loss %	90.9%	**0**	74.0%	2.6%	81.8%	16.9%	5.2%	6.5%		

D. EFFECTS OF HYPERPARAMETER TUNING IN MACHINE LEARNING MODELS

At first, we visualize the effect of hyperparameter tuning by comparing tuned and un-tuned graphs of a model's forecasting performance in the test set. Then we illustrate the impact of hyperparameter tuning using the Wilcoxson effect size test.

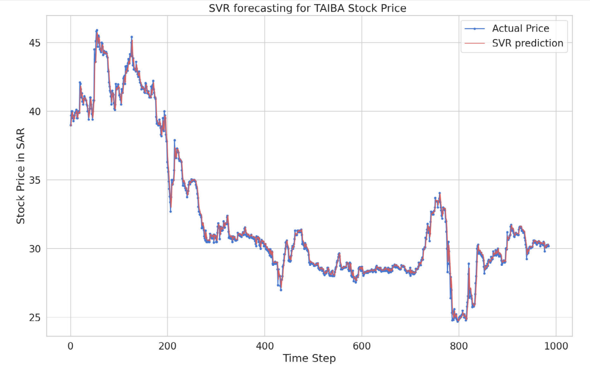
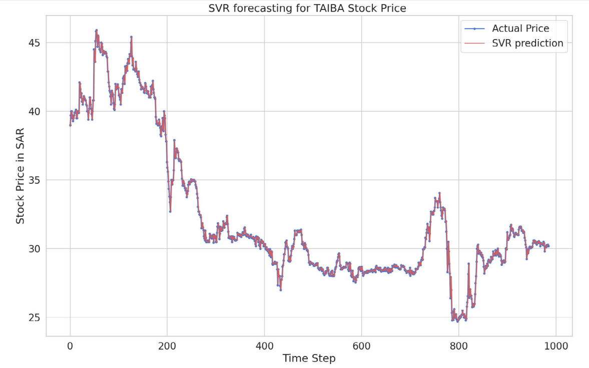
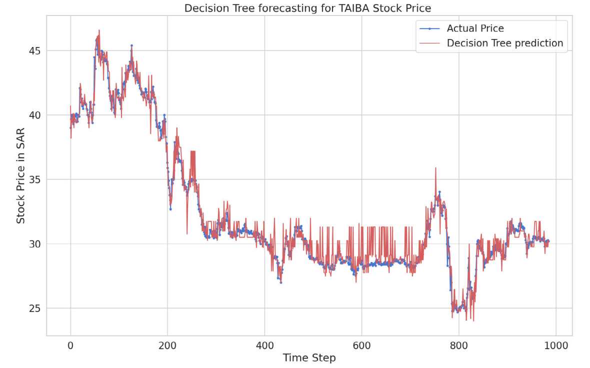
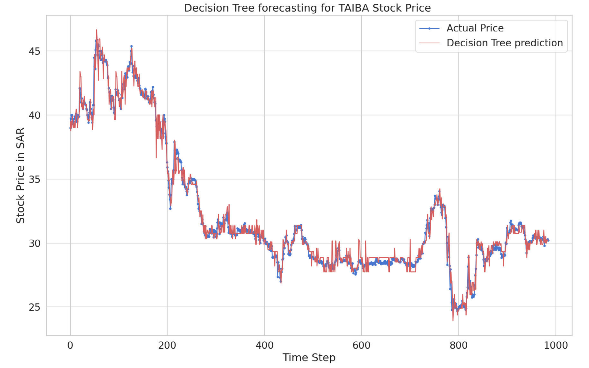
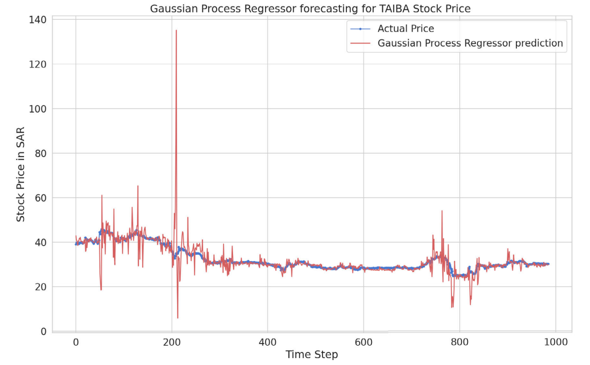
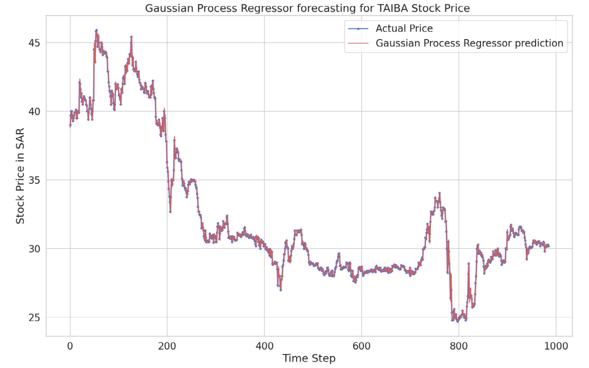
1) VISUALIZING THE IMPACT OF HYPERPARAMETER TUNING

