

Financial time series forecasting using Simulated Annealing and Support Vector Regression

Juan Javier González-Mancha¹, Juan Frausto-Solís¹, Guadalupe Castilla Valdez¹, Jesús David Terán-Villanueva¹, y Juan Javier González Barbosa¹.

¹Tecnológico Nacional de México: Instituto Tecnológico de Ciudad Madero, México.

*jjgmancha@outlook.com, juan.frausto@itcm.edu.mx, gpe_cas@yahoo.com.mx,
david_teran01@yahoo.com.mx, jjgonzalezbarbosa@hotmail.com*

Abstract. The behavior of stock exchanges around the world has an important role in the financial development of countries. The Mexican Stock Exchange (BMV, for its acronym in Spanish) is the financial entity in Mexico where investors build and manage investment portfolios to generate profit. Financial time series forecasting is an important problem for investment portfolios creation and optimization because it allows them to have a prediction of the value of investment assets over time, therefore reducing at some extent the uncertainty of these operations. Classical models, such as ARIMA, artificial neural networks and support vector machines, are usually used as forecasting tools. In this paper, we forecast financial time series of the BMV using a hybrid parallel algorithm that employs both simulated annealing (SA) and support vector machines for regression (SVR). Finally, the resulting Mean Absolute Percent Error (MAPE) of this hybrid algorithm is compared with that of an ARIMA model.

Keywords: Simulated Annealing, Financial Portfolios, Parallel Algorithm, Financial Forecast

1. Introduction

Stock exchange's growth around the world has an important role in improving countries' economies. Stock brokers and traders aim to generate profit by buying and selling shares [1]. A common strategy for investors is to diversify their assets distributing them in stock portfolios [2], [3]. To build a stock portfolio (SP), it is useful to know the historic behavior of the market to try to forecast its future behavior. The ratio of investment for each stock is then determined through this forecasting. Commonly SPs use the Markowitz quadratic optimization model [4] or its extensions to forecast. This model aims to minimize the risk and maximize the profit even if using multi-objective methods [2], [5].

To build a stock portfolio in an efficient way, it is important to ground the forecast of the closing prices and volatilities for each one of the stocks based on historical data and using the more precise methods

Nowadays ARIMA and GARCH are the most used methods for forecasting time series; however, there are other methods that have shown better results. Recently some methods from computational intelligence have appeared for this area; particularly, Artificial Neural Networks (ANN) [6] are becoming very popular. In the case of financial time series from the BMV, ARIMA and GARCH are commonly used; however, sometimes these methods obtain very bad results. For solving this problem, an alternative is to hybridize classical methods with ANN. Nevertheless, ANN methods have the problem of getting stuck in local optima. In addition, to determine the ANN parameters, particularly those of the hidden layers, is an open issue. Thus, we cannot have confidence in the classical methods and neither in ANN. Fortunately, heuristic and machine learning methods developed to solve difficult NP-hard problems are available since many years ago. Recently some of them were adapted to forecast time series. Particularly, SVR, genetic algorithms and simulated annealing (SA) have shown to be very successful for financial time series [7]–[9].

SVR parameters are usually tuned with different methods, such as grid search. This method could get stuck at a local optimum and has exponential computational cost [1]. This paper introduces a hybrid parallel algorithm for the forecasting of financial time series of the BMV, which is named SAP-SVR (SA algorithm with Parallel SVR). This parallel algorithm has a sequential phase where SA algorithm is firstly executed. The purpose of SA is to tune the parameters of an SVR algorithm which is executed in a parallel second phase. The proposed SAP-SVR algorithm is compared in this work with the classical ARIMA method. The paper is organized as follows. Section two describes related works that use classical forecasting methods and new heuristic and machine learning algorithms of financial time series forecasting. In section three SAP-SVR algorithm is introduced. Finally, section four and five present the experimental results and conclusions respectively.

