STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–11.
Published online in International Academic Press (www.IAPress.org)

Enhancing Interval Forecasting Accuracy of Iraqi Stock Market Prices based on v-support vector regression

Noor Adnan Ahmed Abdullah, Zakariya Yahya Algamal*

Department of Statistics and Informatics, University of Mosul, Mosul, Iraq

Abstract Stock price forecasting poses significant challenges due to non-stationarity, nonlinearity, and noise in financial markets, particularly for the Iraqi stock exchange. This study proposes an enhanced interval-valued forecasting model for daily prices of the national chemical and plastic industry (WSKB) company (2020–2025) using v-support vector regression (VSVR) with hyperparameters optimized via the coati optimization algorithm (COA). Interval time series are constructed from lower and upper price bounds, modeling center and radius components to capture uncertainty more effectively than point forecasts. The COA approach tunes key VSVR parameters through population-based exploration and exploitation phases inspired by coati hunting behaviors, outperforming grid search (GS-VSVR) and cross-validation (CV-VSVR). On training data (637 days), COA-VSVR achieves superior metrics for center (MAE=0.158, RMSE=0.253, DA=0.665, R^2=0.965) and radius (MAE=0.169, RMSE=0.264, DA=0.642, R^2=0.957) compared to baselines; testing results (308 days) confirm robustness. Further, Diebold-Mariano tests validate center-based COA-VSVR superiority over radius-based at 95% confidence ($p<0.05$). Visualizations and error reductions demonstrate the model's practical value for risk-aware investment in volatile emerging markets.

Keywords Interval-valued time series, stock price forecasting, v-support vector regression, coati optimization algorithm, Hyperparameter tuning, Iraqi stock market

DOI: 10.19139/soic-2310-5070-3316

1. Introduction

Stock price forecasting has become a very important task in the field of finance that is expected to establish the future price or trend of stocks, to enable the investors and traders make a wise decision. It entails modeling the patterns and trends in the market by examining historical price data, market indicators and external factors [1, 2, 3, 4]. The financial markets are characterized by a combination of the appropriate signals and random noise and it is hard to distinguish between the important and unimportant fluctuations [5, 6]. This noise will seriously affect the quality of forecasting as well as cause volatility on the prediction. The prices of stock tend to be non-stationary in that the statistical characteristics, such as the mean and the variance, of the price vary with time. In addition, the market forces are very non-linear because of the multifaceted interplay of economic, political and psychological forces and the non-linearity of these forces makes it very difficult to apply the traditional methods [6].

Stock price forecasting methods encompass a range of traditional statistical approaches and modern machine learning models. Statistic based stock price forecasting has a long history and will not disappear due to the emergence of machine learning and deep learning methods. These conventional statistical methods tend to concentrate more on the time series model and make use of the past trends of stock prices to draw predictions [7]. Statistical stock price forecasting focuses on the time series analysis, such as autoregressive integrated

*Correspondence to: Zakariya Yahya Algamal (Email: zakariya.algamal@uomosul.edu.iq). Department of Statistics and Informatics, University of Mosul, Mosul, Iraq.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

moving average (ARIMA), and volatility models, such as generalized autoregressive conditional heteroskedasticity (GARCH) [8]. These approaches are interpretable and effective when dealing with nonmoving linear data, but become problematic when dealing with complicated nonlinear market dynamics. The advantages of statistical models are that they are easy to interpret, have a good theoretical basis and are effective when the data are linear and stationary [9].

Stock markets are, however, by definition, noisy, nonlinear and impacted by a myriad of external factors [5]. These characteristics restrain the predictability of solely statistical models, especially in the long-term forecasting, and in times of market regime changes. Other assumptions made in the models include the stationarity which may be limiting in the real-world financial data.

Machine learning methods have significantly advanced stock price forecasting by addressing limitations of traditional models through better nonlinear pattern recognition and integration of diverse data sources [10, 11]. These models are now fundamental tools for both practitioners and researchers aiming at more accurate and adaptive stock price predictions. Machine learning algorithms like support vector regression (SVR) [12, 13, 14] and random forest (RF) algorithms are commonly used. Further, deep learning methods, especially recurrent neural networks (RNN) and long short-term memory (LSTM) networks, are highly popular for their ability to model sequential and temporal dependencies in stock price data [15, 16, 17, 18, 19].

Interval-valued data occur quite naturally in a number of situations where such data reflect uncertainty (as in confidence intervals), variability (minimum and maximum of daily temperature) etc. Interval-valued data have received various perspectives. The interval analysis field presupposes the observations and estimation of the real world being typically incomplete or imprecise and, therefore, do not reflect the actual data accurately. This field suggests that in the case of preciseness, data should be expressed in terms of intervals containing real quantities [20, 21].

Interval-valued stock price forecasting is an advanced method of predicting not only a single point estimate of what the stock prices will be in the future, but also a range or interval of what the stock prices will be within [22]. This approach is less biased as it captures the uncertainty and variability of stock price changes and gives a more complete information to make risk management and investment decision. The essence here is to model stock price data as intervals, or better described by lower and upper responses of price in a specified interval, and not as the values. This kind of interval data is a more realistic representation of the volatility and randomness inherent to markets in relation to point forecasts. This extra dimension assists investors and policymakers to know about possible price changes and the risks involved and it is especially helpful in a volatile or a very uncertain market [23].

v-support vector regression (VSVR) is a superior version of support vector regression (SVR) that aims at enhancing the trade-off between the complexity of the model and its predictive accuracy. It adds a parameter υthat directly regulates the number of support vectors and the margin of the regression model and hence provide a more flexible control over the sparsity and generalization capacity of the model [24, 25, 26]. VSVR is particularly helpful in the stock price forecasting context since it has a solid theoretical basis and is capable of working with nonlinear and noisy financial data. It employs the idea of support vectors to guess the underlying functional relationship between input features and the target stock price. Through the use of kernel functions, VSVR has the ability to project input information into a feature space of upper dimension and segment the space by a linear regression analysis, effectively trying to pick up nonlinear trends that are internal to stock price fluctuations [27, 28].