Table 12, 13, 14 illustrate comparisons of tuned and un-tuned models by plotting forecasting performance. In those tables, the un-tuned and tuned pair of figures demonstrate different effects of hyperparameter tuning on the corresponding machine learning models. In table 12, the difference between un-tuned and tuned SVR model forecasting performance in the TAIBA dataset can not be determined by figures. But Wilcoxson's effect size test results in a positive small effect sizes, which denotes a slight improvement of performance after hyperparameter tuning. On the other hand, after tuning the hyperparameters, DT and GPR models' performance improvement can easily be identified. These models' predicted value is closer to the actual value in the tuned version than the un-tuned version. Wilcoxson effect size test acknowledges this phenomenon by resulting in medium and large effect size, respectively.

In table 13, LASSO and KRR models forecasting performance are presented where visually no difference can be identified between tuned and un-tuned models' performance. But Wilcoxson test reports negative negligible and negative small effects, respectively, which denotes the forecasting performance degradation after hyperparameter tuning. From the un-tuned forecasting performance figures, we can notice the remarkable forecasting performance of un-tuned LASSO and KRR. After having good forecasting performance, when we try to optimize the hyperparameter, it overfits the corresponding model and results in slightly poor performance. As a result, hyperparameter tuning of LASSO and KRR leads to insignificant performance degradation.

In table 14, the positive large effect in the SVR model illustrates remarkable forecasting improvement after hyperparameter tuning, whereas negative large impact in the DT

TABLE 12. Forecasting Performance of Un-Tuned vs. Tuned Models in TAIBA dataset (Positive Effects).

Un-tuned Model Forecasting Performance	Tuned Model Forecasting Performance	Wilcoxson Effect Size
		Small (S)
		Medium (M)
		Large (L)

model demonstrates poor forecasting performance in both un-tuned and tuned versions of the machine learning algorithm. When a machine learning model is performing extremely poor, slight or moderate degradation can cause a large effect on forecasting performance.

2) WILCOXSON EFFECT SIZE TEST

Table 15 illustrates the impact of hyperparameter tuning of each machine learning model in all datasets. In each column, an un-tuned and tuned model forecasting performance is compared and reported according to Wilcoxson's effect size. For example, in the second column, the forecasting performance of un-tuned SVR and tuned SVR is evaluated using Wilcoxson effect size across all stock datasets. The range of Wilcoxson effect size is [-1,1]. A positive value of effect size indicates that hyperparameters tuning enhances the forecasting performance. While a negative value denotes that hyperparameter tuning degraded the forecasting performance. The standard interpretation of Wilcoxson effect size value is as follows:

- Negligible Effect (N) < 0.1
- $0.1 \leq$ Small Effect (S) < 0.3
- $0.3 \leq$ Medium Effect (M) < 0.5
- Large Effect (L) ≥ 0.5

Table 16 represents the Wilcoxson effect size of hyperparameter tuning with the standard interpretation mentioned above.

Previously in section V-C, we assumed that the hyperparameter tuning of GPR and SGD might have a significant effect which is now empirically proven according to the table.

TABLE 13. Forecasting Performance of Un-Tuned vs. Tuned Models in ALMARAI and TAIBA dataset (Negative Effects).