2. Related work

Financial time series forecasting methods can be classified in two groups. The first group is composed by the classical methods, based on traditional economic models such as ARIMA and GARCH. In the second group are those from machine learning and heuristic algorithms. Among the classical methods, the most commonly used in the BMV are ARIMA and GARCH. Machine learning methods are those using a learning strategy, such as Artificial Neural Networks (ANN) and SVR. Finally, heuristics are approximate methods where the most common are Genetic Algorithms (GA) and Simulated Annealing (SA) which incorporate strategies based on empirical knowledge to improve its performance. In this section, we describe both group of methods.

In this section, we briefly show some of the commonly used methods and other works that have taken different approaches to time series forecasting.

2.1 Classical methods

Autoregressive integrated moving-average models (ARIMA) are described in terms of p , d and q parameters, denoted as ARIMA(p,d,q). These three parameters are associated with three different processes that are used in the model: autoregression (AR), integration (I), and the moving average (MA). These models are appropriated if the time series observations are statistically dependent or if they are related to each other[10]. ARIMA(p,q) is the combination of models AR and MA, which is formulated as follows [10]:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t - w_1 \epsilon_{t-1} - w_2 \epsilon_{t-2} - \cdots - w_q \epsilon_{t-q}$$

ARIMA(p,d,q) model offers the potential to adjust models that could not fit correctly using AR and MA models separately.

During the past three decades ARIMA has been used as a model in time series forecasting [11] although recent researches have shown that methods such as ANN and SVR are viable alternatives [12], ARIMA is still used in different forecasting methods and for comparison purposes.

Generalized Autoregressive conditional heteroscedasticity (GARCH) [13] is used when time series have very high volatility variations over time; in these cases heteroscedasticity is present. GARCH improves ARCH model, as it expresses variance as an ARIMA process. Nevertheless, it is known than GARCH has a poor performance [7].

GARCH models similarly to ARIMA models, are still used to forecast time series, but in recent researches they are mostly used for comparison purposes and developing hybrid methods [14], [15].

2.2 Machine learning methods

Support Vector Machines (SVM) [16], are supervised machine learning methods developed for classification and regression. The SVM are a tool for prediction, which uses machine learning theory to maximize its precision. In addition, the SVM have the property of avoiding data overfitting [17].

SVM tools were initially developed to solve classification problems, and later they were used for regression problems, keeping the main characteristics of the SVM. This approach is known as Support Vector Machines for Regression (SVR) [18].

When data do not have a linear separability SVMs map the data to a higher dimensional space called *feature space*, this mapping results from a mathematical function called *kernel trick* [19].

2.3 Heuristic methods

A common method for time series forecasting are Artificial Neural Networks (ANN). Khashei and Bijari [12] proposed a hybrid model of ANN using ARIMA models in order to yield a more accurate forecasting than traditional Artificial Neural Networks. This model was tested using the Wolf's sunspot, the Canadian lynx, and the British pound/United States dollar exchange rate data. Also a Bayesian regularized ANN for stock market forecasting was implemented by Ticknor in which experiments were performed with data from NASDAQ [20], specifically Microsoft Corp and Goldman Sachs Group stocks. The results show that the model performed well in comparison to other models as ARIMA and Hassan's model [21].

On the other hand, a hybrid models based on SVR for forecasting time series was proposed by Kazem [2]. The Kazem's model tunes the SVR parameters using a chaos-based firefly algorithm (SVR-CFA). The performance of this implementation is compared with different models using SVR such as: SVR with genetic algorithm (SVR-GA), SVR with chaos-based genetic algorithm (SVR-CGA), SVR with firefly algorithm (SVR-FA); and ANN. This experimentation was done using NASDAQ's stocks such as Intel's, National Bank's and Microsoft's, obtaining better results with SVR-CFA than ANN as it is shown in Figure 1.

Figure 1. Kazem's models performance under MAPE.

