Using Support Vector Machines to Predict Global Food Price Index

Shady I. Altelbany[1] and Anwar A. Abualhussein[2]

[1]*AL-Azhar University – Gaza*
[2]*AL-Azhar University - Gaza*

Abstract: *This research aims to Using Support Vector Machines (SVM) to predict global food price index, during the period from January 1990 to August 2020. The SVM model, with (Cost(C) = 10000, Epsilon(ε) = 0.1, gamma(γ) =90) has the lowest value of training error, with a small number of support vectors, is the best fit for monthly food price index predicting among all other SVM models with different parameter values.*

Keywords*: Support Vector Machines, Food Price Index, predicting, Training error.*

1. Introduction

The Food and Agriculture Organization's Food Price Index is a measure of the monthly change in international prices of a basket of food commodities. It consists of the average of five commodity group price indices: meat, dairy products, cereal, oils and fats, and sugar.

Global concerns have arisen over issues related to economic stability, poverty and food security following the increase of prices for agricultural and food commodities in international markets in 2008 and the resulting severe economic crisis. What exacerbated these concerns is the ongoing food price increase in the international markets of which a constant rise of food price has been a dominant characteristic as well as the forecasts of further long-term price increases in the coming years. The current and future food price increases add extra challenges to these stemmed from the global food crisis, underscoring the urgent need to exert appropriate efforts to avert their negative impact [3].

International food prices witnessed significant increases and fluctuations during the period from the beginning of the millennium to 2014. The FOA Food Price Index rose from 53.1 in 2000 to 131.9 in 2011 and then dropped but remained higher than 100 until 2014. The FOA Food Price Index averaged 95.5 in September of this year, up 79.1% compared to the beginning of the millennium [20].

Many countries pay great attention to food prices due to their importance for human life and their impact on people's welfare and poverty levels. Food expenditure comprises a large share of poor and low-income households' budget, and thus price increases affect the standard of living, increase the number of poor people and deepen impoverishment, posing a threat to political and security stability.

The accuracy of food price forecasts can assist in effective planning and decision-making to achieve food security for all countries. This entails an urgent need to devise a model that can forecast the future food prices accurately based on historical data. Forecasting future values based on time series data has recently received great attention in numerous and various research fields. Many techniques and methods have been developed to forecast the future behavior of phenomena. In the past, ARIMA model, a traditional forecasting approach, was used, but an alternative method known as Support Vector Machines (SVM) can be used to forecast the nonlinear time series and can overcome the problems of nonlinearity and instability [24]. This research aims to use SVM, an accurate and novel forecast method, to forecast the future food price index.

2. Support Vector Machines (SVM)

Support vector machines (SVM), introduced by Vapnik in the early 90's, has proven very effective computational tool in machine learning. SVM has already outperformed most other computational intelligence methodologies mainly because they are based on sound mathematical principles of statistical learning theory and optimization theory. It is a supervised learning algorithm. SVM has been applied successfully to a wide spectrum of areas, ranging from pattern recognition, text categorization, biomedicine, bioinformatics and brain–computer interface to financial time series forecasting [19].

It is usually implemented for classification problems but is also used for regression analysis. In SVM literature, when the SVM algorithm is used for classification problems, it is called Support Vector Classification (SVC) and when it is used for regression problems, it is called Support Vector Regression (SVR) [16]. With its ability to solve nonlinear regression estimation problems. In recent years, SVM appears to be an efficient tool which has been widely used for time series forecasting which have datasets coming from an unstable and nonlinear system such as the stock market ([23]; [22]; [24]).

The idea of SVM is to separate the dataset into a high-dimensional feature space and find the hyperplane that maximizes the margin. In regression problems, a linear learning machine learns nonlinear function in a kernel-induced feature space. It uses an implicit mapping of input data into a high dimensional feature space defined by a kernel function. Using a kernel function is useful when the data is far from being linearly separable [4]; [22]; [8]).

SVM usually offer better results than other methods, they have no problem with local minima (the big issue with Neural Nets), SVM don't require to specify many parameters as other methods do, usually the capacity and the Kernel to use and any parameter required by the kernel [25].