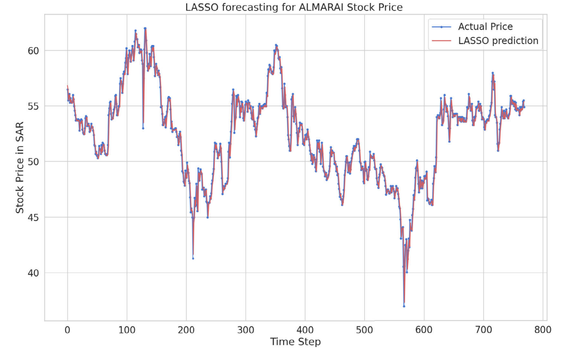
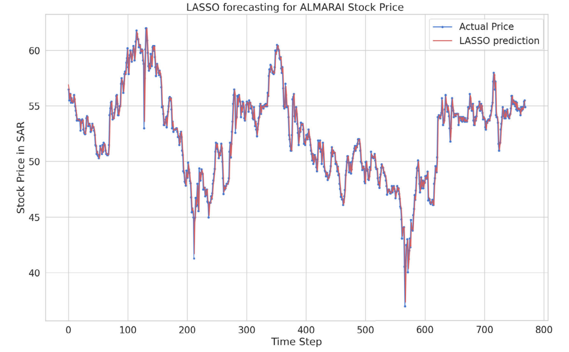
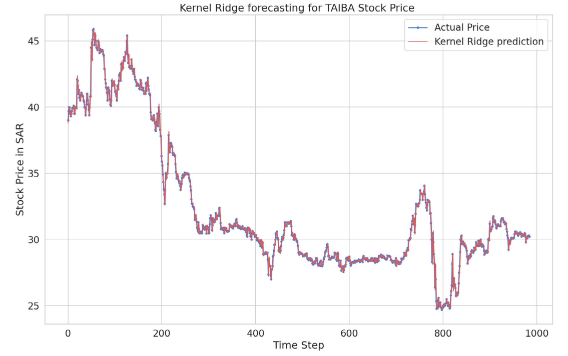
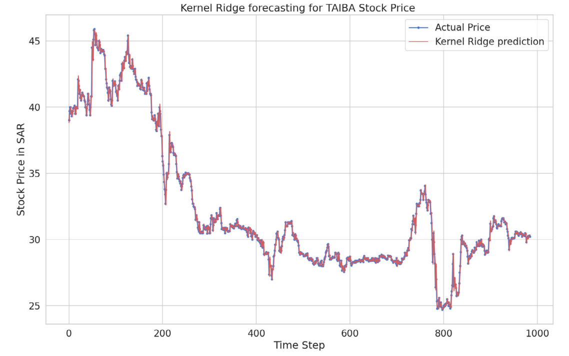
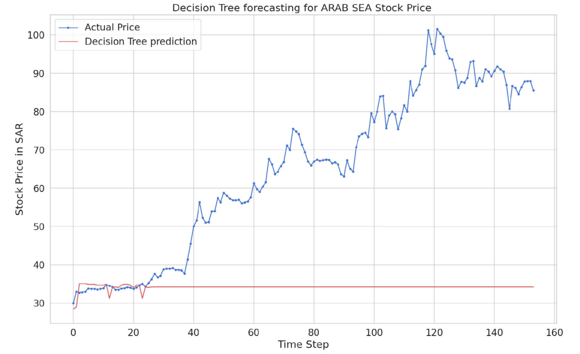
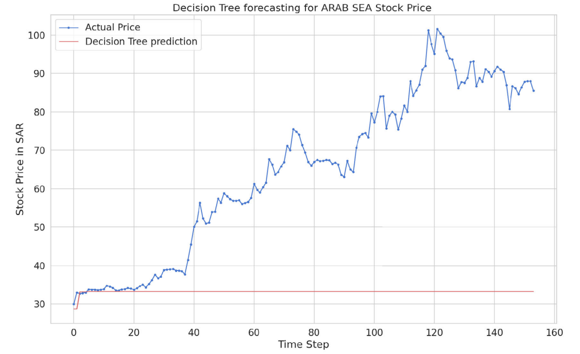
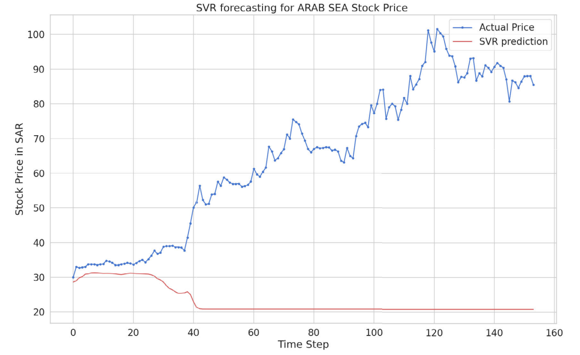
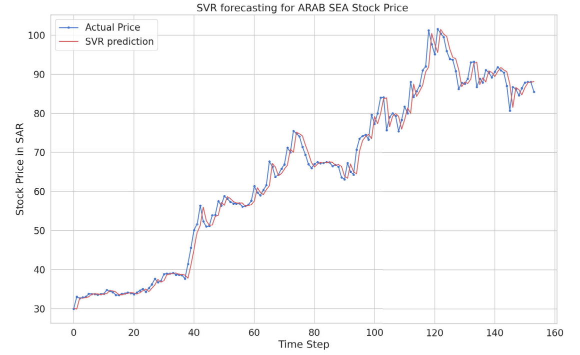
Un-tuned Model Forecasting Performance	Tuned Model Forecasting Performance	Wilcoxson Effect Size
 LASSO forecasting for ALMARAI Stock Price	 LASSO forecasting for ALMARAI Stock Price	Negative Negligible (-N)
 Kernel Ridge forecasting for TAIBA Stock Price	 Kernel Ridge forecasting for TAIBA Stock Price	Negative Small (-S)