Model	Intel	NationalBank shares	Microsoft
SVR-GA	0.047385	0.049147	0.066795
SVR-CGA	0.046709	0.045997	0.061031
SVR-FA	0.045626	0.047267	0.052653
SVR-CFA	0.044594	0.045847	0.051907
ANN	0.047088	0.046742	0.054386
ANFIS	0.04635	0.047982	0.056344

Even though Kazem showed that SVRs have good results, this research was not focused on forecasting assets from BMV. Also, the main issues in this research are related to the parameters tuning of SVR, and the selection of the kernel; in other words, the best Kernels performance is not really known. Because of this, in this work the proposed algorithm uses a SVR algorithm automatically tuned by an SA algorithm. This algorithm forecast close prices of assets, particularly, from the BMV. Later in Section 4 the experimentation to assess the performance of the proposed SAP-SVR algorithm is presented.

3. SAP-SVR proposed algorithm

In this section, a hybrid parallel algorithm SAP-SVR that uses classic Simulated Annealing (SA) and a parallel SVR is described. SVR by itself can forecast close prices of financial time series. However, SVR accuracy cannot be guarantee if its parameters are not correctly tuned. As was mentioned before, in the proposed algorithm SA tunes the SVR parameters.

The purpose of SA is to find the best results for SVR in the time series forecasting; to achieve this it seeks to minimize the forecast error by modifying the SVR parameters. The SVR main function is to return a model that can forecast accurately the time series of close prices. The algorithm SAP-SVR is presented in Figure 2. Initially, the algorithm receives an initial and final temperature (c_i, c_f , respectively), a parameter α that states the constant drop in temperature; L that defines the Markov chain length, and finally the time series data is split into a test and train sets. Line two generates the best random parameters for SVR; they are updated in S_{best} variable. In line three, an initial SVR model is generated using S_{best} and train data set of the time series. This model is then optimized in a SA method; line 4 to line 19 is the annealing loop, followed by the inner loop (line 4 to 17). The old S_{best} model is updated by S_{new} when the latter has a lower error than the former; this is measured (lines 8 to 10) by the classical MAPE measure used in forecasting [23]; on the contrary, Boltzmann distribution determines if the new solution is accepted. Line 18 is the geometric function to determine the new temperature. Finally, the algorithm returns the best SVR model in the variable S_{best} estimating the close prices of the series for several periods in the future.

In SAP-SVR, the solution obtained by SA is the set of the parameters (S_{best}) that will tune the SVR. To determine the regression equation, LIBSVM [24] is called to calculate the current energy of SA. This energy represents the mean absolute percentage error (MAPE) which is commonly used in financial forecasting [1]. Real numbers are used to represent the SVR parameters, and to explore the neighborhood the polynomial mutation is used [25].

Figure 2. SAP-SVR algorithm.

SAP-SVR Algorithm	
1	Initialize ($c = c_i, c_f, S_{best}, \alpha, L, test, train$)
2	Generate S_{best}
3	$model_{best} = SVR(train, S_{best})$
4	Do while $c > c_f$
5	Repeat L times
6	Generate S_{new}
7	$model_{new} = SVR(train, S_{new})$
8	If $MAPE(S_{new}, model_{new}) < MAPE(S_{best}, model_{best})$
9	$S_{best} = S_{new}$
10	$model_{best} = model_{new}$
11	Else
12	If $random < \exp(\frac{MAPE(S_{best}, model_{best}) - MAPE(S_{new}, model_{new})}{c})$
13	$S_{best} = S_{new}$
14	$model_{best} = model_{new}$
15	End
16	End
17	End
18	$c = \alpha * c$
19	End

4. Experimental results

Financial time series forecasting was done using stocks from the BMV, one of the main reasons is that these series have been barely studied in recent researches, and FEMSA and CEMEX had the highest capitalization value during 2009 [26]. These series were GCARSO, FEMSA, LIVEPOL and CEMEX. Microsoft from NASDAQ was used for comparison purposes. Data was obtained from Yahoo Finance website [27], using close prices data from January 1, 2000 to August 8, 2016 for GCARSO, LIVEPOL and CEMEX; January 6, 2003 to April 4, 2013 for FEMSA; and the same dates given in Kazem's work for Microsoft [22]. These series were chosen because their available data is very large, going from 10 to 16 years of data. In addition, these assets are very robust financial companies so experimentation in the future could be performed. In fact, they also trade in NASDAQ and NYSE. Finally, the proposed method can be compared against the classical methods

SAP-SVR uses a geometric cooling function which includes some configuration parameters in SA, such as the initial and final temperature, and the cooling parameter. These values were determined by SA analytical tuning method. There were developed eight different variants of SAP-SVR depending on which kernel was utilized.