The VSVR computational efficient is highly reliant on a number of hyperparameters and effects that either have direct or indirect effects upon the optimal solution. The main contribution of our study is to enhance interval-valued forecasting accuracy for Iraqi stock market prices by integrating VSVR with metaheuristic algorithms for hyperparameter optimization, addressing VSVR's sensitivity issues.

2. Interval-valued time series construction

Interval-valued time series (ITS) represent time-ordered sequences where each observation is not a single point but an interval $[L_t, U_t]$ with lower bound, L_t, and upper bound, U_t, where $L_t \leq U_t$. This approach captures uncertainty and variability within each time period like stock price ranges [29]. In stock price modeling, an interval-valued variable, z, is defined as $[z_t] = \left\{ \left[z_t^L, z_t^U \right]^T : z_t^L, z_t^U \in \mathbb{R}, z_t^L \leq z_t^U \right\}$, in which $z_t^L$and z_t^U are the lower and upper stock price at the time t, respectively.

The center (mid-point), z_t^C, and the radius (half- range), z_t^R, of an interval-valued variable, are, respectively, calculated as

$$z_t^C = \frac{z_t^L + z_t^U}{2} \tag{1}$$

$$z_t^R = \frac{z_t^U - z_t^L}{2} \tag{2}$$

3. v-Support vector algorithm

SVM have been applied successfully in solving various classification problems. But, the SVM has been extended to deal with the nonlinear regression problems with the introduction of $\varepsilon - insensitive$loss function by Vapnik [30, 59, 60, 61, 62].

Given a training dataset of $n$observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i = (x_{i,1}, x_{i,2}, ..., x_{i,p}) \in \mathrm{R}^p$ represents a vector of the i^{th} feature, $y_i \in R$ for $i = 1, ..., n$ is the target variable, which is a quantitative variable, and $\varepsilon - insensitive$loss function, the SVR can be obtained through solving the following optimization problem

$$\begin{array}{l} \min\limits_{\mathbf{w},b} \left\{ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^n \left(\zeta_i + \tilde{\zeta}_i\right) \right\} \\ \text{S.T.} \quad \left\{ \begin{array}{l} y_i - (\mathbf{w} \bullet \varphi(\mathbf{x}_i) + b) \leq \varepsilon + \tilde{\zeta}_i \\ (\mathbf{w} \bullet \varphi(\mathbf{x}_i) + b) - y_i \leq \varepsilon + \zeta_i \\ \zeta_i, \tilde{\zeta}_i \geq 0, \end{array} \right. \end{array} \tag{3}$$

where $C > 0$ is a penalized parameter that controls the tradeoff between the model complexity and training error, ζ_i and $\tilde{\zeta}_i$ are slack variables, $\varphi(\mathbf{x}_i)$ is a nonlinear mapping which is induced by a kernel function, $\mathbf{w}$ is the weight vector and $b$is bias.

Then, Eq.(3) can be solved by the Lagrangian multipliers after reformulated it into its dual problem as

$$\begin{array}{l} \min\limits_{\tilde{\alpha},\alpha} \frac{1}{2} \sum_{i,j=1}^n (\tilde{\alpha}_i - \alpha_i)(\tilde{\alpha}_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) + \varepsilon \sum_{i=1}^n (\tilde{\alpha}_i - \alpha_i) - \sum_{i=1}^n y_i (\tilde{\alpha}_i - \alpha_i) \\ \text{S.T.} \quad \left\{ \begin{array}{l} \sum_{i=1}^n (\alpha_i - \tilde{\alpha}_i) = 0 \\ 0 \leq \alpha_i, \tilde{\alpha}_i \leq C, \end{array} \right. \end{array} \tag{4}$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$stands for kernel mapping, and α_i, $\tilde{\alpha}_i$are Lagrangian multipliers. The regression hyperplane for the underlying regression problem is then given by

$$y_i = f(\mathbf{x}_i) = \sum_{\mathbf{x}_i = \text{SV}} (\tilde{\alpha}_i + \alpha_i) K(\mathbf{x}_i, \mathbf{x}_j) + b, \tag{5}$$

where SV is the support vectors set.

The original problem in v-SVR leads to convex quadratic programming with inequality constraints as [27, 31, 32, 33, 34, 35, 36]

$$\begin{array}{l} \min\limits_{\mathbf{w},b} \left\{ \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\left[v\varepsilon + \frac{1}{n}\sum_{i=1}^n \left(\zeta_i + \tilde{\zeta}_i\right) \right] \right\} \\ \text{S.T.} \quad \left\{ \begin{array}{l} y_i - (\mathbf{w} \bullet \varphi(\mathbf{x}_i) + b) \leq \varepsilon + \tilde{\zeta}_i \\ (\mathbf{w} \bullet \varphi(\mathbf{x}_i) + b) - y_i \leq \varepsilon + \zeta_i \\ \zeta_i, \tilde{\zeta}_i \geq 0, \ \varepsilon \geq 0, \end{array} \right. \end{array} \tag{6}$$

Equation (4) can be solved by the Lagrangian multipliers after reformulated it into its dual problem as follows:

$$\begin{array}{l} L(\mathbf{w}, b, \varepsilon, \zeta, \tilde{\zeta}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\left(\nu\varepsilon + \frac{1}{n}\sum_{i=1}^n (\zeta_i + \tilde{\zeta}_i)\right) - \sum_{i=1}^n \theta_i\zeta_i - \sum_{i=1}^n \tilde{\theta}_i\tilde{\zeta}_i - \gamma\varepsilon \\ + \sum_{i=1}^n \alpha_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b - y_i - \varepsilon - \zeta_i) + \sum_{i=1}^n \tilde{\alpha}_i(\mathbf{w}^T\varphi(\mathbf{x}_i) + b - y_i - \varepsilon - \tilde{\zeta}_i), \end{array} \tag{7}$$

where $\alpha_i,\, \tilde{\alpha}_i,\, \theta_i,\, \tilde{\theta}_i, \gamma \geq 0$ are Lagrange multipliers. The solution of Eq. (5) can be achieved by partially differentiating with respect to ζ_i, $\mathbf{w}$, b, ε,and $\tilde{\zeta}_i$ as