TABLE 14. Forecasting Performance of Un-Tuned vs. Tuned Models in ARAB SEA dataset.

Un-tuned Model Forecasting Performance	Tuned Model Forecasting Performance	Wilcoxson Effect Size
 Decision Tree forecasting for ARAB SEA Stock Price	 Decision Tree forecasting for ARAB SEA Stock Price	Negative Large (-L)
 SVR forecasting for ARAB SEA Stock Price	 SVR forecasting for ARAB SEA Stock Price	Large (L)

So, we can say that hyperparameter tuning of the GPR and SGD model has a major impact on improving forecasting performance over all the stock datasets utilized in our experiment. Although, in the ARAB SEA stock dataset, we notice

TABLE 15. Wilcoxson Effect Size of Un-Tuned vs. Tuned Models.

	DT	SVR	KNN	GPR	SGD	PLS	KRR	LASSO
ARAMCO	0.022	0.251	-0.037	0.799	1.0	0.215	0	0.885
MAADEN	0.019	0.245	0	0.966	1.0	0.121	-0.038	-0.11
SAPTCO	0.279	0.249	0.157	0.943	1.0	0.233	0	0.129
SIDC	0.058	0.47	0.143	0.959	1.0	-0.389	-0.049	0.273
ALMARAI	0.034	0.284	-0.02	0.949	1.0	0.187	-0.045	-0.048
CHEMICAL	0.211	0.155	0.221	0.941	1.0	-0.308	0.107	0.162
ALINMA	-0.083	0.180	0.096	0.972	1.0	0.172	0.018	0.947
ARAB SEA	0.862	0.999	-0.978	0.999	0.25	0.115	-0.057	0.487
STC	0.147	0.316	0.096	0.995	1.0	0.206	-0.006	0.055
GASCO	0.267	0.364	0.138	0.937	1.0	-0.354	0.338	0.033
TAIBA	0.225	0.179	0.219	0.906	1.0	-0.275	0	-0.119

TABLE 16. Interpreting Wilcoxson Effect Size of Un-Tuned vs. Tuned Models.

	DT	SVR	KNN	GPR	SGD	PLS	KRR	LASSO
ARAMCO	N	S	-N	L	L	S	N	L
MAADEN	N	S	N	L	L	S	-N	-S
SAPTCO	S	S	S	L	L	S	N	S
SIDC	N	M	S	L	L	-M	-N	S
ALMARAI	N	S	-N	L	L	S	-N	-N
CHEMICAL	S	S	S	L	L	-M	S	S
ALINMA	-N	S	N	L	L	S	N	L
ARAB SEA	L	L	-L	L	S	S	-N	M
STC	S	M	N	L	L	S	-N	N
GASCO	S	M	S	L	L	-M	M	N
TAIBA	S	S	S	L	L	-S	N	-S

a negligible effect of hyperparameter tuning in the SGD model performance. The reason behind this phenomenon lies in section V-C, where we showed that in the ARAB SEA dataset, the tuned SGD model performed remarkably poorly even after hyperparameter tuning. Moreover, tuning SVR has consistently improved the forecasting performance by minor to major impact, supporting results in section V-C, where we show that tuned SVR is the best performing model. Additionally, tuning SVR and GPR models have never degraded the forecasting performance across the stock datasets used in this study. On the contrary, we can visualize and comprehend the reason behind the performance degradation of tuned DT, KRR, LASSO, and PLS in section V-C, using the table 16. For example, after hyperparameter tuning in PLS, forecasting performance deteriorates in four stock datasets. Additionally, in most of the datasets, the effect of hyperparameter tuning is negligible to small. Furthermore, in KRR, the impact of hyperparameter tuning is found to be negligible across nine stock datasets. Thus it reveals the null or negative effect of hyperparameter tuning on these models.