The dual formulation of the SVM using the method of Lagrange multipliers [9], makes it possible to perform a nonlinear transformation and a large margin separation in the high dimensional feature space [2].

The objective of SVM is to find a decision rule with good generalization ability through selecting some particular subset of training data, called support vectors. Training SVM requires the solving of a Quadratic Programming (QP) problem over a solution space known to be convex. Therefore, every local optima will also be a global solution. Hence, SVM training always finds a global solution that is usually unique [6]. This is superior to ANN, a technique that often optima ([5]; [32]; [26]).

Vapnik's SVM technique is based on the Structural Risk Minimization (SRM) principle [13]. The formulation embodies the SRM principle, which has been shown to be superior, [18] to traditional Empirical Risk Minimisation (ERM) principle, employed by conventional neural networks. SRM minimises an upper bound on the expected risk, as opposed to ERM that minimises the error on the training data [17]. SVM implements the SRM principle which minimizes the risk function consisting of the empirical error and a regularized term [7].

The remarkable characteristic of SVM is that it is not only destined for good classification but also intended for a better generalization of the training data. For this reason the SVM methodology has become one of the well-known techniques, especially for time series forecasting problems in recent years [1].

Another important characteristic of SVM is that here the training process is equivalent to solving a linearly constrained quadratic programming problem. So, contrary to other networks' training, the SVM solution is always unique and globally optimal. However a major disadvantage of SVM is that when the training size is large, it requires an enormous amount of computation which increases the time complexity of the solution [7].

3. The Theory of Support Vector Machines

In this section, a detailed presentation of the theory behind SVM equations is given, based on the formulation by [33]. Consider a given training set of N data points $\{X_i, Y_i\}_{i=1}^{N}$ with input data $X_i \epsilon \mathfrak{R}^P$, where P is the total number of data patterns and the output "desired value" is $Y_i \epsilon \mathfrak{R}$. Generally, the idea of building

SVM to approximate a function involves mapping the data x into a high-dimensional feature space via nonlinear mapping and performing a linear regression in the feature space ([35]; [23]; [34]).

Assume a non-linear function, $y(\mathrm{x})$ given by:

$$y(\mathrm{x}) = w^T\varphi(\mathrm{x}) + b \quad (1)$$

where, w is the weight vector, b is the bias or threshold, and $\varphi(x)$ represents a high-dimensional feature space that is nonlinearly mapped from the input space x [36].

The goal of SVM is to determine the values of w and b to orientate the hyperplane to be as far as possible from the closest samples [31]. The coefficients w and b are estimated by minimizing the following function:

$$Min\left(\frac{1}{2}w^T w\right) \quad (2)$$

subject to the following constraints:

$$\begin{cases} y_i - w^T\,\varphi(x_i) - b \le \varepsilon \\ w^T\,\varphi(x_i) + b - y_i \le \varepsilon \end{cases} \quad (3)$$

This gives:

$$y_i(w^T\,\mathrm{x}_i + b) - 1 \ge 0 \qquad \text{for } i = 1,2,\ldots\ldots,N \quad (4)$$

To estimate w and b, the above equation is transformed to the prime function below by introducing the positive slack variables ξ and ξ*, which represent the distance from actual values to the corresponding boundary values of ε – tube [34], see Fig.1:

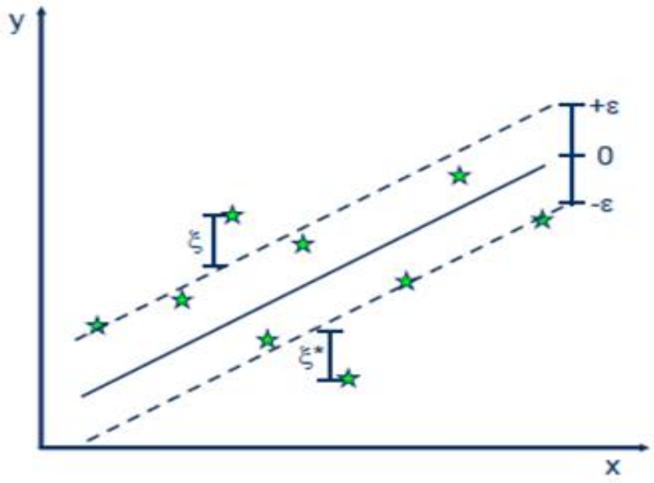

Fig.1: Illustration of Slack Variables and ε - tube in SVM

Minimize

$$R(w, \xi, \xi^*) = \frac{1}{2} W^T W + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \tag{5}$$

subject to the constraints:

$$\begin{cases} y_i - w^T x_i - b \le \varepsilon + \xi_i \\ w^T x_i + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{6}$$