$$\begin{cases} \frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} + \sum_{i=1}^{n} \alpha_i x_i - \sum_{i=1}^{n} \tilde{\alpha}_i x_i = 0 \\ \frac{\partial L}{\partial b} = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \tilde{\alpha}_i = 0 \\ \frac{\partial L}{\partial \varepsilon} = \frac{C}{n} \sum_{i=1}^{n} \nu - \gamma - \sum_{i=1}^{n} (\alpha_i + \tilde{\alpha}_i) = 0 \\ \frac{\partial L}{\partial \zeta} = \sum_{i=1}^{n} \frac{C}{n} - \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \theta_i = 0 \\ \frac{\partial L}{\partial \tilde{\zeta}} = \sum_{i=1}^{n} \frac{C}{n} - \sum_{i=1}^{n} \tilde{\alpha}_i - \sum_{i=1}^{n} \tilde{\theta}_i = 0 \end{cases} \Rightarrow \begin{cases} \mathbf{w} = \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) x_i \\ \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) = 0 \\ \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) = C\nu - \gamma \leq C\nu \\ \alpha_i = \frac{C}{n} - \theta_i \leq \frac{C}{n} \\ \tilde{\alpha}_i = \frac{C}{n} - \tilde{\theta}_i \leq \frac{C}{n} \end{cases} \tag{8}$$

Substituting Eq. (6) into Eq. (5), the Lagrange function can be rewritten as follows:

$$L = -\frac{1}{2} \sum_{i,j=1}^{n} (\tilde{\alpha}_i - \alpha_i)(\tilde{\alpha}_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i)\, y_i, \tag{9}$$

Based on the Karush–Kuhn–Tucker (KKT) conditions, the optimization problem in Eq. (7) can be solved by working with its dual form [24, 25]. After obtaining the optimal solution from the dual problem, the final decision function of the v-SVR model can be written as:

$$y_i \;=\; f(\mathbf{x}_i) = \sum_{i=1}^{n} (\tilde{\alpha}_i + \alpha_i)\, K(\mathbf{x}_i, \mathbf{x}_j) \;+ b\,. \tag{10}$$

4. The proposed improving

In SVR, several important settings—known as hyperparameters—must be chosen before the model can work properly. These include the penalty parameter, the -insensitive loss, and the kernel parameter. The performance of SVR depends strongly on how these hyperparameters are selected, but there is no exact mathematical method to determine the best values [37, 63, 64, 65]. Because of this, choosing suitable hyperparameters is a major part of SVR research [37, 38, 39, 40, 41, 42]. Many studies have tried different ways to improve SVR performance by selecting better hyperparameters, and several nature-inspired optimization algorithms have been used for this purpose [40, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52]. However, most of these methods focus only on tuning hyperparameters and do not perform feature selection at the same time.

The methods which are applicable to calculating the value of the hyper parameters include randomized search (RS), Bayesian optimization (BO), cross-validation (CV) and grid search (GS). Before this was done on all possible combinations of the hyper parameters and the combination of hyper parameters that yielded the best results on the selected criterion was returned. However, they are computationally costly and they are yet to exhaust all the combination of the hyperparameters [53].

Therefore, more efficient and superior means of hyperparameters optimization of v-SVR should be obtained. In the past several years, there has been a wide application of metaheuristic optimization algorithms to the problem of hyperparameter tuning [54].

Over the last few years, scientists proposed a variety of new nature-based algorithms to enhance and develop the scope of exploration and utilization of the existing algorithms. One of the most popular algorithms of its new algorithms is a coati optimization algorithm (COA) because it is very high-performin. [55].

COA is a population-based metaheuristic and the population members are known as coatis. The positions of each coati in a search space will influence the decisions of the decision variables. In accordance with this, in the COA setup, coatis are the solutions to the problem. At the start of the implementation, the coatis are randomly positioned in the search space as defined thereafter:

$$X_i : X_{i,j} = Lb_j + r * (Ub_j - Lb_j), \quad i = 1, 2, ..., N; j = 1, 2, ..., m \tag{11}$$

where $X_i$is the position of the ith coati in the search space, $X_{i,j}$ is the value of the jth decision variable of ith coati, $N$is the coati number, m is the number of decision variables, r is a random number in the range $[0, 1]$; and, finally,

$Lb_j$and Ub_jrepresent the lower bound and upper bound values of the jth decision variable, respectively. COA has two phases: exploration phase and exploitation phase.

One of the most significant things that are involved in this investigation process is that one-half of the coatis climb to the tree to terrify the iguana to the real level on the ground where other coatis are waiting to eat them. This kind of behavior proves the ability of this algorithm to explore or to investigate some aspects in a particular context. The best position is the favorable position of a member of the population of the iguana in COA. The behavior simulated by the algorithm follows the idea that half of the population hikes the trees in order to make the iguana fall, and the other half waits at the ground to grab the iguana when it falls, and it reads as described in the following Eq. (2). In the equation, r is a random number within a specified range of $[0,\ 1]$, I represents the iguana (i.e., the best individual in the population), I_j is the value of the jth decision variable of the iguana, and ϑ is an integer randomly selected from the set $\{1,\ 2\}$.

$$X_i^{new} : X_{i,j}^{new} = X_{i,j} + r * (I_j - \vartheta * X_{i,j}),\ i = 1, 2, ..., [N/2]\,; j = 1, 2......, m \tag{12}$$

Once the iguana drops to the ground, it is assigned a random position (I^G) within the search space, and the coatis under the tree adjust their position according to Eq. (3) and Eq. (4). If the objective function value at the new position calculated for each coati is an improvement over the current value, the new position is accepted. Conversely, if the new position yields a worse objective function value, the coati remains in its original position (Eq. 5).

$$I^G : I_j^G = Lb_j + r * (Ub_j - Lb_j), j = 1, 2, ..., m \tag{13}$$

$$X_i^{new} : X_{i,j}^{new} = \begin{cases} X_{i,j} + r * (I_j^G - \vartheta * x_{i,j}), F(I^G) < F(X_i) \\ X_{i,j} + r * (X_{i,j} - I_j^G), else \end{cases} \tag{14}$$

with $i = \left[\frac{N}{2}\right] + 1, \left[\frac{N}{2}\right] + 2, ..., N, j = 1, 2, ..., m$

$$X_i = \begin{cases} X_i^{new}, F(X_i^{new}) < F(X_i) \\ X_i, else \end{cases} \tag{15}$$

where $X_i$is the current position of the ith coati, $X_i^{new}$is the newly calculated position for the ith coati, $I^G$is the randomly generated position of iguana, F is the objective function value.