RQ3 Answer: Hyperparameter tuning of SVR, GPR, and SGD significantly improve forecasting performance or at least have a positive impact, whereas hyperparameter tuning in other models may reduce the forecasting performance in the context of forecasting the selected 11 Saudi companies' stock prices.

VI. THREATS TO VALIDITY

Our findings are restricted to the selected 11 Saudi Stock Exchange (Tadawul) companies used in this study and cannot be generalized to other stock companies. As a result, replication of this empirical study is required using additional stock datasets around the world. Random initialization in machine learning models can lead to a threat to validity by generating non-deterministic results. In this experiment, each model is fit 30 times in each dataset, and the average error is calculated to gain confidence in our empirical findings. Consequently, our experiment results are consistent and robust, minimizing the risk of this particular threat to validity. Moreover, an additional threat to the validity is associated with the different evaluation metrics used for measuring forecasting capabilities. Nevertheless, our empirical study performance measures and statistical testing are widely applied and mathematically accepted in the research community.

VII. CONCLUSION

This paper aimed to investigate conventional machine learning models forecasting capabilities in the Saudi Stock Exchange (Tadawul) context. Moreover, we empirically investigated the impact of hyperparameter tuning on a machine learning model forecasting performance. A thorough investigation of eight conventional machine learning models forecasting performance was conducted using 11 stock datasets from different sectors in Tadawul. This paper's primary contributions can be summarized in four folds: First, we compared the applicability of un-tuned conventional machine learning models in forecasting stocks price. Second, we searched the hyperparameter space for each conventional machine learning model and reported the best hyperparameters combination in each stock dataset. Third, we compared the forecasting performance of the machine learning models after tuning. Fourth, we analyzed to which extent hyperparameter tuning affects the performance of conventional machine learning models in forecasting stock prices.

Our empirical study findings demonstrate the applicability of conventional machine learning models to forecast stock prices for data analysts, stock investors, and machine learning practitioners. Conventional machine learning models like SVR and KRR can be employed to precisely forecast stock prices, which might be an excellent tool for a stock investor. Our empirical results align with existing literature suggesting superior forecasting performance of SVR and KRR in time series forecasting [67]–[69]. The ability to efficiently address nonlinear regression is the primary reason for their noteworthy performance, which is also supported by the existing studies [70], [71]. Empirical findings revealed that hyperparameter tuning of machine learning models could extensively impact forecasting performance. Suppose, after hyperparameter tuning, SVR became one of the best models in forecasting stock prices. On the contrary, un-tuned best performing models like KRR, LASSO, and PLS showed a negligible or negative impact on hyperparameter tuning. As in these scenarios, negligible impact implies that hyperparameters tuning result in a similar model compared with the default un-tuned model. While the hyperparameter tuning overfits the model, resulting in a negative impact denoting forecasting performance degradation after hyperparameter tuning. Nevertheless, hyperparameter tuning could be a better choice for improving the machine learning model forecasting performance while avoiding overfitting.

Our work in this paper can be expanded in several directions. This experiment can be replicated using additional stock datasets of varying sizes from other global stock markets, and findings can be compared to those reported in this paper. In addition, a sensitivity analysis can be performed to identify which hyperparameter has the most crucial impact on enhancing the forecasting performance for each machine learning model. Finally, investigating deep neural networks and ensemble models can be an interesting future direction for researchers to evaluate whether these can achieve additional forecasting performance compared with the conventional models.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia in the development of this work.

REFERENCES

[1] R. J. Teweles and E. S. Bradley, *The Stock Market*, vol. 64. Hoboken, NJ, USA: Wiley, 1998.

[2] S. Mehtab, J. Sen, and S. Dasgupta, ''Robust analysis of stock price time series using CNN and LSTM-based deep learning models,'' in *Proc. 4th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Nov. 2020, pp. 1481–1486.