The first term (1/2) $W^T W$ in Eq.(5) is the weights vector norm "the regularized term" whereas the second term $(C \sum_{i=1}^{N} (\xi_i + \xi_i^*))$ is called the empirical term and measures the ε-insensitive loss function. y_i is the desired value, and C "cost value" is referred to as the regularized constant, determining the trade-off between the empirical error and the regularized term. ε is called the tube size of SVM and is equivalent to the approximation accuracy placed on the training data points. Here, the slack variables ξ and ξ* are introduced to cope with possible infeasible optimization problems ([23]; [27]; [2]).

Eq.(5) assumes ε -insensitive loss function [33] defined as:

$$|\xi|_\varepsilon = \begin{cases} 0, & if \ |\xi| \le \varepsilon \\ |\xi| - \varepsilon, & otherwise \end{cases} \tag{7}$$

To solve the decision function given by Eq.(5), some Lagrangian multipliers $(\alpha_i, \alpha_i^*, \eta_i, \eta_i^*)$ are introduced in order to eliminate some of the primal variables. Hence, the Lagrangian of Eq.(5) is given as:

$$L = \frac{1}{2} W^T W + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) - \sum_{i}^{N} \alpha_i (\varepsilon + \xi_i - y_i + w^T x_i + b)$$

$$- \sum_i^N \alpha_i^* (\varepsilon + \xi_i^* + y_i - w^T x_i - b) - \sum_{i=1}^{N} (\eta_i \xi_i + \eta_i^* \xi_i^*) \quad \text{s.t} \quad \alpha_i, \alpha_i^*, \eta_i, \eta_i^* \ge 0$$
(8)

The Eq.(8) formulation permits the extension of SVM to nonlinear functions. It follows from the saddle point condition (the point where the primal objective function is minimal and the dual objective function is maximal) that the partial derivatives of L with respect to the primal variables (w, b, ξ_i, ξ_i^*) have to vanish for optimality [23].

Therefore,

$$\partial_b L = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0 \tag{9}$$

$$\partial_w L = w - \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) x_i = 0 \tag{10}$$

$$\partial_{\xi_i^{(*)}} L = c - \alpha_i^{(*)} - \eta_i^{(*)} = 0 \tag{11}$$

where (*) denotes variables with ξ and ξ^* superscripts. Substituting Eq.(9) and Eq.(11) into Eq.(8) lets the terms in b and ξ vanish. In addition, Eq.(11) can be transformed into, $\alpha \epsilon [0, C]$. Therefore, substituting Eqs.(9) to (11) into Eq.(8) yields the following dual optimization problem ([28]; [23]; [34]):

maximize

$$R(\alpha_i, \alpha_i^*) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) y_i - \varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*). K(x_i, x_j) \tag{12}$$

with the constraints:

$$\sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0$$

$$\sum_{i=1}^{N} \alpha_i = \sum_{i=1}^{N} \alpha_i^*, \tag{13}$$

$$0 \le \alpha_i^* \le c, \qquad i = 1,2, \dots, \mathrm{N}$$

$$0 \le \alpha_i \le c, \qquad i = 1,2, \dots, \mathrm{N}$$

The optimization problem is now a quadratic programming problem with linear constraints, which is easier to solve than Eq.(8) and ensures a unique global optimum. In deriving Eq.(12), the dual variables η_i, η_i^* were already eliminated through the condition in Eq.(11). Therefore, Eq.(10) can be rewritten as [23]:

$$w = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) x_i \tag{14}$$

Finally, the optimal decision hyperplane is obtained as:

$$y(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*). (\varphi(\mathrm{x}_i), \varphi(x)) + b \tag{15}$$

or more generally as:

$$y(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*). K(x_i, x) + b \tag{16}$$

where $K(x_i, x)$ is defined as the kernel function.