In the next phase, the exploitation phase, the behavior of coatis employing a predator-escape strategy is described. This strategy keeps the coati close to its current position and in a safe stance, enhancing the algorithm's exploitation capability. To accomplish this, a random position is created near each coati using Eq. (6) and Eq. (7). If this newly generated position results in an improved objective function value, the coati assumes the new position; or else, it retains its original position (Eq. (5)).

$$Lb_j^{local} = \frac{Lb_j}{t}, Ub_j^{local} = \frac{Ub_j}{t}, t = 1, 2, ..., T \tag{16}$$

$$X_i^{new} : X_{i,j}^{new} = X_{i,j} + (1 - 2 * r) * (Lb_j^{local} + r * (Ub_j^{local} - Lb_j^{local})), i = 1, 2, ..., N, j = 1, 2, .., m \tag{17}$$

where $X_i^{new}$is the new position calculated for the ith coati, t is the iteration number, $T$is the maximum iteration number.

In order to optimize the hyperparameters of VSVR with the improvement proposition of COA, the position vector X of every coati is established as a dimension D vector that signifies the position of the coati on the COA. As a result, the vectors X would be associated with a particular configuration of the RF, and the hyperparameters of the v-SVR are represented by the dimensions of X. Thus, three locations that each coati in the swarm will be searching will be located. Consequently, our proposed improving is as:

Step 1: The number of coatis, N_{coati}, is set to 35 and the maximum number of iterations is T= 1000.

Step 2: The positions of each coati are randomly specified. The three positions represent the hypermeters are randomly generated from uniform distribution as $C \sim U(0, 7)$, $\sigma \sim U(0, 5)$, and $\upsilon \sim U(0, 1)$.

Step 3: The fitness function is defined as

$$\text{fitness} = \min\ (\text{prediction error}). \tag{18}$$

Step 4: The positions of the coati are updated using Eq. (14) and Eq. (17), respectively.
Step 5: Steps 3 and 4 are repeated until a T is reached.

5. Prediction evaluation criteria

To measure the forecasting performance of the proposed approach, four evaluation criteria, namely, mean absolute error (MAE), root mean squared error (RMSE), direction accuracy (DA), and coefficient of determination (R^2), are selected in this study to verify the prediction accuracy of the proposed model [56, 57]. Their mathematical expressions are listed in Table 1.

Table 1. Prediction evaluation criteria

Evaluation criterion	Mathematical formula	Decision
MAE	$\frac{1}{n}\sum_{t=1}^{n}\lvert z_t-\hat{z}_t\rvert$	Lower value is better
RMSE	$\sqrt{\frac{1}{n}\sum_{t=1}^{n}(z_t-\hat{z}_t)^2}$	Lower value is better
DA	$\frac{1}{n}\sum_{t=1}^{n}q_t, \quad q_t = \begin{cases} 1, & \text{if } (z_{t+1}-z_t)(\hat{z}_{t+1}-\hat{z}_t)\geq 0 \\ 0, & \text{otherwise} \end{cases}$	Higher value is better
R^2	$1-\left[\sum_{t=1}^{n}(z_t-\hat{z}_t)^2 \Big/ \sum_{t=1}^{n}(\hat{z}_t-\bar{z}_t)^2\right]$	Larger value near to 1 is better

6. Experimental setting

6.1. Data description

In this study, the daily Iraqi stock prices of the national chemical and plastic industry (WSKB) company from January 15, 2020, to October 14, 2025, were utilized as sample data for the experiment (https://www.investing.com), as depicted in Figure 1. The training data set was composed of 637 daily prices from January 15, 2020, to December 31, 2023 and the remaining 308 daily prices from January 3, 2024, to October 14, 2025was reserved for the test set. In Figure 1, both the lower and the upper daily stock prices are depicted.

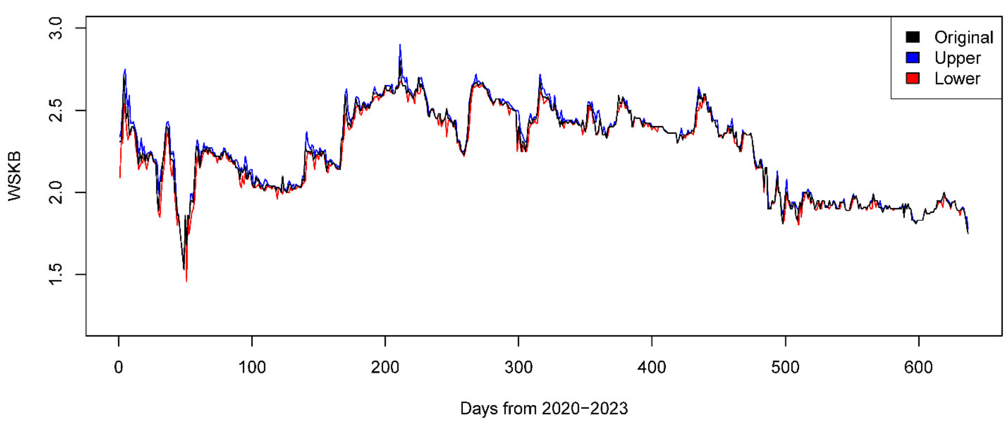

Figure 1. The daily stock price for the training data set of the WSKB.

6.2. Experimental results

The performance of our proposed algorithm, the COA-VSVR, in relation to the forecasting is examined by conducting comprehensive comparison tests based on the grid search strategy (GS-VSVR), and the cross-validation approach with ten folds (CV-VSVR). We consider the forecasts as one of the various approaches of the interval valued stock price as per the equation. Eq. (1) and Eq. (2). Table 2 to Table 5 contain the forecasting performances of the GS-VSVR, CV-VSVR and COA-VSVR model in terms of evaluation criteria.