[3] A. Azlan, Y. Yusof, and M. F. M. Mohsin, ''Univariate financial time series prediction using clonal selection algorithm,'' *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 10, no. 1, pp. 151–156, 2020.

[4] S. De, A. K. Dey, and D. K. Gouda, ''Construction of confidence interval for a univariate stock price signal predicted through long short term memory network,'' *Ann. Data Sci.*, vol. 2020, pp. 1–14, Jul. 2020.

[5] J. Du, Q. Liu, K. Chen, and J. Wang, ''Forecasting stock prices in two ways based on LSTM neural network,'' in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 1083–1086.

[6] J.-S. Chou and T.-K. Nguyen, ''Forward forecast of stock price using sliding-window Metaheuristic-optimized machine-learning regression,'' *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3132–3142, Jul. 2018.

[7] B. M. Henrique, V. A. Sobreiro, and H. Kimura, ''Stock price prediction using support vector regression on daily and up to the minute prices,'' *J. Finance Data Sci.*, vol. 4, no. 3, pp. 183–201, 2018.

[8] S. O. Olatunji, M. Saad Al-Ahmadi, M. Elshafei, and Y. A. Fallatah, ''Saudi Arabia stock prices forecasting using artificial neural networks,'' in *Proc. 4th Int. Conf. Appl. Digit. Inf. Web Technol. (ICADIWT)*, Aug. 2011, pp. 81–86.

[9] C. M. Authority, ''Corporate governance regulations in the Kingdom of Saudi Arabia,'' Capital Markets Authority Saudi Arabia, Riyadh, Saudi Arabia, Tech. Rep. 1-7-2021, 2006.

[10] B. A. Gouda, ''The Saudi securities law: Regulation of the Tadawul stock market, issuers, and securities professionals under the Saudi capital market law of 2003,'' *Ann. Surv. Int. Comput.*, vol. 18, p. 115, Dec. 2012.

[11] *Tadawul Annual Report*, Tadawul, Riyadh, Saudi Arabia, 2019.

[12] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*. Boca Raton, FL, USA: CRC Press, 2016.

[13] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, ''Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data,'' *Ecol. Model.*, vol. 406, pp. 109–120, Aug. 2019.

[14] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning: Methods, Systems, Challenges*. Basingstoke, U.K.: Springer, 2019.

[15] P. Probst, M. N. Wright, and A. Boulesteix, ''Hyperparameters and tuning strategies for random forest,'' *WIREs Data Mining Knowl. Discovery*, vol. 9, no. 3, p. e1301, May 2019.

[16] L. Yang and A. Shami, ''On hyperparameter optimization of machine learning algorithms: Theory and practice,'' *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.

[17] S. Abreu, ''Automated architecture design for deep neural networks,'' 2019, *arXiv:1908.10714*.

[18] O. S. Steinholtz, ''A comparative study of black-box optimization algorithms for tuning of hyper-parameters in deep neural networks,'' Ph.D. dissertation, Dept. Elect. Eng., Luleå Univ. Technol., Luleå, Sweden, 2018.

[19] S. Ho and M. Xie, ''The use of ARIMA models for reliability forecasting and analysis,'' *Comput. Ind. Eng.*, vol. 35, nos. 1–2, pp. 213–216, 1998.

[20] P.-F. Pai and C.-S. Lin, ''A hybrid ARIMA and support vector machines model in stock price forecasting,'' *Omega*, vol. 33, pp. 497–505, Dec. 2005.

[21] D. Bhuriya, G. Kaushal, A. Sharma, and U. Singh, ''Stock market predication using a linear regression,'' in *Proc. Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Apr. 2017, pp. 510–513.

[22] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, ''Stock price prediction using the ARIMA model,'' in *Proc. 16th Int. Conf. Comput. Modeling Simulation*, Mar. 2014, pp. 106–112.

[23] T. T.-L. Chong and W.-K. Ng, ''Technical analysis and the London stock exchange: Testing the MACD and RSI rules using the FT30,'' *Appl. Econ. Lett.*, vol. 15, no. 14, pp. 1111–1114, Nov. 2008.

[24] J. S. Armstrong, *Principles of Forecasting: A Handbook for Researchers and Practitioners*, vol. 30. Berlin, Germany: Springer, 2001.