4. Fitting the SVM Model to the Data

This section focuses on fitting the SVM model described in Previous Section into the time series data of the monthly food price index from Jan. 1990 to Oug. 2020. The data had 368 observations, as shown in Fig.2. The time series data were divided into two parts. The first set included 332 Observations (90% of the series) in the period (Jan. 1990 – Oug. 2017), which were used for training—to train the SVM and to find its optimal parameters. The test set was composed of the remaining 36 Observations (10% of the series) in the period (Sep. 2017 – Oug. 2020), which was used to check the predictive power of SVM.

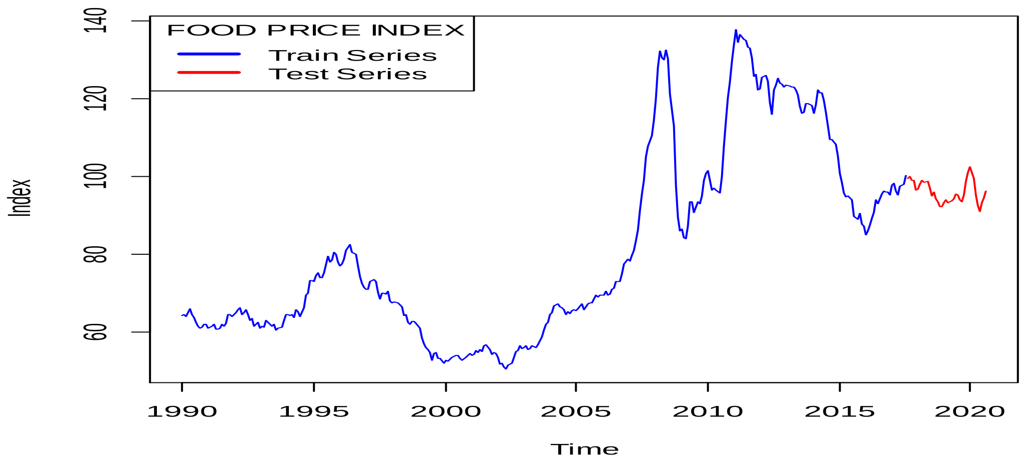

Fig.2: Time Series Plot of the Monthly food price index (1/1990-8/2020)

The “e1071” Package in R software helped us to analyse SVM. Furthermore, the "SVM" function in " e1071" library gives the SVM model. “e1071” has the following menu of choices to specify the type of kernel to be employed: Radial, Polynomial, Sigmoid, and Linear. The Radial Basis Function (RBF) was selected as the kernel function of the SVM models. This study employed the RBF kernel because of its good performance and advantages in time series forecasting problem proved in previous research ([12]; [14]; [34]). The RBF kernel maps nonlinear samples into a higher dimensional space to handle the nonlinearity problem which influences the complexity of model selection. RBF kernel has fewer numerical difficulties and less parameters [27].

For the implementation of the SVM, it is necessary to determine the approximate values of optimal hyper-parameters C, ε, and γ. The parameters of SVM were selected empirically by trying a finite number of values and keeping the values that reveal the maximum prediction accuracy. This procedure needs an exhaustive search over the search space to find the feasible region and feasible solution, which is a great challenge of SVM ([15]; [30]). The number of support vectors is automatically selected by the constrained minimization procedure [10]. The insensitive-loss function (ε) used in our experiments has a default value of 0.1 according to ([11]; [16]).

According to [29] and [21], the appropriate range of C is between 1 and 1000. However, in order to have a more thorough analysis, we extended the higher limit of the range for C to 10000. We searched the literatures for suggestions on appropriate range of γ. We found some previous studies; [24]where the searching range of γ was bounded by γ min = 0.0001 and γ max =100.