Figure 1 shows how the price of WSKB Iraqi stock varies every day with time. One will not have a hard time concluding that the price of WSKB on a daily basis is non-linear and non-smooth, as well as highly erratic. Based on Tables 2 and 3 findings, one can demonstrate that the forecasting results of WSKB stock price suggest that the proposed method, COA-VSVR, could lead to a significant increase in accuracy and regarding generalization the contribution to forecasting as the method has smaller MAE, RMSE, higher DA and higher R2. It is evident that COA-VSVR is much better than these GS-VSVR and CV-VSVR methods. Based on Table 2, the reduction in terms of COA-VSVR compared to GS-VSVR, CV-VSVR is 12.71%, 93.31% and 14.52,89.63, respectively. The same applies to the testing data (Table 3) as the respective criteria reduced by 8.67% and 94.72, 22.91 and 92.19, respectively, than the GS-VSVR and CV-VSVR respectively. Even without data processing, meta-heuristic algorithms are very effective in prediction except the classical methods, GS-VSVR and CV-VSVR. The arbitrary hyperparameter options of CV and GS could cause VSVR performance. Moreover, in terms of comparing CV to GS as estimating the most appropriate hyperparameters of VSVR, the outcomes of the evaluation indicators in terms of training set, it can be stated that CV approach is better than GS approach.

Table 2. The prediction results of the used methods for the training set based on center interval-valued method

	CV-VSVR	GS-VSVR	COA-VSVR
MAE	0.181	2.365	0.158
RMSE	0.296	2.421	0.253
DA	0.608	0.392	0.665
R^2	0.836	0.632	0.965

Table 3. The prediction results of the used methods for the testing set based on center interval-valued method

	CV-VSVR	GS-VSVR	COA-VSVR
MAE	0.196	3.392	0.179
RMSE	0.358	3.536	0.276
DA	0.594	0.375	0.638
R^2	0.805	0.621	0.952

Tables 4 and 5 extend the center-based analysis to radius-based interval-valued time. COA-VSVR delivers the best training performance across all metrics. It has the lowest prediction errors MAE and RMSE, indicating more accurate radius forecasts. Further, COA-VSVR has highest DA (0.642) and means it is better at capturing ups and downs in the radius component. Additionally, it has the highest R^2 (0.957) shows it explains about 95.7% of the variance in the radius data. Moreover, CV-VSVR is competitive but clearly inferior to COA-VSVR, indicates cross-validation tuning is effective but does not reach the global optimum like COA.

Table 4. The prediction results of the used methods for the training set based on radius interval-valued method

	CV-VSVR	GS-VSVR	COA-VSVR
MAE	0.192	2.376	0.169
RMSE	0.307	2.432	0.264
DA	0.585	0.369	0.642
R^2	0.937	0.608	0.957

In Figures 2 - 5, it can be seen that the forecast of the model COA-VSVR outcome is basically the same as the actual price of WSKB stock price per day and this is showing that the quality of the model is of high nature when it comes to forecasting the prices. Moreover, the prediction ability of COA-VSVR is highly enhanced by

Table 5. The prediction results of the used methods for the testing set based on radius interval-valued method

	CV-VSVR	GS-VSVR	COA-VSVR
MAE	0.164	3.355	0.145
RMSE	0.324	3.501	0.242
DA	0.563	0.341	0.604
R^2	0.771	0.587	0.918

training and testing costs. This is mixed with comparatively smooth time series of the COA-VSVR. Consequently, the COA-VSVR model forecasts daily WSKB stock price e with high quality. Furthermore, the WSKB stock price prediction based on CV-VSVR are a bit accurate than the COA-VSVR. Conversely, the WSKB stock price forecast of GS-VSVR is ineffective in their forecasts in the long run.

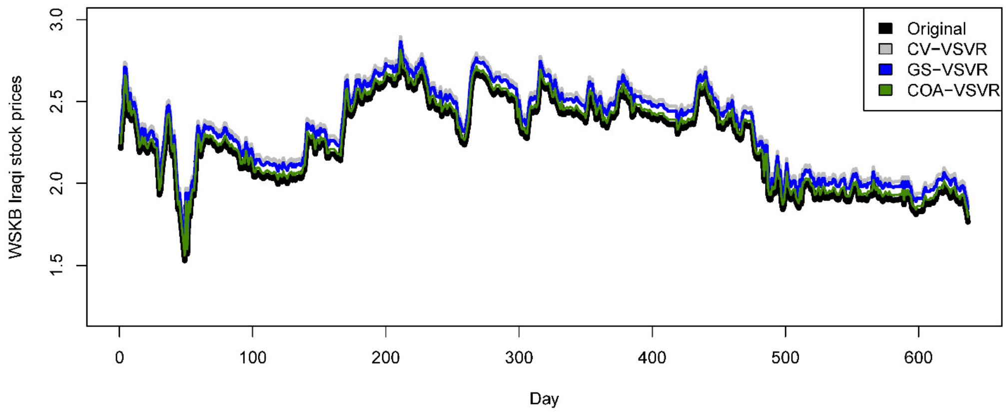

Figure 2. Prediction results in training dataset based on center interval-valued method.

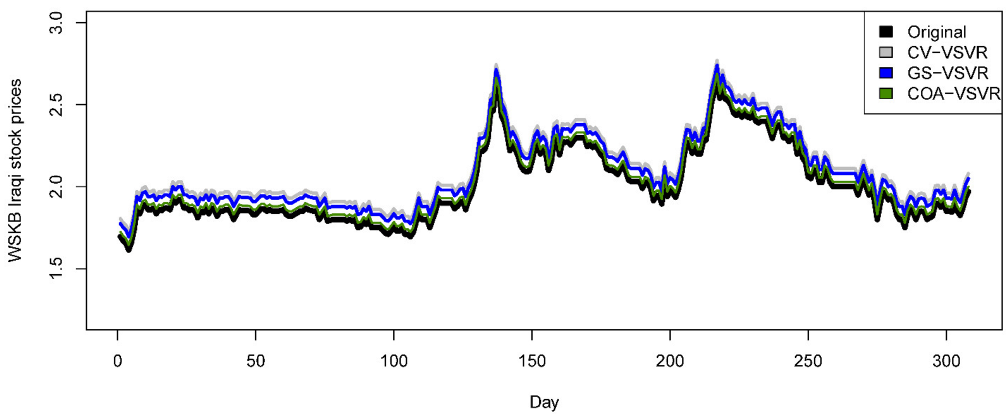

Figure 3. Prediction results in testing dataset based on center interval-valued method.