For each time series, 90% of the data were used for training and 10% for testing, except for Microsoft tests, where 80% of data were used for training as was done in [22]. All the tests were executed on the Ehecatl cluster terminal of the Instituto Tecnológico de Ciudad Madero. The cluster had the following specifications: Intel® Xeon® 2.30 GHz, RAM: 64gb (4x16gb) ddr4-2133, Linux CentOS.

Tables 1 to 5 show the results obtained for each variant of SAP-SVR applied to five BMV time series. These tables present a comparison of different kernels applied in SAP-SVR. The first column specifies the SAP-SVR variant, second and third column show the average MAPE and time respectively, finally fourth and fifth column show the minimum MAPE and the minimum time obtained for each SAP-SVR variant.

Results obtained for GCARSO can be seen in Table 1. The best results for average MAPE were Epsilon-Linear, Nu-Linear and Nu-Sigmoidal, but only Epsilon-Linear and Nu-linear had a significant minimum average time. Considering the minimum MAPE every variant had a similar performance; however, the worst was Epsilon-Polynomial, the minimum time was obtained by Epsilon-Linear and Nu-Linear variants. Therefore, it can be stated that linear kernels have a good performance considering performance in solution quality and running time.

Table 1. Results obtained by SAP-SVR for GCARSO

Variant SAP-SVR	Avg MAPE (%)	Avg Time(sec)	Minimum MAPE (%)	Minimum time (sec)
Epsilon-Linear	3.41091	16.53922222	3.41	15.61
Epsilon-Polynomial	3.786174444	18.95222222	3.4144	17.59
Epsilon-Radial	4.043787778	23.5673333	3.39692	20.905
Epsilon-Sigmoidal	3.403532222	29.86122222	3.39353	28.33
Nu-Linear	3.402051111	14.73211111	3.4	14.03
Nu-Polynomial	4.197105556	19.261	3.4085	18.341
Nu-Radial	4.659482222	19.83422222	3.39504	18.817
Nu-Sigmoidal	3.40268	30.10444444	3.39456	28.86

Overall, for experimental tests, assets from BMV were mainly chosen because of the availability to retrieve data from several previous years. In the tests we preferred assets that had enough data starting from January, 2000. Regardless of the above, tests were carried out using FEMSA time series which had less data, and finally Microsoft's which was tested in order to compare with Kazem's results [23].

Table 2 show SAP-SVR results obtained for FEMSA time series; epsilon-polynomial and nu-polynomial obtained the best results in average and minimum MAPE, however epsilon-linear and nu-linear were the ones with better performance in average and minimum time. In terms of quality, for this time series, polynomial kernels presume better results in terms of quality and time performance.

Table 2. Results obtained by SAP-SVR for FEMSA

Variant SAP-SVR	Avg MAPE (%)	Avg Time(sec)	Minimum MAPE (%)	Minimum time (sec)
Epsilon-Linear	48.4031	18.7389667	48.4031	13.862
Epsilon-Polynomial	8.484137	20.0218	5.94461	14.958
Epsilon-Radial	16.84922	23.4795	16.4368	18.003
Epsilon-Sigmoidal	44.6949317	29.6124333	5.97046	24.632
Nu-Linear	42.485	18.0091	42.485	13.119
Nu-Polynomial	5.97739633	20.0866333	5.93463	15.018
Nu-Radial	15.70569	21.4593	15.6031	16.607
Nu-Sigmoidal	23.2490977	30.6045667	5.98164	25.025

CEMEX results are shown in Table 3, for this time series the variants with better average MAPE were linear variants epsilon and nu, and polynomial variants epsilon and nu as well, in the case of average time the linear kernels had better performance. Minimum MAPE was obtained by polynomial kernels but linear kernels had a better performance in minimum time, followed by polynomial kernels.