[25] A. Atla, R. Tada, V. Sheng, and N. Singireddy, ''Sensitivity of different machine learning algorithms to noise,'' *J. Comput. Sci. Colleges*, vol. 26, no. 5, pp. 96–103, 2011.

[26] A. W. Li and G. S. Bastos, ''Stock market forecasting using deep learning and technical analysis: A systematic review,'' *IEEE Access*, vol. 8, pp. 185232–185242, 2020.

[27] L. K. Shrivastav and R. Kumar, ''An empirical analysis of stock market price prediction using arima and SVM,'' in *Proc. 6th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, 2019, pp. 173–178.

[28] J. R. Quinlan, ''Induction of decision trees,'' *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[29] V. N. Vapnik, ''An overview of statistical learning theory,'' *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[30] M. Awad and R. Khanna, ''Support vector regression,'' in *Efficient Learning Machine*. New York, NY, USA: Springer, 2015, pp. 67–80.

[31] A. J. Smola and B. Schölkopf, ''A tutorial on support vector regression,'' *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[32] D. W. Aha, D. Kibler, and M. K. Albert, ''Instance-based learning algorithms,'' *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.

[33] N. Meade, ''A comparison of the accuracy of short term foreign exchange forecasting methods,'' *Int. J. Forecasting*, vol. 18, no. 1, pp. 67–83, 2002.

[34] C. E. Rasmussen, ''Gaussian processes in machine learning,'' in *Summer School Machine Learning*. New York, NY, USA: Springer, 2003, pp. 63–71.

[35] L. Bottou, ''Large-scale machine learning with stochastic gradient descent,'' in *Proc. COMPSTAT*. New York, NY, USA: Springer, 2010, pp. 177–186.

[36] L. Bottou, ''Stochastic gradient descent tricks,'' in *Neural Networks: Tricks Trade*. New York, NY, USA: Springer, 2012, pp. 421–436.

[37] A. Alazba and H. Aljamaan, ''Code smell detection using feature selection and stacking ensemble: An empirical investigation,'' *Inf. Softw. Technol.*, vol. 138, Oct. 2021, Art. no. 106648.

[38] F. Kabir, S. Siddique, M. R. A. Kotwal, and M. N. Huda, ''Bangla text document categorization using stochastic gradient descent (SGD) classifier,'' in *Proc. Int. Conf. Cognit. Comput. Inf. Process. (CCIP)*, Mar. 2015, pp. 1–4.

[39] J. O. Ighalo, A. G. Adeniyi, and G. Marques, ''Application of linear regression algorithm and stochastic gradient descent in a machine-learning environment for predicting biomass higher heating value,'' *Biofuels, Bioproducts Biorefining*, vol. 14, no. 6, pp. 1286–1295, Nov. 2020.

[40] N. Deepa, B. Prabadevi, P. K. Maddikunta, T. R. Gadekallu, T. Baker, M. A. Khan, and U. Tariq, ''An AI-based intelligent system for healthcare analysis using ridge-adaline stochastic gradient descent classifier,'' *J. Supercomput.*, vol. 77, no. 2, pp. 1998–2017, Feb. 2021.

[41] J. P. Guilford, *Fundamental Statistics in Psychology and Education*, 2nd ed. New York, NY, USA: McGraw-Hill, 1950.

[42] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Proc. Int. Stat. Optim. Perspect. Workshop Subspace, Latent Struct. Feature Selection*. New York, NY, USA: Springer, 2005, pp. 34–51.

[43] N. Cristianini, *An Introduction to Support Vector Machine Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[44] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 515–521.

[45] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., Ser. B, Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[46] V. R. Basili and H. D. Rombach, "The TAME project: Towards improvement-oriented software environments," *IEEE Trans. Softw. Eng.*, vol. SE-14, no. 6, pp. 758–773, Jun. 1988.

[47] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.

[48] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, 2012.

[49] Y. Bengio, "Gradient-based optimization of hyperparameters," *Neural Comput.*, vol. 12, no. 8, pp. 1889–1900, 2000.

[50] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

[51] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," 2019, *arXiv:1912.06059*.

[52] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Melbourne, VIC, Australia: OTexts, 2018.

[53] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statist. Surv.*, vol. 4, pp. 40–79, Jan. 2010.

[54] Y. Zhao, J. Li, and L. Yu, "A deep learning ensemble approach for crude oil price forecasting," *Energy Econ.*, vol. 66, pp. 9–16, Aug. 2017.