Among the center and the radius interval-valued methods, the center interval-valued method was the best option for predicting WSKB Iraqi stock prices, achieving the lowest MAE and RMSE for both training and testing dataset with a highest DA and R^2. To further highlight the forecast performance of the center and the radius interval-valued methods, the Diebold Mariano (DM) test [58] as a statistical test is performed to check their performances. The outcomes of the DM test for the training and testing datasets' forecasted future WSKB Iraqi stock prices are shown in Table 6. The COA-VSVR using center interval-valued is superior to the forecasted values provided by the COA-VSVR using radius interval-valued at least at a 95% confidence level for WSKB Iraqi stock prices prediction,

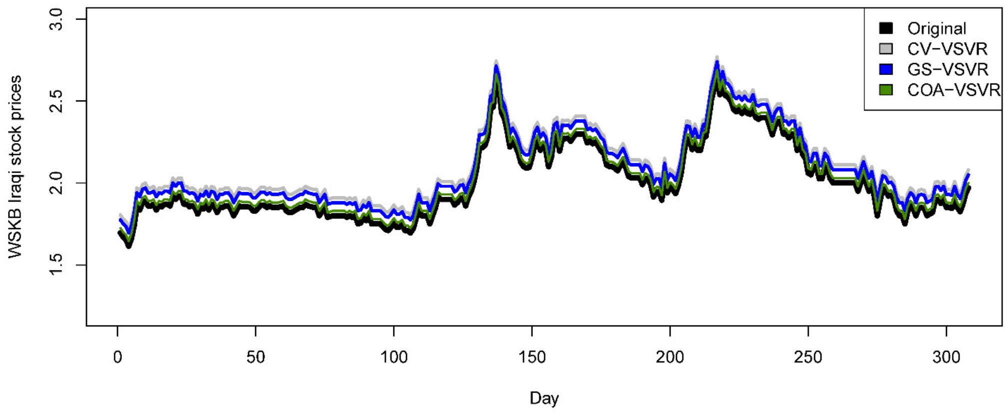

Figure 4. Prediction results in training dataset based on radius interval-valued method.

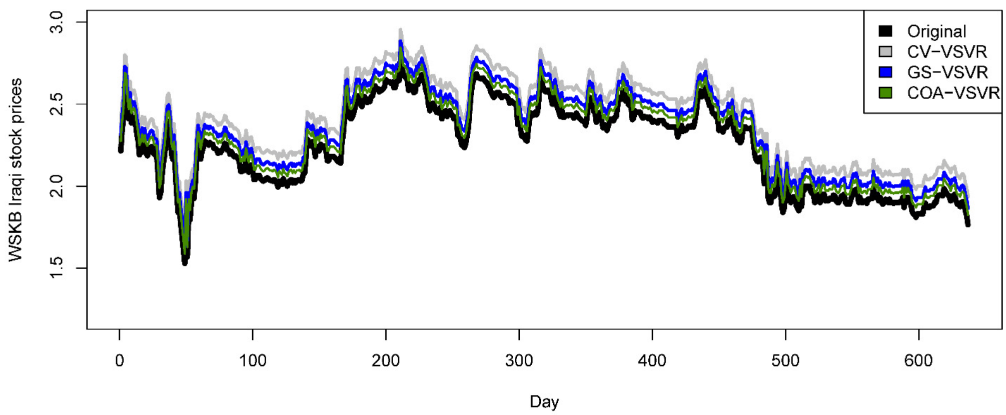

Figure 5. Prediction results in testing dataset based on radius interval-valued method.

according to the DM test, which indicates that when the COA-VSVR using center interval-valued is treated as the target approach, the p-values are less than the significance level of 5%. At least a 95% confidence level.

Table 6. DM test results for the COA-VSVR

	center interval-valued	COA-VSVR (train)	COA-VSVR (test)
radius interval-valued	COA-VSVR (train)	3.017 (p-value=0.0397)	
	COA-VSVR (test)		3.175 (p-value=0.0371)

7. Conclusion

The proposed interval-valued forecasting framework using VSVR with coati-based hyperparameter optimization provides a highly effective tool for modeling noisy, nonlinear dynamics in the Iraqi stock market. By constructing center and radius interval series from daily WSKB prices and jointly optimizing key VSVR hyperparameters, the model achieves markedly lower errors and higher explanatory power than conventional GS and CV strategies in both training and testing stages. The empirical results demonstrate that the COA–VSVR configuration consistently yields the smallest MAE and RMSE and the highest DA and R^2 for both center- and radius-based forecasts, confirming its strong generalization ability. Moreover, Diebold–Mariano tests indicate that center-based interval modeling is statistically superior to radius-based modeling at the 5% significance level, underscoring the practicality of center intervals for capturing price dynamics in this context. These findings highlight the promise of nature-inspired metaheuristics for hyperparameter tuning in advanced regression models and suggest that COA-VSVR can serve as a robust decision-support tool for risk-aware investors and policymakers in emerging markets. Future research may extend this work by incorporating additional financial and macroeconomic predictors, testing other interval construction schemes, and comparing COA-VSVR with recent deep learning–based interval forecasting approaches on broader market datasets.