Table 3. SAP-SVR results for CEMEX

Variant SAP-SVR	Avg MAPE (%)	Avg Time(sec)	Minimum MAPE (%)	Minimum time (sec)
Epsilon-Linear	16.5779	48.4122414	16.5779	43.239
Epsilon-Polynomial	15.7460931	55.8145517	13.8406	47.607
Epsilon-Radial	35.0924966	72.6803793	15.5062	59.689
Epsilon-Sigmoidal	17.0088034	92.1935517	14.4589	81.997
Nu-Linear	15.7997	42.0014138	15.7997	36.847
Nu-Polynomial	15.9450414	57.4573793	13.85	49.99
Nu-Radial	19.8398517	58.3574138	15.614	52.077
Nu-Sigmoidal	17.3194655	91.7783103	14.4378	81.336

Table 4 show results obtained by SAP-SVR for LIVEPOL. Similar to other tests, polynomial kernels demonstrate a good average and minimum MAPE, but were over performed by linear kernels in average time. Minimum time is similar for linear and polynomial kernels, but linear kernels show a noticeable disadvantage against polynomial in terms of minimum MAPE. Sigmoidal kernels also show a good average MAPE, but its running time is very poorly in comparison to other variants.

Table 4. Results obtained by SAP-SVR for LIVEPOL

Variant SAP-SVR	Avg MAPE (%)	Avg Time(sec)	Minimum MAPE (%)	Minimum time (sec)
Epsilon-Linear	27.5373	51.1965	27.5373	47.72
Epsilon-Polynomial	7.94243733	55.3528333	6.92357	44.194
Epsilon-Radial	13.42848	67.7229333	13.3251	56.793
Epsilon-Sigmoidal	7.69495567	89.8794333	7.30943	85.224
Nu-Linear	28.4235	44.4761	28.4235	41.041
Nu-Polynomial	7.19511233	54.8760333	6.93011	45.327
Nu-Radial	13.8289533	57.7890667	13.727	53.529
Nu-Sigmoidal	7.44902833	90.24	7.32272	85.898

Finally, Table 5 show Microsoft results. Similarly, to previous results, epsilon-linear and nu-linear variants were the better ones in average MAPE and average time followed by polynomial kernels. The best Minimum MAPE was obtained by polynomial kernels, and minimum time was better in linear kernels.

Table 5. Results obtained by SAP-SVR for Microsoft

Variant SAP-SVR	Avg MAPE (%)	Avg Time(sec)	Minimum MAPE (%)	Minimum time (sec)
Epsilon-Linear	3.6798	9.91026667	3.6798	6.714
Epsilon-Polynomial	4.00522933	10.6734	3.60416	7.563
Epsilon-Radial	3.75351567	11.3927667	3.60723	8.137
Epsilon-Sigmoidal	3.778506	12.5135	3.6516	9.53
Nu-Linear	3.66813	10.2498667	3.66813	7.007
Nu-Polynomial	3.93371267	10.8349667	3.6051	7.457
Nu-Radial	4.27915733	11.2216667	3.64016	7.882
Nu-Sigmoidal	3.719003	12.5792667	3.61995	9.334

Table 6 shows the comparative results of the best SAP-SVR variant versus ARIMA. As can be observed, for this case, the hybrid algorithm SAP-SVR has a better performance in most of the cases. The variants of SAP-SVR and ARIMA taken for Table 6 are shown in Table 7. ARIMA experimentation was done using the same training and testing samples that the SAP-SVR algorithm.

Table 6. Comparative results between SAP-SVR and ARIMA.

	SAP-SVR Best Minimum MAPE	SAP-SVR Best Average MAPE	ARIMA MAPE
GCARSO	3.39353	3.402051111	9.364781
FEMSA	5.93463	5.97739633	2.099846
CEMEX	13.8406	15.7460931	39.27661
LIVEPOL	6.92357	7.19511233	20.07579
MICROSOFT	3.60416	3.66813	3.791307

Table 7. SAP-SVR variant with best performance and ARIMA parameters.