5. The Best Model of SVM

There are different SVM models with different types and parameters. Since there is no structured method for selecting the free parameters of SVM, ε was set to 0.1 and "type" to "eps-regression as default values for our experiments. To obtain the best SVM model for the monthly food price index, we attempted to fit all possible options of data models. All models then were compared using different criteria and the results are reported in Table 1. Training error and the number of support vectors with respect to the free parameters were investigated. The best model must have the smallest Training error and minimum number of support vectors. When having a small number of support vectors, SVM will have good generalization ability even in a very high-dimensional space.

Table 1: Several Models of SVM for the Monthly food price index

Model	C	γ	Number of Support Vectors	Training Error	Model	C	γ	Number of Support Vectors	Training Error
M1	1	0.0001	310	650.426665	M11	1	10	133	47.458317
M2		0.0005	308	512.640731	M12		20	102	22.363320
M3		0.001	284	422.007585	M13		30	93	17.979043
M4		0.005	260	341.373655	M14		40	92	15.409171
M5		0.01	265	327.079279	M15		50	86	12.547959
M6		0.02	265	287.299575	M16		60	85	10.855614
M7		0.05	273	256.696483	M17		70	80	8.768017
M8		0.1	287	316.369662	M18		80	81	7.342295
M9		0.5	238	92.867136	M19		90	71	6.015036
M10		1	218	78.362909	M20		100	72	5.304495
M21	10	0.0001	284	421.719589	M31	10	10	113	28.220420
M22		0.0005	261	342.704808	M32		20	98	17.641492
M23		0.001	266	330.043702	M33		30	94	14.344556
M24		0.005	271	318.391200	M34		40	80	10.044153
M25		0.01	269	307.081293	M35		50	69	6.394032
M26		0.02	271	262.909675	M36		60	67	5.126893
M27		0.05	290	212.959590	M37		70	66	4.733742
M28		0.1	280	282.172211	M38		80	66	4.568711
M29		0.5	223	79.374096	M39		90	65	4.513913
M30		1	193	73.362734	M40		100	66	4.200498
M41	100	0.0001	266	330.317700	M51	100	10	114	20.010884
M42		0.0005	276	321.873721	M52		20	97	16.318957
M43		0.001	275	319.317260	M53		30	83	9.867818
M44		0.005	268	291.561780	M54		40	73	5.713400
M45		0.01	276	285.188870	M55		50	70	4.694886
M46		0.02	288	221.152994	M56		60	61	4.539009
M47		0.05	288	128.484045	M57		70	66	4.239907

M48		**0.1**	**259**	**276.766114**	**M58**		**80**	**70**	**4.068735**
M49		**0.5**	**218**	**76.910618**	**M59**		**90**	**64**	**3.938326**
M50		**1**	**174**	**70.718714**	**M60**		**100**	**63**	**3.928498**
M61		**0.0001**	**277**	**321.844385**	**M71**		**10**	**104**	**18.486264**
M62		**0.0005**	**272**	**317.748212**	**M72**		**20**	**95**	**13.145726**
M63		**0.001**	**269**	**307.679543**	**M73**		**30**	**72**	**5.764451**
M64		**0.005**	**282**	**288.190599**	**M74**		**40**	**68**	**4.794641**
M65	**1000**	**0.01**	**290**	**278.269553**	**M75**	**1000**	**50**	**66**	**4.315311**
M66		**0.02**	**298**	**139.966976**	**M76**		**60**	**64**	**4.167988**
M67		**0.05**	**252**	**104.886269**	**M77**		**70**	**67**	**4.027690**
M68		**0.1**	**248**	**245.335348**	**M78**		**80**	**62**	**3.997039**
M69		**0.5**	**195**	**71.193581**	**M79**		**90**	**64**	**3.781844**
M70		**1**	**175**	**74.320877**	**M80**		**100**	**61**	**3.505355**
M81		**0.0001**	**277**	**319.062836**	**M91**		**10**	**103**	**16.858672**
M82		**0.0005**	**268**	**292.418037**	**M92**		**20**	**91**	**9.866294**
M83		**0.001**	**274**	**288.758910**	**M93**		**30**	**71**	**5.035019**
M84		**0.005**	**290**	**276.737246**	**M94**		**40**	**66**	**4.331561**
M85	**10000**	**0.01**	**300**	**243.678243**	**M95**	**10000**	**50**	**70**	**4.204117**
M86		**0.02**	**270**	**110.434560**	**M96**		**60**	**63**	**3.958627**
M87		**0.05**	**257**	**100.008395**	**M97**		**70**	**63**	**4.033741**
M88		**0.1**	**237**	**187.917380**	**M98**		**80**	**65**	**3.690653**
M89		**0.5**	**183**	**69.587086**	**M99**		**90**	**59**	**3.326208**
M90		**1**	**175**	**73.785007**	**M100**		**100**	**60**	**3.386194**