REFERENCES

1. M. Darwish, E.E. Hassanien, and A.H.B. Eissa, *Stock Market Forecasting: From Traditional Predictive Models to Large Language Models*, Computational Economics, 2025.
2. W. Zhu, et al., *PMANet: a time series forecasting model for Chinese stock price prediction*, Scientific Reports, vol. 14, no. 1, p. 18351, 2024.
3. T.T. Thach, *Forecasting Stock Market Indices Using Integration of Encoder, Decoder, and Attention Mechanism*, Entropy, vol. 27, no. 1, 2025.
4. A.D. Kayit and M.T. Ismail, *Advancing stock price prediction through the development of hybrid ensembles: a comprehensive comparative analysis of machine learning approaches*, Journal of Big Data, vol. 12, no. 1, 2025.
5. P.H. Vuong, et al., *A bibliometric literature review of stock price forecasting: From statistical model to deep learning approach*, Science Progress, vol. 107, no. 1, p. 00368504241236557, 2024.
6. Q. Ge, *Enhancing stock market Forecasting: A hybrid model for accurate prediction of S&P 500 and CSI 300 future prices*, Expert Systems with Applications, vol. 260, p. 125380, 2025.
7. H. Mehtarizadeh, et al., *Stock price prediction with SCA-LSTM network and Statistical model ARIMA-GARCH*, The Journal of Supercomputing, vol. 81, no. 2, 2025.
8. S.S. Bagalkot, D.H. A, and N. Naik, *Novel grey wolf optimizer based parameters selection for GARCH and ARIMA models for stock price prediction*, PeerJ Computer Science, vol. 10, p. e1735, 2024.
9. J.K. Mutinda and A.K. Langat, *Stock price prediction using combined GARCH-AI models*, Scientific African, vol. 26, 2024.
10. T.B. Shahi, et al., *Stock Price Forecasting with Deep Learning: A Comparative Study*, Mathematics, vol. 8, no. 9, 2020.
11. C.-Y. Yang, et al., *Advancing Financial Forecasts: Stock Price Prediction Based on Time Series and Machine Learning Techniques*, Applied Artificial Intelligence, vol. 38, no. 1, 2024.
12. B.M. Henrique, V.A. Sobreiro, and H. Kimura, *Stock price prediction using support vector regression on daily and up to the minute prices*, The Journal of Finance and Data Science, vol. 4, no. 3, pp. 183–201, 2018.
13. A. Mahmoodi, et al., *A developed stock price forecasting model using support vector machine combined with metaheuristic algorithms*, Opsearch, vol. 60, no. 1, pp. 59–86, 2023.
14. R. Kuo and T.-H. Chiu, *Hybrid of jellyfish and particle swarm optimization algorithm-based support vector machine for stock market trend prediction*, Applied Soft Computing, vol. 154, p. 111394, 2024.
15. A. Staffini, *Stock price forecasting by a deep convolutional generative adversarial network*, Frontiers in Artificial Intelligence, vol. 5, p. 837596, 2022.
16. T. Kim and H.Y. Kim, *Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data*, PLoS One, vol. 14, no. 2, p. e0212320, 2019.
17. M. Nikou, G. Mansourfar, and J. Bagherzadeh, *Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms*, Intelligent Systems in Accounting, Finance and Management, vol. 26, no. 4, pp. 164–174, 2019.
18. 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.
19. S. Mukherjee, et al., *Stock market prediction using deep learning algorithms*, CAAI Transactions on Intelligence Technology, vol. 8, no. 1, pp. 82–94, 2023.
20. A.L.S. Maia, F.d.A. de Carvalho, and T.B. Ludermir, *Forecasting models for interval-valued time series*, Neurocomputing, vol. 71, no. 16-18, pp. 3344–3352, 2008.
21. Q. Lu, et al., *Forecasting interval-valued crude oil prices using asymmetric interval models*, Quantitative Finance, vol. 22, no. 11, pp. 2047–2061, 2022.
22. J. Wang, J. Liu, and W. Jiang, *An enhanced interval-valued decomposition integration model for stock price prediction based on comprehensive feature extraction and optimized deep learning*, Expert Systems with Applications, vol. 243, 2024.
23. D. Wang, et al., *Interval-valued financial time series prediction based on improved Elman neural network*, ICIC Express Letters, vol. 13, no. 2, pp. 159–166, 2019.
24. O.M. Ismael, O.S. Qasim, and Z.Y. Algamal, *Improving Parameters of V-Support Vector Regression with Feature Selection in Parallel by Using Quasi-Oppositional and Harris Hawks Optimization Algorithm*, Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, vol. 14, no. 2, pp. 113–118, 2024.
25. O.M. Ismael, O.S. Qasim, and Z.Y. Algamal, *Improving Harris hawks optimization algorithm for hyperparameters estimation and feature selection in v-support vector regression based on opposition-based learning*, Journal of Chemometrics, vol. 34, no. 11, 2020.