	SAP-SVR Variant	SAP-SVR Variant	ARIMA
GCARSO	Epsilon Sigmoidal	Nu-Linear	(2,3,2)
FEMSA	Nu-Polynomial	Nu-Polynomial	(3,1,3)
CEMEX	Epsilon-Polynomial	Epsilon-Polynomial	(3,2,2)
LIVEPOL	Epsilon-Polynomial	Nu-Polynomial	(3,2,3)
MICROSOFT	Epsilon-Polynomial	Nu-Linear	(0,0,1)

Microsoft is the only asset that was not taken from BMV, as it was tested in order to compare SAP-SVR to Kazem's work [22]. Table 8 shows the comparison, in which SAP-SVR obtained a better performance in terms of MAPE.

Table 8. Microsoft's results comparison between SAP-SVR and Kazem's models[22].

Model	MAPE
SVR-GA	6.6795
SVR-CGA	6.1031
SVR-FA	5.2653
SVR-CFA	5.1907
SAP-SVR	3.66813

SAP-SVR results show a better MAPE in comparison to classical method ARIMA. It was expected that SAP-SVR overperformed ARIMA, as SVR have shown better results than classical methods according to state-of-the-art. The results also showed an improvement in Microsoft's forecast in comparison to Kazem's work [22].

Not only epsilon-linear and epsilon-polynomial variant were the ones with better performance over the other SAP-SVR variants. Also, comparison between commonly used kernels in SVR was done in which linear kernels showed a faster running time in most cases, with good results.

5. Conclusions

In this paper, a new parallel forecasting algorithm (SAP-SVR) for time series is presented, which was tested using different BMV financial time series. This algorithm was developed using common kernels in regression financial time series problems. The algorithm has the advantage that SVR parameters are online tuned automatically, instead of doing a complicated offline tuning.

In addition, this hybrid algorithm does not just find the best parameters, but it also achieves a forecast with low average error. One of the main contributions of this work is that SA in the SAP-SVR is a simple and efficient method that guides the search of the best parameters during the execution that explores the *solution space*. Besides, the kernel with best performance for the time series forecasted was found. The paper shows a comparison of different SVR kernels in SVR.

The use of an external SVR library [24][24][24][25][24][24][25] was an important issue to simplify the implementation of the proposed algorithm. As a future project, we propose to remove external libraries dependency of SAP-SVR and the hybridization with other metaheuristics, as well as other parallelization techniques to reduce the execution time.

Acknowledgements

The authors would like to acknowledge with appreciation and gratitude to CONACYT, TECNM and PRODEP. Also, we thank the Laboratorio Nacional de Tecnologías de la Información del Instituto Tecnológico de Ciudad Madero for the access to its cluster. This work has been partial supported by CONACYT Project.254498.J. Javier González-Mancha would like to thank CONACyT for the support 710088.

References

- [1] G. Santamaría-Bonfil, J. Frausto-Solís, and I. Vázquez-Rodarte, "Volatility Forecasting Using Support Vector Regression and a Hybrid Genetic Algorithm," *Comput. Econ.*, vol. 45, no. 1, pp. 111–133, 2015.
- [2] J. Branke, B. Scheckenbach, M. Stein, K. Deb, and H. Schmeck, "Portfolio optimization with an envelope-based multi-objective evolutionary algorithm," *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 684–693, 2009.
- [3] D. J. Hillier, S. A. Ross, R. W. Westerfield, J. Jaffe, and B. D. Jordan, *Corporate finance*. McGraw Hill, 2010.
- [4] H. Markowitz, "Portfolio Selection," *J. Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [5] G. Hassan and C. D. Clack, "Multiobjective robustness for portfolio optimization in volatile environments," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO '08*, 2008, p. 1507.
- [6] J. P. Donate and P. Cortez, "Evolutionary optimization of sparsely connected and time-lagged neural networks for time series forecasting," *Appl. Soft Comput.*, vol. 23, pp. 432–443, 2014.
- [7] J. Frausto-Solís, M. Chi-Chim, and L. Sheremetov, "Forecasting Oil Production Time Series with a Population-Based Simulated Annealing Method," *Arab. J. Sci. Eng.*, vol. 40, no. 4, pp. 1081–1096, 2015.
- [8] A. Hossain, F. Zaman, M. Nasser, and M. M. Islam, "Comparison of GARCH, Neural Network and Support Vector Machine in Financial Time Series Prediction," in *Pattern Recognition and Machine Intelligence: Third International Conference, PReMI 2009 New Delhi, India, December 16-20, 2009 Proceedings*, S. Chaudhury, S. Mitra, C. A. Murthy, P. S. Sastry, and S. K. Pal, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 597–602.