[55] L. Tang, Y. Wu, and L. Yu, "A randomized-algorithm-based decomposition-ensemble learning methodology for energy price forecasting," *Energy*, vol. 157, pp. 526–538, Aug. 2018.

[56] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[57] C. Tofallis, "A better measure of relative prediction accuracy for model selection and model estimation," *J. Oper. Res. Soc.*, vol. 66, no. 8, pp. 1352–1362, Aug. 2015.

[58] T. Gneiting, "Making and evaluating point forecasts," *J. Amer. Stat. Assoc.*, vol. 106, no. 494, pp. 746–762, Jun. 2011.

[59] F. Wilcoxon, "Individual comparisons by ranking methods," *Breakthroughs Statistics*. New York, NY, USA: Springer, 1992, pp. 196–202.

[60] J. M. Bland and D. G. Altman, "Multiple significance tests: The Bonferroni method," *BMJ*, vol. 310, no. 6973, p. 170, 1995.

[61] J. H. Zar, *Biostatistical Analysis*. London, U.K.: Pearson, 1999.

[62] D. S. Kerby, "The simple difference formula: An approach to teaching nonparametric correlation," *Comprehensive Psychol.*, vol. 3, Jan. 2014, Art. no. 1131.

[63] D. G. Altman and J. M. Bland, "Standard deviations and standard errors," *BMJ*, vol. 331, no. 7521, p. 903, Oct. 2005.

[64] G. M. Sullivan and R. Feinn, "Using effect size-or why the p value is not enough," *J. Graduate Med. Educ.*, vol. 4, no. 3, p. 279, 2012.

[65] M. T. Bradley and A. Brand, "Alpha values as a function of sample size, effect size, and power: Accuracy over inference," *Psychol. Rep.*, vol. 112, no. 3, pp. 835–844, Jun. 2013.

[66] H. Aljamaan and A. Alazba, "Software defect prediction using tree-based ensembles," in *Proc. 16th ACM Int. Conf. Predictive Models Data Anal. Softw. Eng.*, Nov. 2020, pp. 1–10.

[67] A. Altan and S. Karasu, "The effect of kernel values in support vector machine to forecasting performance of financial time series," *J. Cognit. Syst.*, vol. 4, no. 1, pp. 17–21, 2019.

[68] M. A. Villegas, D. J. Pedregal, and J. R. Trapero, "A support vector machine for model selection in demand forecasting applications," *Comput. Ind. Eng.*, vol. 121, pp. 1–7, Jul. 2018.

[69] P. Jiang, R. Li, N. Liu, and Y. Gao, "A novel composite electricity demand forecasting framework by data processing and optimized support vector machine," *Appl. Energy*, vol. 260, Feb. 2020, Art. no. 114243.

[70] R. Samsudin, A. Shabri, and P. Saad, "A comparison of time series forecasting using support vector machine and artificial neural network model," *J. Appl. Sci.*, vol. 10, pp. 950–958, Nov. 2010.

[71] P. Exterkate, P. J. F. Groenen, C. Heij, and D. V. Dijk, "Nonlinear forecasting with many predictors using kernel ridge regression," *Int. J. Forecast.*, vol. 32, no. 3, pp. 736–753, Jul. 2016.

KAZI EKRAMUL HOQUE received the B.Sc. degree in computer science and engineering from the Chittagong University of Engineering and Technology (CUET), in 2015. He is currently pursuing the master's degree in computer science with the Department of Information and Computer Science (ICS), King Fahd University of Petroleum and Minerals (KFUPM). He started working as a Faculty Member with the Department of Computer Science and Engineering, Premier University Chittagong (PUC), in 2016. His research interests include machine learning, time series forecasting, software security, empirical software engineering, and deep learning.

HAMOUD ALJAMAAN received the Ph.D. degree in computer science from the University of Ottawa, Canada, in December 2015. Afterwards in January 2016, he rejoined the Information and Computer Science (ICS) Department, King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, as an Assistant Professor. His research interests include machine learning, code smell detection, software defects, software maintainability, and time series analysis. He has been appointed as Head of the Department, since 2019. He led the major revisions for the computer science and software engineering undergraduate programs. In addition, established the masters programs in artificial intelligence and cybersecurity. Lastly, established the annual "4IR Data Science Summer School" for Saudi Aramco employees, since 2019.