In Table 2, M100: $\gamma = 90$, $C = 10000$, and $\varepsilon = 0.1$ were concluded as the best selections for our experiment, because they have the smallest Training error and minimum number of support vectors.

Table 2: The Best Model of SVM

Model	C	γ	Number of Support Vectors	Training Error
M20	1	100	72	5.304495
M40	10	100	66	4.200498
M60	100	100	63	3.928498
M80	1000	100	61	3.505355
M100	10000	90	59	3.326208

From Fig.3, we can also observe that the fitted values are approximately equal to the actual values. These parameters were used to train the model again and then to predict the test set.

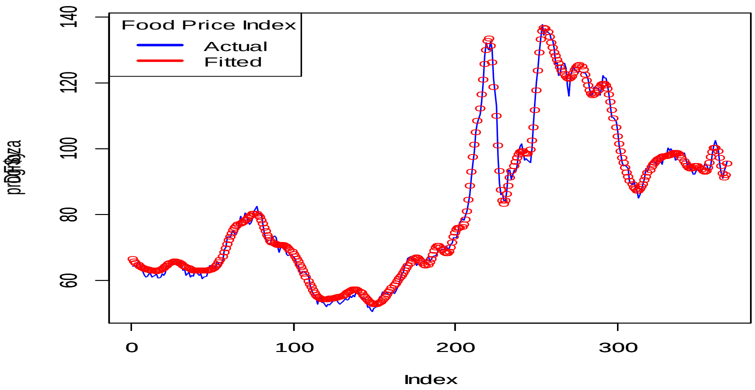

Fig.3: Actual and Forecast results by SVM Method When (γ = 90, C= 10000)

6. Summary

The SVM model, with (Cost(C) = 10000, Epsilon(ε) = 0.1, gamma(γ) =90) has the lowest value of Training error, with a small number of support vectors, is the best fit for monthly food price index predicting among all other SVM models with different parameter values.

The SVM model offers important advantages over other methods, such as having a smaller number of free parameters and producing more accurate forecasts. Although there is little effect on the generalization error with respect to the free parameters of SVM.

References

1- Adhikari, R., & Agrawal, R. K. (2013). An introductory study on time series modeling and forecasting. arXiv preprint arXiv:1302.6613.

2- Alamili, M. (2011). Exchange rate prediction using support vector machines: a comparison with artificial neural networks . Master Thesis. Faculty of Technology, Policy and Management. Delft University Of Technology.

3- Badwy, A. (2013). The implications of global food price increases for the economies of Arab countries. Arab Monetary Fund, UAE.

4- Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. Neural Information Processing-Letters and Reviews, 11(10), 203-224.

5- Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. Data mining and knowledge discovery, 2(2), 121-167.

6- Burges, C. J., & Crisp, D. J. (2000). Uniqueness of the SVM solution. In Advances in Neural Information Processing Systems (pp. 223-229).

7- Cao, L. J., & Tay, F. E. (2003). Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on neural networks, 14(6), 1506-1518.

8- Chen, T. T., & Lee, S. J. (2015). A weighted LS-SVM based learning system for time series forecasting. Information Sciences, 299, 99-116.

9- Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

10- Czermiński, R., Yasri, A., & Hartsough, D. (2001). Use of support vector machine in pattern classification: Application to QSAR studies. Quantitative Structure-Activity Relationships, 20(3), 227-240.

11- Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., & Weingessel, A. (2008). Misc functions of the Department of Statistics (e1071), TU Wien. R package, 1, 5-24.