26. O.M. Ismael, O.S. Qasim, and Z.Y. Algamal, *A new adaptive algorithm for v-support vector regression with feature selection using Harris hawks optimization algorithm*, in Journal of Physics: Conference Series, IOP Publishing, 2021.
27. J. Wang, et al., *Improved v -Support vector regression model based on variable selection and brain storm optimization for stock price forecasting*, Applied Soft Computing, vol. 49, pp. 164–178, 2016.
28. T. Xiong, Y. Bao, and Z. Hu, *Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting*, Knowledge-Based Systems, vol. 55, pp. 87–100, 2014.
29. M. Jiang, et al., *A novel interval dual convolutional neural network method for interval-valued stock price prediction*, Pattern Recognition, vol. 145, 2024.
30. V.N. Vapnik, *An overview of statistical learning theory*, IEEE Transactions on Neural Networks, vol. 10, no. 5, pp. 988–999, 1999.
31. P.Y. Hao, *Pair- v -SVR: A Novel and Efficient Pairing nu-Support Vector Regression Algorithm*, IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 11, pp. 2503–2515, 2017.
32. D. Kong, et al., *Tool wear monitoring based on kernel principal component analysis and v-support vector regression*, The International Journal of Advanced Manufacturing Technology, vol. 89, no. 1-4, pp. 175–190, 2016.
33. N. Li, et al., *Force-based tool condition monitoring for turning process using v-support vector regression*, The International Journal of Advanced Manufacturing Technology, vol. 91, no. 1-4, pp. 351–361, 2016.
34. Y. Liu and G. Pender, *A flood inundation modelling using v-support vector machine regression model*, Engineering Applications of Artificial Intelligence, vol. 46, pp. 223–231, 2015.
35. X. Teng, H. Dong, and X. Zhou, *Adaptive feature selection using v-shaped binary particle swarm optimization*, PLoS One, vol. 12, no. 3, p. e0173907, 2017.
36. Y. Zhang and Y. Xie, *Forecasting of short-term freeway volume with v-support vector machines*, Transportation Research Record, vol. 2024, no. 1, pp. 92–99, 2007.
37. P. Tsirikoglou, et al., *A hyperparameters selection technique for support vector regression models*, Applied Soft Computing, vol. 61, pp. 139–148, 2017.
38. J.-S. Chou and A.-D. Pham, *Nature-inspired metaheuristic optimization in least squares support vector regression for obtaining bridge scour information*, Information Sciences, vol. 399, pp. 64–80, 2017.
39. R. Laref, et al., *On the optimization of the support vector machine regression hyperparameters setting for gas sensors array applications*, Chemometrics and Intelligent Laboratory Systems, vol. 184, pp. 22–27, 2019.
40. S. Li, H. Fang, and X. Liu, *Parameter optimization of support vector regression based on sine cosine algorithm*, Expert Systems with Applications, vol. 91, pp. 63–77, 2018.
41. V. Cherkassky and Y. Ma, *Practical selection of SVM parameters and noise estimation for SVM regression*, Neural Networks, vol. 17, no. 1, pp. 113–126, 2004.
42. K. Ito and R. Nakano, *Optimizing support vector regression hyperparameters based on cross-validation*, in Proceedings of the International Joint Conference on Neural Networks, IEEE, 2003.
43. C.-H. Wu, G.-H. Tzeng, and R.-H. Lin, *A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression*, Expert Systems with Applications, vol. 36, no. 3, pp. 4725–4735, 2009.
44. A. Kazem, et al., *Support vector regression with chaos-based firefly algorithm for stock market price forecasting*, Applied Soft Computing, vol. 13, no. 2, pp. 947–958, 2013.
45. M. Nait Amar and N. Zeraibi, *Application of hybrid support vector regression artificial bee colony for prediction of MMP in CO2-EOR process*, Petroleum, 2018.
46. C.-F. Huang, *A hybrid stock selection model using genetic algorithms and support vector regression*, Applied Soft Computing, vol. 12, no. 2, pp. 807–818, 2012.
47. C.-T. Cheng, et al., *Optimizing Hydropower Reservoir Operation Using Hybrid Genetic Algorithm and Chaos*, Water Resources Management, vol. 22, no. 7, pp. 895–909, 2007.
48. W.-C. Hong, et al., *SVR with hybrid chaotic genetic algorithms for tourism demand forecasting*, Applied Soft Computing, vol. 11, no. 2, pp. 1881–1890, 2011.
49. J. Cheng, J. Qian, and Y.-n. Guo, *Adaptive chaotic cultural algorithm for hyperparameters selection of support vector regression*, in International Conference on Intelligent Computing, Springer, 2009.
50. B. Üstün, et al., *Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization*, Analytica Chimica Acta, vol. 544, no. 1-2, pp. 292–305, 2005.
51. J. Zhang, et al., *Optimization enhanced genetic algorithm-support vector regression for the prediction of compound retention indices in gas chromatography*, Neurocomputing, vol. 240, pp. 183–190, 2017.
52. G. Cao and L. Wu, *Support vector regression with fruit fly optimization algorithm for seasonal electricity consumption forecasting*, Energy, vol. 115, pp. 734–745, 2016.
53. E. Abdulsaed, M. Alabbas, and R. Khudeyer, *Hyperparameter optimization for convolutional neural networks using the salp swarm algorithm*, Informatica, vol. 47, no. 9, 2023.
54. Z.Y. Algamal, et al., *Improving grasshopper optimization algorithm for hyperparameters estimation and feature selection in support vector regression*, Chemometrics and Intelligent Laboratory Systems, vol. 208, p. 104196, 2021.
55. M. Dehghani, et al., *Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems*, Knowledge-Based Systems, vol. 259, p. 110011, 2023.
56. H. Zhang, et al., *Forecasting monthly copper price: A comparative study of various machine learning-based methods*, Resources Policy, vol. 73, 2021.
57. X. Li, et al., *Forecasting the lithium mineral resources prices in China: Evidence with Facebook Prophet (Fb-P) and Artificial Neural Networks (ANN) methods*, Resources Policy, vol. 82, p. 103580, 2023.
58. F.X. Diebold and R.S. Mariano, *Comparing predictive accuracy*, Journal of Business and Economic Statistics, vol. 13, no. 3, pp. 253–263, 1995.
59. Qasim, M. K., Algamal, Z. Y., & Ali, H. M. *A binary QSAR model for classifying neuraminidase inhibitors of influenza A viruses (H1N1) using the combined minimum redundancy maximum relevancy criterion with the sparse support vector machine*, SAR and

QSAR in Environmental Research, vol. 29, no.(7), p.517-527, 2018
60. Qasim, O. S., & Algamal, Z. Y. *A gray wolf algorithm for feature and parameter selection of support vector classification* , International Journal of Computing Science and Mathematics, vol. 13, 93-102, 2021.
61. Al-Fakih, A. M., Qasim, M. K., Algamal, Z. Y., Alharthi, A. M., & Zainal-Abidin, M. H. *QSAR classification model for diverse series of antifungal agents based on binary coyote optimization algorithm* SAR and QSAR in Environmental Research, vol.34, no.4, p. 285-298, 2023.
62. Al-Fakih, A. M., Algamal, Z. Y., & Qasim, M. K. *An improved opposition-based crow search algorithm for biodegradable material classification* SAR and QSAR in Environmental Research, vol.33, no.5, p. 403-415, 2022.
63. Salih, A. M., Algamal, Z., & Khaleel, M. A. *A new ridge-type estimator for the gamma regression model*, Iraqi Journal for Computer Science and Mathematics, vol . 5 , no .1 , p. 85-98, 2023.
64. Naziyah, A. A., & Algamal, Z. Y. *Jackknifed Liu-type estimator in Poisson regression model* Journal of the Iranian Statistical Society, vol.19, no.1, p. 21-37, 2020.
65. Algamal, Z., & Ali, H. M. *An efficient gene selection method for high-dimensional microarray data based on sparse logistic regression* , Electronic Journal of Applied Statistical Analysis, vol . 10 , no . 1 , 242-256 , 2017.