- [9] J. Wang, B. Zhou, S. Zhou, and Z. Sheng, “Forecasting nonlinear chaotic time series with function expression method based on an improved genetic-simulated annealing algorithm,” *Comput. Intell. Neurosci.*, vol. 2015, p. 42, 2015.
- [10] E. Hanke John, W. Wichern Dean, and G. Reitsch Arthur, “Business Forecasting,” *Ed. 8th, PHI Publ. New Delhi,(p)*, vol. 521, 2005.
- [11] G. P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [12] M. Khashei and M. Bijari, “An artificial neural network (p, d, q) model for timeseries forecasting,” *Expert Syst. Appl.*, vol. 37, no. 1, pp. 479–489, 2010.
- [13] T. Bollerslev, “Generalized autoregressive conditional heteroskedasticity,” *J. Econom.*, vol. 31, no. 3, pp. 307–327, 1986.
- [14] P. Ou and H. Wang, “Volatility Modelling and Prediction by Hybrid Support Vector Regression with Chaotic Genetic Algorithms,” *Int. Arab J. Inf. Technol.*, vol. 11, no. 3, pp. 1–6, 2013.
- [15] S. Chen, K. Jeong, and W. Härdle, “Support vector regression based GARCH model with application to forecasting volatility of financial returns,” 2008.
- [16] V. Vapnik and A. Chervonenkis, “A note on one class of perceptrons,” *Autom. Remote Control*, vol. 25, no. 1, 1964.
- [17] A. B. M. Shawkat Ali, *Smart Grids: Opportunities, Developments, and Trends*. Springer London, 2013.
- [18] N. Chen, W. Lu, J. Vang, and G. Li, *Support Vector machine in Chemistry*. World Scientific, 2004.
- [19] A. Statnikov, C. F. Aliferis, and D. P. Hardin, *A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and methods*. World Scientific, 2011.
- [20] J. L. Ticknor, “A Bayesian regularized artificial neural network for stock market forecasting,” *Expert Syst. Appl.*, vol. 40, no. 14, pp. 5501–5506, 2013.
- [21] M. R. Hassan, B. Nath, and M. Kirley, “A fusion model of HMM, ANN and GA for stock market forecasting,” *Expert Syst. Appl.*, vol. 33, no. 1, pp. 171–180, 2007.
- [22] A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, and O. K. Hussain, “Support vector regression with chaos-based firefly algorithm for stock market price forecasting,” *Appl. Soft Comput.*, vol. 13, no. 2, pp. 947–958, 2013.
- [23] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *Int. J. Forecast.*, vol. 22, no. 4, pp. 679–688, 2006.
- [24] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27:1--27:27, 2011.
- [25] K. Deb and M. Goyal, “A combined genetic adaptive search (GeneAS) for engineering design,” *Comput. Sci. Informatics*, vol. 26, pp. 30–45, 1996.
- [26] H. Banda-Ortiz, F. A. Pérez-Sosa, and D. Gómez-Hernández, “Value at Risk (VaR) in uncertainty: Analysis with parametric method and Black & Scholes simulations (Valor en Riesgo (VaR) en incertidumbre: Análisis con el método paramétrico y simulaciones con Black & Scholes),” 2014.
- [27] “Yahoo Finance - Business Finance, Stock Market, Quotes, News.” [Online]. Available: <http://finance.yahoo.com/>. [Accessed: 16-Aug-2016].