12- Ding, Y., Song, X., & Zen, Y. (2008). Forecasting financial condition of Chinese listed companies based on support vector machine. Expert Systems with Applications, 34(4), 3081-3089.

13- Eidenberger, H. (2014). Categorization and Machine Learning: The Modeling of Human Understanding in Computers. BoD–Books on Demand.

14- Eslamian, S. S., Gohari, S. A., Biabanaki, M., & Malekian, R. (2008). Estimation of monthly pan evaporation using artificial neural networks and support vector machines. J Appl Sci, 8(19), 3497-3502.

15- Friedrichs, F., & Igel, C. (2005). Evolutionary tuning of multiple SVM parameters. Neurocomputing, 64, 107-117.

16- Garg, A. (2012). Forecasting exchage rates using machine learning models with time-varying volatility. Master Thesis. Linköping University.

17- Gunn, S. R. (1998). Support vector machines for classification and regression. ISIS technical report, 14(1), 5-16.

18- Gunn, S. R., Brown, M., & Bossley, K. M. (1997). Network performance assessment for neurofuzzy data modelling. In International Symposium on Intelligent Data Analysis (pp. 313-323). Springer, Berlin, Heidelberg.

19- Jayadeva, Khemchandani, R., & Chandra, S. (2017). Twin Support Vector Machines: Models, Extensions and Applications. Springer International Publishing.

20- Jelidan, S. (2020). Developments in global food prices. Economic Newspaper, Saudi Arabia.

21- Meyer, D.(2018). Support Vector Machines—the Interface to libsvm in package e1071. FH Technikum Wien, Austria. David.Meyer@R-Project.org.

22- Msiza, I.S., Nelwamondo, F.V., & Marwala, T. (2007). Artificial Neural Networks and Support Vector Machines for water demand time series

forecasting. 2007 IEEE International Conference on Systems, Man and Cybernetics, 638-643.
23- Ojemakinde, B. T. (2006). Support Vector Regression for Non-Stationary Time Series. Master Thesis. University of Tennessee – Knoxville.
24- Okasha, M. K. (2014). Using support vector machines in financial time series forecasting. International Journal of Statistics and Applications, 4(1), 28-39.
25- Parrilla Gutiérrez, J. M. (2010). Support Vector Machines. Similarity functions to work with heterogeneous data and classifying documents. . Master Thesis, Syddansk universitet.
26- Raicharoen, T., Lursinsap, C., & Sanguanbhokai, P. (2003). Application of critical support vector machine to time series prediction. In Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on (Vol. 5, pp. V-V). IEEE.
27- Samsudin, R., Shabri, A., & Saad, P. (2010). A comparison of time series forecasting using support vector machine and artificial neural network model. Journal of applied sciences, 10(11), 950-958.
28- Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and computing, 14(3), 199-222.
29- Suganyadevi, M. V., & Babulal, C. K. (2014). Support vector regression model for the prediction of loadability margin of a power system. Applied Soft Computing, 24, 304-315.
30- Tharwat, A., & Hassanien, A. E. (2018). Chaotic antlion algorithm for parameter optimization of support vector machine. Applied Intelligence, 48(3), 670-686.
31- Tharwat, A., & Hassanien, A. E. (2019). Optimizing Support Vector Machine Parameters Using Bat Optimization Algorithm. In Machine Learning Paradigms: Theory and Application (pp. 351-374). Springer, Cham.
32- Vapnik, V.N (1998). Statistical learning theory. Wiley, New York.
33- Vapnik, V.N. (1995). The nature of statistical learning theory. Springer. New York.
34- Wang, W. C., Chau, K. W., Cheng, C. T., & Qiu, L. (2009). A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. Journal of hydrology, 374(3-4), 294-306.
35- Yang, H., Huang, K., Chan, L., King, I., & Lyu, M. R. (2004). Outliers treatment in support vector regression for financial time series prediction. In International Conference on Neural Information Processing (pp. 1260-1265). Springer, Berlin, Heidelberg.
36- Zhang, L., Zhou, W. D., Chang, P. C., Yang, J. W., & Li, F. Z. (2013). Iterated time series prediction with multiple support vector regression models. Journal Neurocomputing, 99, 411-422.